Nonlinear Estimation and Modeling of Noisy Time-Series by Dual Kalman Filtering Methods

Alex Tremain Nelson
M.S., University of California, San Diego, 1995
Sc.B., Brown University, 1993

A dissertation submitted to the faculty of the
Oregon Graduate Institute of Science and Technology
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Electrical and Computer Engineering

September 2000

The dissertation "Nonlinear Estimation and Modeling of Noisy Time-Series by Dual Kalman Filtering Methods" by Alex Tremain Nelson has been examined and approved by the following Examination Committee:

Eric A. Wan
Associate Professor
Dept. of Electrical & Computer Engineering
Thesis Research Advisor

Todd K. Leen
Professor
Dept. of Computer Science

Michael W. Macon
Assistant Professor
Dept. of Electrical & Computer Engineering

John E. Moody
Professor
Dept. of Computer Science

Dedication

I dedicate this thesis to all of my teachers
– especially my parents, my brother, and my friends –
who have taught me so much over the years.

Acknowledgements

I would like to thank my professors and colleagues at the Oregon Graduate Institute for their intellectual support, and for providing a congenial and stimulating academic environment. In particular, I would like to thank Eric Wan for imparting his great enthusiasm for research and problem solving, and Todd Leen for sharing his mathematical expertise and rigor. Special thanks go to Michael Macon and John Moody for their input, and their efforts in reviewing this work, and to Bob Jaffe for putting a human face on DSP and optimization theory.

I would also like to express my appreciation to the following individuals for their invaluable support throughout the completion of this work (in approximate alphabetical order): Carrington Barrs, Chuck and Lois Delcamp, Ben Diedrich, Paul Esch and Peter Storniolo, "Texas" Dave Gambel, Leonie and Bill Griswold, Edd Hunter, Jimmy Langston, Aaron Marmorstein and Catherine Aylmer, John Melby, Erik Nelson, Phil Nelson, Sally Nelson, Ken Petty, Fred Philips and members of the OGI Aikido Club, Monty and Irene Snell, Rudolph van der Merwe, Greg and Julie Washburn, and Joy Zook.

Finally, I would like to thank the National Science Foundation for its generous financial support under Grants ECS-9410823 and IRI-9712346, and the Center for Spoken Language Understanding for the stimulation provided by countless lunch talks and cups of coffee.

Contents

List of Tables

List of Figures

List of Formulas

Abstract

Nonlinear Estimation and Modeling of Noisy Time-Series by Dual Kalman Filtering Methods

Alex Tremain Nelson

Ph.D., Oregon Graduate Institute of Science and Technology
September 2000

Thesis Advisor: Eric A. Wan

Numerous applications require either the estimation or prediction of a noisy time-series. Examples include speech enhancement, economic forecasting, and geophysical modeling. A noisy time-series can be described in terms of a probabilistic model, which accounts for both the deterministic and stochastic components of the dynamics. Such a model can be used with a Kalman filter (or extended Kalman filter) to estimate and predict the time-series from noisy measurements. When the model is unknown, it must be estimated as well; dual estimation refers to the problem of estimating both the time-series, and its underlying probabilistic model, from noisy data. The majority of dual estimation techniques in the literature are for signals described by linear models, and many are restricted to off-line application domains. Using a probabilistic approach to dual estimation, this work unifies many of the approaches in the literature within a common theoretical and algorithmic framework, and extends their capabilities to include sequential dual estimation of both linear and nonlinear signals. The dual Kalman filtering method is developed as a method for minimizing a variety of dual estimation cost functions, and is shown to be an effective general method for estimating the signal, model parameters, and noise variances in both on-line and off-line environments.

Chapter 1

Introduction

1.1 Overview

This thesis addresses the problem of modeling and estimating noisy discrete-time signals, or *time-series*. Numerous applications – ranging from speech enhancement, to economic forecasting, to adaptive control – require either the estimation, prediction, or modeling of a noisy time-series. In *estimation*, all data up to the current time is used to approximate the current value of the underlying clean time-series. *Prediction* is concerned with using all available data to approximate a *future* value of the clean series. *Modeling* (sometimes referred to as *identification*) is the process of approximating the underlying dynamics that generated the clean time-series

These tasks are strongly interdependent. For example, an accurate model of the system that generated the time-series can be used for estimation of the signal. Conversely, if the clean signal is available, it can be used to build an accurate model of the dynamics. Furthermore, if an accurate model and good signal estimates are available, good predictions can be generated by using the estimates as inputs to the model.

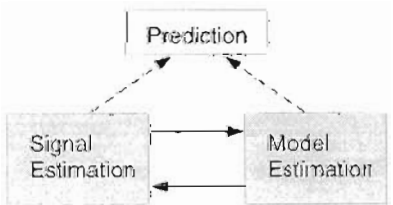


Figure 1.1: The dual estimation problem. Signal and model estimation are interdependent tasks; prediction requires solving both.

However, when neither the model nor the clean signal are known, the situation is much more challenging (see Figure 1.1). The problem of estimating (from noisy data) both the underlying signal and the model that produced it is the central topic of this thesis, and will be referred to herein as the *dual estimation* problem.

The next section presents a set of basic assumptions about how the noisy data were generated, and introduces much of the notation used throughout the thesis. The remainder of this introductory chapter contains a brief motivational description of the dual estimation problem, followed by a review of work done by other researchers to date, and a preview of the contributions made in this thesis.

Chapter 2 uses a probabilistic approach to generate several cost functions that quantify (in different ways) what is meant in the preceding text by "good" or "accurate" estimates and models. Chapter 3 describes an algorithmic framework for minimizing these cost functions, which includes the expectation-maximization (EM), recursive prediction error (RPE), and some new algorithms as specific examples. Although particular attention is paid to linear and neural network models, the algorithms described are applicable to a broader class of models that are differentiable in their inputs and parameters. Finally, Chapter 4 gives an experimental comparison of the cost functions, and Chapter 5 demonstrates the practical application of the algorithms using several real-world examples.

1.2 Assumptions and Notation

1.2.1 Model Structure

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

$$\begin{aligned} x_k &= f\big(x_{k-1}, \ldots, x_{k-M}, \mathbf{w}\big) + v_k \\ y_k &= x_k + n_k, \qquad\qquad \forall k \in \{1 \ldots N\} \end{aligned} \tag{1.1}$$

where x_k corresponds to the true underlying time-series driven by process noise v_k, and $f(\cdot)$ is a nonlinear function (*e.g.*, a neural network) of the past M values of x_k parameterized by $\mathbf{w}$. The only available observation is y_k, which contains additive noise n_k. The time-series is one-dimensional; *i.e.*, the noisy observation $y_k \in \Re$ is a scalar. The situation is depicted in Figure 1.2. The notation $\{y_k\}_1^t$ is used herein to represent the sequence of data, $\{y_1, y_2, y_3, \ldots, y_t\}$.

This model structure is fairly general. Loosely speaking, Takens' theorem [82] states that the dynamics of a discrete-time system with state-space dimension d can be reconstructed in a $2d+1$ dimensional space constructed from a vector of observations on the system $[x_{k-1}, \ldots, x_{k-(2d+1)}]^T$. In other words, the first part of Equation 1.1 can accurately model the dynamics of any unknown d-dimensional system with observed variable x_k, as long as $M \geq 2d+1$ (see also [73, 31]), and as long as the parameterized class of models $f(\cdot)$ is broad enough.

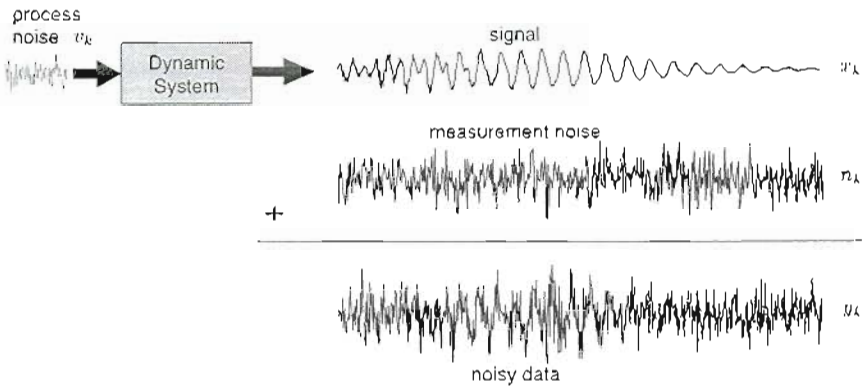


Figure 1.2: The data are assumed to be generated by an unknown nonlinear autoregressive model, and corrupted by additive measurement noise.

Although this thesis is concerned exclusively with *time-series* modeling and estimation, the concepts it explores can be readily generalized to other system identification applications. For example, including a user-determined input to the function $f(\cdot)$ would produce a nonlinear ARX (autoregressive, exogenous input) model. Similar extensions to $ARMA$ (autoregressive moving average) and $ARMAX$ models are also possible, as are extensions to multi-dimensional data sets ($\dim(y_k) > 1$).

A more general formulation might also include nonlinear *channel effects* of the form $y_k = g(x_k, \ldots, x_{k-M+1}, n_k)$. The framework developed in this thesis can be easily adjusted to include such a nonlinear measurement equation, as long as the channel function $g(\cdot)$ is known and differentiable[1]. In Equation 1.1, this function takes the special form $g(x_k, \ldots, x_{k-M+1}, n_k) = x_k + n_k$, representing corruption by additive noise.

A Gaussian assumption on the noise terms will facilitate the derivation of cost functions from a probabilistic perspective in Chapter 2. However, much of the analysis is valid for non-Gaussian noise as well. Also, the algorithms discussed in this thesis often remain useful when the Gaussian assumption ceases to hold, as is demonstrated experimentally by the example applications in Chapter 4.

1.2.2 System Identification Loop

The methods described in this thesis must inevitably be used within a *system-identification* loop [46] of repeatedly: (1) selecting a model set, (2) choosing a cost function and algorithm to search for and select a model from that set, and then (3) validating the model (see Figure 1.3). The

[1] Corruption by *unknown* channel effects represents a *blind deconvolution* problem, and is considerably more difficult unless additional constraints or assumptions are used.

methods presented herein address only the second of the three steps, under the assumption that mechanisms are in place for performing model set selection and model validation.

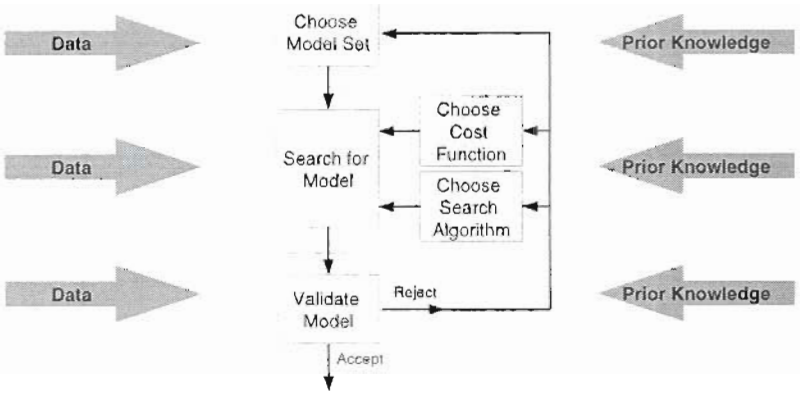


Figure 1.3: The system identification loop. The elements enclosed by the dashed line are addressed in this thesis. Figure adapted from [46].

Although the model set is partially defined by the noisy AR process of Equation 1.1, it remains overly broad because the form of the function $f(\cdot)$ is not specified. Before the optimization methods described in this thesis can be applied, the model set must be more narrowly defined in terms of the order M and particular functional form of $f(\cdot)$. For example, $f(\cdot)$ might be defined as a 2-layer feedforward neural network with 10 inputs ($M = 10$), 5 hidden units, and one output, or as fifth-order ($M = 5$) linear model. The methods of this thesis apply for any $f(\cdot)$ differentiable in x_k and $\mathbf{w}$.

A parameterization of the model set in terms of $\mathbf{w}, \sigma_v^2$, and σ_n^2 can then be defined to allow for a search over the model set. This parameterization is known as a *model structure*[2][46]. This thesis discusses the selection of a suitable cost function and algorithm for searching within a pre-specified model structure. As mentioned in the Overview, estimating the signal $\{x_k\}_1^N$ is an integral and necessary part of this model estimation step. A detailed discussion of this fact follows in the next section.

[2] More formally, a model structure is a differentiable mapping from a connected, open subset of $\Re^d$ (the d-dimensional parameter space) to a model set [46].

1.3 The Dual Estimation Problem

What follows is a qualitative description of the dual estimation problem. We consider the problem from three different motivational perspectives: modeling, estimation, and prediction.

1.3.1 Modeling

Suppose we are interested in modeling the dynamics $f(\cdot)$ of the underlying clean time-series x_k.[3] A simplistic approach is to ignore the effect of the additive noise n_k, and build an autoregressive model $f_y(\cdot)$ from a vector $\mathbf{y}_k = [y_k, \ldots, y_{k-M+1}]^T$ of past values to predict the next value y_{k+1}, as shown below. Such a model could be trained directly on the noisy data by minimizing the squared

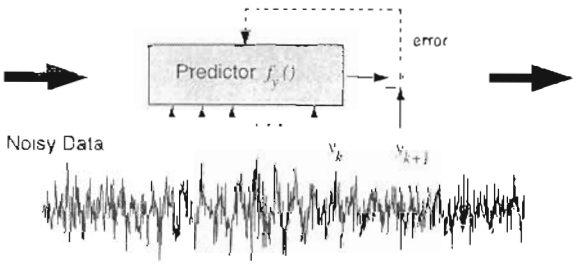


Figure 1.4: Building a predictor on noisy data

prediction error, or some other cost. Unfortunately, the resulting model $f_y(\cdot)$ will be biased with respect to $f(\cdot)$ in Equation 1.1. This is because the former is a function of y_k, while the latter is a function of x_k. The effect is most easily seen by considering the least squares predictor for a linear model:

$$\hat{\mathbf{w}}_y \triangleq (\widehat{E[\mathbf{y}_k \mathbf{y}_k^T]})^{-1} \cdot \widehat{E[\mathbf{y}_k y_{k+1}]}, \tag{1.2}$$

where $\widehat{E[\cdot]}$ denotes the sample average. The expectation can be shown to be the optimal Wiener solution for a finite causal linear model on the data.

$$E[\hat{\mathbf{w}}_y] = (E[\mathbf{y}_k \mathbf{y}_k^T])^{-1} \cdot E[\mathbf{y}_k y_{k+1}] \qquad \triangleq \mathbf{w}_y^*. \tag{1.3}$$

The signal and noise are mutually independent, so:

$$\begin{aligned} & E[\hat{\mathbf{w}}_y] = (E[\mathbf{x}_k \mathbf{x}_k^T] + E[\mathbf{n}_k \mathbf{n}_k^T])^{-1} \cdot (E[\mathbf{x}_k x_{k+1}] + E[\mathbf{n}_k n_{k+1}]) \\ \Rightarrow\ & E[\hat{\mathbf{w}}_y] \neq (E[\mathbf{x}_k \mathbf{x}_k^T])^{-1} \cdot E[\mathbf{x}_k x_{k+1}] \qquad \triangleq \mathbf{w}, \\ \therefore\ & E[\hat{\mathbf{w}}_y] \neq \mathbf{w}. \end{aligned} \tag{1.4}$$

[3] This task is also referred to as dynamic reconstruction [31].

The optimal Wiener predictor for $\{y_k\}_1^N$ (given by $\mathbf{w}_y^\star$) is *not* the same as the Wiener predictor $\mathbf{w}$ for $\{x_k\}_1^N$. Hence, a least-squares model for $\{y_k\}_1^N$ (which has the expected value $\mathbf{w}_y^\star$) will be *biased* with respect to $\mathbf{w}$.

However, an unbiased linear model can be generated using a simple adjustment to the least-squares solution, provided that the statistics of the noise are known:

$$\begin{aligned} \hat{\mathbf{w}}_y^u &\triangleq (\widehat{E[\mathbf{y}_k\mathbf{y}_k^T]} - E[\mathbf{n}_k\mathbf{n}_k^T])^{-1} \cdot (\widehat{E[\mathbf{y}_k y_{k+1}]} - E[\mathbf{n}_k n_{k+1}]) \\ \Rightarrow E[\hat{\mathbf{w}}_y^u] &= (E[\mathbf{x}_k\mathbf{x}_k^T])^{-1} \cdot E[\mathbf{x}_k x_{k+1}] \\ &= \mathbf{w}. \end{aligned} \tag{1.5}$$

Unfortunately, the additive noise in the data $\{y_k\}_1^t$ will induce a higher variance in the sample correlations, $\widehat{E[\mathbf{y}_k\mathbf{y}_k^T]}$ and $\widehat{E[\mathbf{y}_k y_{k+1}]}$. This contributes to higher variance in the parameter estimates, $\hat{\mathbf{w}}_y^u$. Even though it is unbiased, the variance in the model estimate means that any particular $\hat{\mathbf{w}}_y^u$ is unlikely to be accurate.

A related deterministic approach, known as *total least squares* (TLS), performs principal components analysis on the noisy data to produce an unbiased estimate [25]. However, TLS solutions are subject to the same variance problems as the above unbiased least squares estimate.

Another alternative is to build a model from signal estimates $\hat{\mathbf{x}}_k$ that have lower variance than $\mathbf{y}_k$, such that the Wiener solution:

$$\hat{\mathbf{w}}_{\hat{x}} \triangleq (E[\hat{\mathbf{x}}_k\hat{\mathbf{x}}_k^T])^{-1} \cdot E[\hat{\mathbf{x}}_k y_{k+1}] \tag{1.6}$$

is unbiased. The least squares approximation to this solution (replacing expectations with $\widehat{E[\cdot]}$) is therefore unbiased and has lower variance. The signal estimates $\hat{\mathbf{x}}_k$ will have this property provided that: (1) they are optimal in the sense that $\hat{\mathbf{x}}_k$ is uncorrelated with the error $\tilde{\mathbf{x}}_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)$; (2) they are causal, so that $\hat{\mathbf{x}}_k$ is independent of the noise term n_{k+1} and v_{k+1} at the next time step. Using the above definition of $\tilde{\mathbf{x}}_k$, and $y_{k+1} = \mathbf{w}^T\mathbf{x}_k + v_{k+1} + n_{k+1}$:

$$\hat{\mathbf{w}}_{\hat{x}} = (E[\hat{\mathbf{x}}_k(\mathbf{x}_k + \tilde{\mathbf{x}}_k)^T])^{-1} \cdot E[\hat{\mathbf{x}}_k(\mathbf{x}_k^T\mathbf{w} + v_{k+1} + n_{k+1})]. \tag{1.7}$$

Applying the optimality condition to the first term and the causality condition to the second term, this reduces to:

$$\hat{\mathbf{w}}_{\hat{x}} = (E[\mathbf{x}\mathbf{x}^T + \tilde{\mathbf{x}}_k\mathbf{x}_k^T])^{-1} \cdot E[\mathbf{x}\mathbf{x}^T + \tilde{\mathbf{x}}_k\mathbf{x}_k^T]\mathbf{w} \tag{1.8}$$

$$= (E[\mathbf{x}\mathbf{x}^T + \tilde{\mathbf{x}}_k\mathbf{x}_k^T])^{-1} \cdot E[\mathbf{x}\mathbf{x}^T + \tilde{\mathbf{x}}_k\mathbf{x}_k^T]\mathbf{w} \tag{1.9}$$

$$= \mathbf{w}, \tag{1.10}$$

so the optimal solution is attained, and the corresponding least squares solution is unbiased. However, the requisite estimates $\hat{\mathbf{x}}_k$ can be generated by an optimal (*i.e.*, Kalman) filter only if the model is *known*. The dual estimation problem can be viewed as the need to generate these estimates in order to estimate an *unknown* model.

1.3.2 Estimation

Sometimes, one is primarily interested in the estimation of a noisy signal; *i.e.*, estimating $\{x_k\}_1^t$ from the noisy data $\{y_k\}_1^t$. A common approach to this problem is to use information about the noise statistics to subtract the noise in the magnitude spectral domain [4]. Nonlinear variations on this *spectral subtraction* approach perform the subtraction in other domains. These approaches typically suffer from distortion of the signal due to an overestimation of the noise spectrum. They also require a block-wise form of processing (to compute the transform) which precludes their use in applications that require on-line estimation.

Another general approach is to find a mapping from a window of the noisy data, to the corresponding window of the clean signal. The mapping (which could also operate in a transform domain) can be found by using a training set of noisy input data and clean target data. Aside from being a non-causal, or block-wise approach, this method suffers from being limited to the data represented in the training set. The learned mapping will not generalize to signals with statistics that are different from the training set.

The focus of this thesis is on *sequential*, on-line estimation approaches that operate in the time-domain and which do not require a separate set of training data. In the case of linear models and Gaussian statistics when the model parameters $\mathbf{w}$ and variances are *known*, the celebrated Kalman filter ([36],1961) produces optimal estimates ($\hat{x}_k = E[x_k|\{y_t\}_1^k, \mathbf{w}]$) of the signal given all the past measurements. The *extended* Kalman filter (EKF) is an approximate method in the case of *nonlinear* models, and approximates the nonlinear model as time-varying linear model during certain steps in the estimation process. Lewis ([43],1986) provides a comprehensive review.

However, an immutable characteristic of Kalman filtering approaches is their requirement that the model of the system dynamics be *known*. This is not the case in the present context; another view of the dual estimation problem is the need to estimate the model in order to estimate the signal.

1.3.3 Prediction

The task of prediction is interesting because it shows how the problems of estimation and modeling are related. Suppose a prediction is required for the next value y_{t+1} of a noisy time-series $\{y_k\}_1^t$, known to be generated according to Equation 1.1. A simple solution is to build the autoregressive model $f_y(\cdot)$ described in Section 1.3.1, and generate predictions as $\hat{y}_{t+1} = f_y(\mathbf{y}_t, \hat{\mathbf{w}}_y)$. While it was noted that the model $f_y(\cdot)$ is biased with respect to $f(\cdot)$, the *predictions* produced by this model would, in fact, be unbiased.

On the other hand, any particular $f_y(\cdot)$ will not necessarily give accurate predictions because of the previously described variance of the modeling process. Furthermore, the predictions obtained from a given $f_y(\cdot)$ will themselves have high variance due to the additive noise on the inputs to the predictor.

The above approach does not take advantage of the special relationship of a particular input to other inputs in the window, or to inputs in other windows. In fact, fitting $f_y(\cdot)$ to the noisy data is equivalent to treating the problem like a standard regression task, where there is no particular relationship between each of the inputs. It is important to note that the variance of the predictions can be reduced by exploiting the knowledge that the data are from a time-series generated according to Equation 1.1.

Because the data are from a noisy AR process, all of the past data $\{y_k\}_1^t$ can, in theory, be used to improve the prediction of y_{t+1}. However, using a growing window of all past data as the input to a predictor is not practical because the number of parameters would increase as well. In the linear case, a Kalman filter uses knowledge of the AR model to summarize the past data $\{y_k\}_1^t$ with a finite vector $\hat{\mathbf{x}}_t$ such that $E[y_{t+1}|\hat{\mathbf{x}}_t] = E[y_{t+1}|\{y_k\}_1^t]$. In fact, $\hat{\mathbf{x}}_t$ represents the conditional expectation of the lagged values of the signal given all the past data, and the model.

Because $\hat{\mathbf{x}}_t$ will have lower variance than $\mathbf{y}_t$, it results in lower variance predictions. Also, as noted previously, a predictor trained using $\hat{\mathbf{x}}_k$ as inputs will be unbiased with respect to the autoregressive function $f(\cdot)$. The problem, of course, is finding the estimates $\hat{\mathbf{x}}_k$ when the model is unavailable (the Kalman filter requires a known model). Once again, this is the dual estimation problem.

1.3.4 Additional Comments

Note that even when the model is linear, (*i.e.*, $f(\cdot) = \mathbf{w}^T\mathbf{x}_{k-1}$, where $\mathbf{x}_{k-1} = [x_{k-1}, x_{k-2}, \ldots, x_{k-M}]$) the inner product of the parameter vector $\mathbf{w}$ with the vector $\mathbf{x}_{k-1}$ indicates a bilinear relationship between these unknown quantities. Hence, even in the simplest case, linear estimation methods

such as least squares are not applicable for the dual estimation problem, and numerical optimization techniques are required.

This thesis focuses primarily on approaches that make use of statistical information about the data; this information ultimately involves the statistics of the noise terms v_k and n_k. However, in many practical applications, the statistics of either one (perhaps both) of these noise processes will be unknown. The dual problem of estimating the weights $\mathbf{w}$ and signal x_k will in such cases also involve the estimation of this additional statistical information.

Most of the previous work on dual estimation has been restricted to the linear model case. Many of these methods are reviewed in the next section, along with the limited number of methods for nonlinear models. This thesis unifies many of these linear and nonlinear approaches in the context of neural network models. This and other contributions are described in Section 1.5.

1.4 Related Work

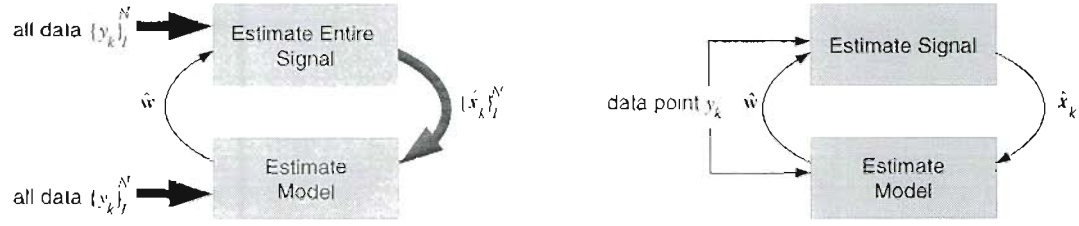
1.4.1 Iterative vs. Sequential Methods

A variety of methods have been proposed for dual estimation. Some involve an *iterative* scheme of repeatedly estimating the time-series using the current model and all available data, and then estimating the model using the estimates and all the data (see Figure 1.5(a)). Some of these iterative methods work in the frequency-domain (or some other transform-domain), and some work directly on the data in the time-domain. Iterative schemes are necessarily restricted to off-line applications, where a batch of data has been previously collected for processing. However, note that both the signal and weight estimation steps of an iterative scheme can be performed using either *batch-mode* or *pattern-mode* forms of processing [4].

Other dual estimation methods involve *sequential* estimation of both the model and the time-series simultaneously from the data (see Figure 1.5(b)). Sequential algorithms are *recursive* in nature, and can be used to process data on-line, as it becomes available (they are necessarily time-domain algorithms). Alternatively, they can also be used for efficient off-line processing, where the sequential algorithm makes several passes over the same block of data. Some discussion of the advantages of this approach is given by Ljung and Söderström [47].

This thesis is primarily concerned with *sequential* algorithms. However, a strong relationship exists between many time-domain iterative methods and sequential methods. For this reason, several of these iterative methods are described in this section.

[4] Batch-mode processing refers to updating the estimates only once, after all the data have been observed. Pattern-mode processing refers to updating the estimates each time a training pattern is observed [30].



(a) Iterative (b) Sequential

Figure 1.5: Two approaches to the dual estimation problem. Iterative approaches use large blocks of data repeatedly. 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 where the noise terms v_k and n_k are uncorrelated zero-mean white Gaussian processes with variances σ_v^2 and σ_n^2, respectively. An overview of these "linear" methods is provided first, followed by the work done for nonlinear models. Additional details about the algorithms will be provided in the contexts of Chapters 2 and 3.

1.4.2 Linear Models

Adaptive Estimation

As mentioned in Section 1.3.2, when the model parameters $\mathbf{w}$ and variances are *known* for the class of linear systems just described, the Kalman filter ([36],1961) produces maximum-likelihood estimates of the signal given all the past measurements[5]. Although originally developed in the context of automatic control systems, the Kalman filter has proven useful in a broad range of fields. For example, Paliwal and Basu ([63], 1987) investigate the use of Kalman filtering for speech enhancement. Additional refinements to the method, including extensions for colored measurement noise n_k [37, 24] have been developed elsewhere.

When the dynamics and statistics of the time-series are not known in advance, however, the Kalman filter cannot be applied. Hence, much of the early work on the dual estimation problem is concerned with Kalman filtering when the model parameters (or noise variances) are *not* completely known; this area of research is called *adaptive estimation* [2].

In early work, Kopp and Orford ([38],1963) and Cox ([12],1964) propose including both the lagged signal vector $\mathbf{x}_{k-1}$ and unknown parameters $\mathbf{w}$ in a combined state vector to form a joint

[5]Rauch, Tung, and Striebel ([68],1965) proposed a variant (often referred to as the Kalman smoother) that combines forward and backward filtering to allow for recursive estimation of the estimates of the signal given *all* the available data, past and future.

nonlinear state-space representation. The extended Kalman filter is then applied to the resulting nonlinear estimation problem. We will refer to this approach as the *joint extended Kalman filter* (joint EKF). Ljung ([45],1979) provides an extensive convergence analysis of the method, and discusses the importance of computing the sensitivity of the Kalman gain to the parameters. Niedźwiecki and Cisowski ([62],1996) make further practical enhancements to the algorithm for detecting and handling outliers.

Motivated by some convergence problems exhibited by the joint EKF, Nelson ([61],1976) proposes using two separate Kalman filters to provide an alternative solution to the dual estimation problem. In this *dual Kalman* approach, one filter produces estimates of the *signal* assuming the model is known, and the other filter produces *parameter* estimates assuming the signal is known. Ljung and Söderström ([47],1983) put the dual Kalman method into a general family of recursive identification algorithms, and include the use of recursive "sensitivity equations" for computing the derivatives of the recursive structure.

Maximum-Likelihood Approaches

Akaike ([1],1973) approaches dual estimation from within a maximum-likelihood context. Gupta and Mehra ([26],1974) discuss the potential pitfalls of maximum-likelihood parameter estimation, and the use of Kalman filtering and nonlinear programming approaches. In the iterative approach of [26], the Kalman filter is used to evaluate the conditional means and error covariances required for evaluating the likelihood function; maximum-likelihood parameter estimates are found by a variety of batch optimization techniques.

Another well-known iterative approach within the maximum-likelihood framework was presented by Lim and Oppenheim ([44],1978) for the problem of speech enhancement. A recursive least squares algorithm was used to estimate the model parameters, while a frequency-domain Wiener filtering approach was used for signal estimation. This paper was largely responsible for introducing the speech enhancement community to dual estimation with AR models.

EM Approaches

A somewhat different iterative approach to maximum-likelihood dual estimation is given by the expectation-maximization (EM) algorithm, first developed by Dempster et al. ([16],1977), and subsequently applied to time-series smoothing by Musicus and Lim ([58],1979) and Shumway and Stoffer ([76],1982). In each iteration, the conditional expectation of the signal is computed, given the data and the current estimate of the model (E-step). Then the model is found which maximizes a function of this conditional mean (M-step). Additional details are given on pages 36,85, and in

Appendix F. The approach has the advantage of some theoretical guarantees of convergence in the linear case. A batch form of the algorithm for pole-zero models is derived in [58]. In [76], the E-step is computed with a Kalman smoother, and the M-step is computed in closed form. The algorithm has been implemented and extended by several other researchers.

Weinstein et al. ([93],1994) extend the EM algorithm of [76] for two-microphone speech enhancement, and suggest a Kalman filter E-step, and gradient based M-step to allow for a sequential version of the algorithm. Other extensions for speech enhancement appear in [41, 42, 21]. Krishnamurthy et al. ([39],1998) propose using Kalman smoothers for both the E and M steps, and apply the algorithm to estimation of a broad class of bilinear systems. A sequential variation based on two Kalman filters is also suggested (but not implemented). Ghahramani ([22],1998) shows how the EM algorithm can be put in the context of learning dynamic Bayesian networks, while Blake et al. ([3],1999) combine a type of Monte Carlo sampling with the EM algorithm for learning multi-class linear dynamics for visual object trackers.

1.4.3 Nonlinear Models

By and large, the methods discussed above are for dynamic system models that are *linear* in the parameters and in the signal. The field of artificial neural networks has generated many papers on the topic of identifying *nonlinear* dynamic systems. While the majority of these papers assume that the training data (*i.e.*, the time-series) are *clean*, several of the approaches in these papers are strongly related to the dual estimation task.

Neural Network Training Methods

Although a pretrained neural network model can be used for the task of signal estimation, the Kalman filter cannot be applied directly to such nonlinear system models. Instead, the model must be linearized at every time step to allow for approximate propagation of the covariance of the estimated state. This algorithm is the *extended Kalman filter* (EKF)[43]. Assuming the model parameters $\mathbf{w}$ are known, and the noise terms v_k and n_k are Gaussian with known variances, the EKF produces *approximate* maximum-likelihood estimates of the signal.

As proposed by Singhal and Wu ([77],1989) and described by Plumer ([65],1995), the EKF can also be used as a means of training (*i.e.*, estimating the parameters of) a neural network. When used as a parameter estimation method, the EKF can be viewed as an efficient second-order nonlinear programming approach similar to the Gauss-Newton update rule [48]. Puskorius and Feldkamp ([66],1994) extend the approach to recurrent neural networks and nonlinear dynamic

systems, and present a decoupled version which exchanges some performance for computational efficiency.

In the context of training recurrent neural networks, Matthews ([51],1990) estimates both the hidden neuron outputs and network weights concurrently by combining them in a single state vector, and applying the EKF. This algorithm is quite similar to the joint EKF mentioned above for the linear case. However, while [38, 12, 45, 62] are concerned with estimating signals from noisy data, Matthews [51] uses the approach for training with clean data. Here, the state-estimation helps provide targets for the hidden layers of the recurrent network at the same time that the weights are being updated. Williams ([96],1992) describes the relationship between the joint EKF and real-time recurrent learning (RTRL) algorithms, and gives an analysis of the computational requirements. More recently, Sum et al. ([81],1998) augment the joint EKF training algorithm with a probabilistic pruning method.

Matthews ([52],1994) also proposes using two separate EKFs for estimating the hidden outputs and weights of recurrent neural networks. This algorithm is essentially a nonlinear extension of the dual KF method [61], in the form outlined by Ljung and Söderström [47]. Again, however, state estimation is used to supply targets to the hidden units of a network trained on clean data, rather than for estimating a noisy signal.

The EM algorithm has been applied by numerous authors to nonlinear system identification. Jordan and Jacobs ([34],1994) develop both batch and on-line algorithms for estimating the parameters of a hierarchical mixture of experts model. An EM algorithm for training neural networks on clean data is presented by de Freitas, Niranjan, and Gee ([14],1998). Here, the weights are given a dynamic system representation of their own (to potentially allow for modeling non-stationary systems). The weights are estimated *via* a Kalman smoother (E-step), and the dynamics of the weights are estimated during the M-step.

While the Kalman-based approaches inherently assume Gaussian densities on the data and states, there has been renewed interest recently in Monte Carlo methods for non-Gaussian state estimation. In the context of neural network training, de Freitas et al. [15] investigate a training algorithm based on sequential Monte Carlo techniques.

Dual Estimation Methods

All of the neural network training methods described above are for parameter estimation using clean data; only a few papers appear in the literature that are explicitly concerned with dual estimation for neural networks models.

Connor et al. ([10],1994) propose an iterative approach to training recurrent neural networks

for robust time-series prediction tasks. The algorithm alternates between applying a robust form of the EKF to estimate time-series, and using these estimates to train the neural network *via* gradient descent (using back-propagation [72, 95]). The work is an extension of robust estimation methods for linear ARMA models described by Martin ([50], 1982).

Weigend and Zimmerman's ([92],1995) *Clearning* algorithm is a heuristic method for training a neural network with noise on the input and target data, and can be applied to dual estimation for noisy time-series. The cost function can be shown to be a simplified approximation to the errors-in-variables cost function discussed on this page. While it allows for sequential estimation, the simplification can lead to severely biased results [87].

An approach developed by Stubberud and Owen ([80],1996) uses an adaptive EKF as a state-observer in a model reference adaptive control framework. Here, the system dynamics are *partially* known, and the EKF estimates the unmodeled component of the dynamics along with the state. Because the state is only observed through additive noise, this essentially constitutes a dual estimation problem (although it is not a time-series problem *per se*). The algorithm is similar to the joint EKF approaches described in Section 1.4.2.

Ghahramani and Roweis ([23],1999) show an EM approach to the dual estimation problem, using radial basis function (RBF) networks. An extended Kalman smoother is used for the E-step, and a closed-form solution to the RBF weights for the M-step. Briegel and Tresp [5] propose some variants on the EM algorithm by offering three alternative E-steps for signal estimation, all based on a Monte Carlo sampling approach. For weight estimation, a generalized M-step is performed by gradient descent. More recently, Wan et al. ([90],2000) demonstrate how an algorithm called the *unscented filter* can produce a more accurate nonlinear E-step without the use of Monte Carlo sampling.

The general idea behind the Monte Carlo (or particle filter) approaches is to either improve the Gaussian approximation to certain crucial densities, or to avoid the Gaussian assumptions altogether. While offering the potential of better performance than Kalman filtering methods, these methods generally incur higher computational expense. While some of the theory developed in Chapter 2 is relevant for the more general non-Gaussian case, the cost functions and algorithms in this thesis will focus on the Gaussian case, wherein Kalman filtering methods are appropriate.

Errors in Variables Models

Errors-in-variables (EIV) models appear in the nonlinear statistical regression literature (Seber and Wild, [75] 1989), and are used for regressing on variables related by a nonlinear function, but measured with some error. EIV methods involve iteratively maximizing a joint likelihood function

for the input and output data of the regression.

The form of the EIV cost function for time-series data is derived in Appendix G. In Chapter 2, this same *joint cost* is derived from a maximum *a posterior* (MAP) perspective, and relationship between this cost and the maximum-likelihood approaches is also discussed. However, errors-in-variables is an iterative approach involving batch computation; it tends not to be practical for time-series data because the computational requirements increase in proportion to N^2, where N is the length of the data.

1.5 Contributions of the Thesis

Dual estimation methods for nonlinear time-series models are relatively few, especially when compared with methods for linear models. This is to be expected, seeing that the fields of linear estimation, signal processing, and control are considerably more developed than their nonlinear counterparts. The existing methods offer a variety of approaches (*i.e.*, adaptive Kalman, maximum-likelihood, and EM) which share some common traits, but whose similarities and differences have not been sufficiently explicated in the literature. The first goal of this thesis is to provide a theoretical foundation for relating these methods; the second goal is to use this foundation to generate a family of sequential dual estimation methods for nonlinear time-series models.

1.5.1 Theoretical Framework

Any approach to dual estimation must be based on some explicit or implicit definition of optimality. A *cost function* provides a quantitative measure of the quality of the model and signal estimates, and generally forms the basis for designing a suitable algorithm. The methods in the preceding section correspond to a variety of cost functions; comparing these methods without an understanding of how their respective cost functions relate is not terribly illuminating.

This thesis provides a probabilistic treatment of the dual estimation problem, and suggests the feasibility of two main approaches to it. The relationship between these approaches, which holds in the general non-Gaussian case, arises from the probabilistic framework. Employing a Gaussian assumption on the noise produces several different cost functions, each corresponding to a different approximation. Some of these cost functions are identical to those investigated previously for the linear case; others are novel. However, the theoretical foundation of these cost functions enables the explication of their relationship to one another.

1.5.2 Sequential Methods

Many applications demand online, or sequential processing of data, as measurements become available. Sequential processing has the additional benefit of reduced memory requirements, and the flexibility to be applied in either on-line or off-line settings. The focus of this thesis is therefore on developing sequential methods for dual estimation within the theoretical framework described above. In some instances, only off-line methods have been previously investigated in the literature.

1.5.3 Algorithmic Framework

The Gaussian assumption on the noise largely justifies the use of Kalman-filter-based approaches to dual estimation. An algorithmic framework called *dual extended Kalman filtering* (dual EKF) is developed for minimizing the various cost functions. The framework includes maximum-likelihood, recursive prediction error, EM, and some novel algorithms as special cases, and is applicable to both linear and neural network model structures. The relationship of these contributions to existing methods is clarified in Table 1.1; the cost functions are explained in Chapter 2.

Table 1.1: Algorithmic contributions of this thesis to the problem of dual estimation. For each cost function listed, references are given for algorithms categorized as iterative or sequential approaches, using either linear or nonlinear models. Symbols indicate:(★) developed in this thesis or in a previous publication by the author; (♦) applied to a problem other than dual estimation; (♣) EIV algorithm;(♠) a significant approximation is made to the cost function.

	Iterative		Sequential	
Cost	Linear	Nonlin.	Linear	Nonlin.
prediction error	[50]	[10]	[47]	[87]★, [52]♦
maximum-likelihood	[26]		★	★
expectation-maximization (EM)	[58, 76]	[23, 5, 34]	★	★, [34]♦
joint (MAP)	[75]♣	[75]♣	[38, 12, 45]	[60]★, [80, 51]♦,[92]♠
error coupled			★	★

Although most of the cost functions explored in this thesis have been previously considered (at least in the context of linear models), many of these costs have been applied primarily in an off-line, iterative setting. Meanwhile, many applications require that the dual estimation problem be solved on-line, as data become available; the dual EKF algorithms minimize the dual estimation costs *sequentially*, thereby offering this needed capability.

Promising results have been published for a prediction error form of the dual EKF (Wan and Nelson [87],1997). The algorithm was successfully applied to single-microphone speech enhancement [59, 87, 89] problems, and is essentially a nonlinear counterpart to the linear RPE algorithm

[61, 47] described in the last section[6]. This thesis develops this and other members of the dual EKF family.

1.5.4 Variance Estimation

As stated in Section 1.3, the statistics of the measurement and process noises are generally useful (if not *crucial*) for dual estimation. Under a zero-mean Gaussian assumption, this information amounts to the variances of these noise processes. However, the few nonlinear-model dual estimation approaches that appear in the literature employ *ad hoc* methods of choosing these variances. An important part of this thesis is its investigation of theoretically justified variance estimation techniques in the context of dual estimation for nonlinear models.

1.5.5 Experimental Comparisons

The various cost functions can all be justified theoretically given different sets of approximations. Determining which of these approximations are better or worse on theoretical grounds is extremely difficult, if not impossible. Experiments are therefore performed on a variety of different data sets in order to facilitate useful conclusions about the pros and cons of the different approaches. For example, the dual EKF is found to perform the best with the maximum-likelihood and joint cost function listed in Table 1.1. Note that these conclusions are made much more meaningful through the use of a common algorithmic framework, which minimizes the spurious differences between the methods.

1.5.6 Applications

The practical use of the dual EKF algorithms is shown in several application domains. Special considerations must be made for different classes of signals. Specifically, in the domain of speech enhancement, the nonstationarity of the speech signal must be taken into account, and perceptually relevant evaluation of the signal estimate should be considered. For economic time-series, data scarcity is a critical issue. Generally, for any specific application domain the questions of model set selection and model validation can be more readily addressed.

1.5.7 Summary of Contributions

The contributions of this thesis are as follows:

[6] Although it was developed independently, the prediction error form of the dual EKF also bears similarity to the method proposed by Matthews [52] for training recurrent networks with clean data.

1. *Unified theoretical framework.* The relationship between several different cost functions and algorithms are shown within a probabilistic framework under a Gaussian assumption. New cost functions are developed that have not been previously explored in the literature.

2. *Nonlinear methods.* New algorithms are proposed that are applicable for both linear and nonlinear model structures. These algorithms extend the range of existing linear methods to include new cost functions, and expand the application domain to include nonlinear time-series models.

3. *Sequential methods.* The new algorithms provide estimates of the signal and model sequentially. This gives them the flexibility of being applicable in both on-line and off-line settings.

4. *Unified algorithmic framework.* A variety of approaches to the dual estimation problem can be unified algorithmically by showing how they are implemented as specific members of a *dual EKF* family of algorithms.

5. *Noise variance estimation.* Novel methods of estimating the process and measurement noise variances are investigated in the context of the dual estimation algorithms.

6. *Experimental comparisons.* Experiments on several different classes of data are included in order to compare the advantages and disadvantages of the various approaches. Linear models are also compared with nonlinear models (exemplified by feedforward neural networks).

7. *Applications.* Example applications of the dual EKF algorithms are provided to demonstrate their use on real-world data, and to address some of the practical considerations that arise for different classes of data.

As stated previously, this author has written several papers on the dual EKF with Dr. Eric Wan. This thesis extends that work by deepening the theoretical foundations of the approach, and broadening the algorithm to encompass a number of heretofore disparate methods. Experimental work is also extended to include data from the domains of speech, econometrics, and geophysics.

Chapter 2

Cost Functions: A Probabilistic Perspective

2.1 Overview

This chapter considers the dual estimation problem from a probabilistic perspective. This perspective is used to show the relationship between many of the algorithms mentioned in Chapter 1, and to generate several new cost functions.

Section 2.2 motivates the maximum *a posteriori* (MAP) approach to dual estimation, the central component of which is the joint conditional density of the signal and weights. Section 2.3 uses this density as the theoretical foundation for developing several different cost functions. In Section 2.4, the expansion of this joint density into a marginal form is considered. The relationship between the joint and marginal forms is used to provide an understanding of the relationship between a variety of cost functions, some of which are exemplified by existing algorithms, and some of which have not been explored in the current literature.

For the sake of conceptual simplicity, the derivations in this chapter are based on the off-line problem of estimating the signal and model from a set of N noisy observations, $\{y_k\}_1^N$. This allows the cost functions to be written in the familiar form as a sum of quadratic (and other) terms. While the algorithms in Chapter 3 will be based directly on these cost functions, they are recursive methods using on-line interpretations of the costs.

This chapter makes explicit use of the Gaussian assumption placed on the noise processes in Section 1.2. The Gaussian noise assumption greatly facilitates the derivation of the necessary cost functions. However, while the cost functions derived in this chapter rely on this assumption for their theoretical justification, the corresponding algorithms in the next chapter are not so limited in their scope. This will be demonstrated by the examples provided at the end of this thesis, some of which involve obviously non-Gaussian data.

Furthermore, the basic relationship between two main classes of algorithms shown in the next section does not rely on a Gaussian assumption. Only when the relevant probabilistic quantities are translated into cost functions and algorithms is a Gaussian assumption employed.

2.2 Bayesian Estimation for Noisy Time-Series

2.2.1 Characterizing the Data

The data are assumed to be generated according to Equation 1.1:

$$x_k = f\left(x_{k-1}, \ldots x_{k-M}, \mathbf{w}\right) + v_k$$
$$y_k = x_k + n_k, \qquad \forall k \in \{1 \ldots N\}.$$

With only $\{y_k\}_1^N$ available, the dual estimation problem is to find estimates $\{\hat{x}_k\}_1^N$ and $\hat{\mathbf{w}}$ of the signal and weights that are in some sense optimal. All of the statistical information contained in the data $\{y_k\}_1^N$ about the signal and parameters is embodied by the joint conditional probability density of the signal $\{x_k\}_1^N$ and weights $\mathbf{w}$, 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 $\{x_k\}_1^N$ and $\{y_k\}_1^N$, respectively. The joint conditional density function is written as:

$$\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}\left(\mathbf{X} = \mathbf{x}_1^N, \mathbf{W} = \mathbf{w} \middle| \mathbf{Y} = \mathbf{y}_1^N\right), \tag{2.1}$$

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 $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$.

An alternative view of generating data according to Equation 1.1 is *sampling* from the distribution given by $\rho_{\mathbf{x}_1^N \mathbf{w} \mathbf{y}_1^N}$. One sample includes the specific data $\{y_k\}_1^N$, as well as the unobserved signal $\{x_k\}_1^N$ and the unknown parameters $\mathbf{w}$. Because $\{y_k\}_1^N$ is the only observable part of the sample, the values of $\{x_k\}_1^N$ and $\mathbf{w}$ can only be estimated by using the knowledge embodied by $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$.

Given the data $\{y_k\}_1^N$, a dual estimation procedure will produce estimates $\{\hat{x}_k\}_1^N$ and $\hat{\mathbf{w}}$. Because $\{y_k\}_1^N$ were drawn according to the random vector $\mathbf{Y}$, it follows that $\{\hat{x}_k\}_1^N$ and $\hat{\mathbf{w}}$ are effectively drawn from the distributions on the random vectors $\hat{\mathbf{X}}$ and $\hat{\mathbf{W}}$, where these random vectors are functions of the random vector $\mathbf{Y}$. The nature of these functions are determined by the estimation procedure.

2.2.2 Expected Loss

A particular sample of data and a particular choice of estimator will produce the errors $\{\hat{x}_k - x_k\}_1^N$ and $\hat{\mathbf{w}} - \mathbf{w}$. A good estimator should generate small errors. To quantify this idea, a *loss function*

combines these error vectors to compute a scalar measure of quality. Examples of loss functions include inner products, or various norms with different weightings between the signal error and weight errors, or between different elements of these errors.

The loss function describes the quality of the estimates produced for a given sample of data drawn according to $\rho_{\mathbf{x}_1^N \mathbf{w} \mathbf{y}_1^N}$. However, the fact that $\{x_k\}_1^N$ and $\mathbf{w}$ are unobserved prevents direct computation of the loss. Instead, the density $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$ is used to compute the conditional expectation of this loss given the data:

$$E_{\mathbf{X}\mathbf{W}|\mathbf{Y}}[L(\{x_k - \hat{x}_k\}_1^N, \mathbf{w} - \hat{\mathbf{w}})|\{y_k\}_1^N], \tag{2.2}$$

where $L(\cdot)$ is a *loss function* of the errors in the signal and weight estimates.

Furthermore, one is generally interested in an estimator that *generalizes* to new data. That is, if *new* samples $\{y_k\}_1^N$, $\{x_k\}_1^N$ and $\mathbf{w}$ are drawn from $\rho_{\mathbf{x}_1^N \mathbf{w} \mathbf{y}_1^N}$, the same estimation procedure should produce $\{\hat{x}_k\}_1^N$ and $\hat{\mathbf{w}}$ with a low loss function value. In other words, whatever the choice of loss function, one is interested in minimizing the *expected loss*, where the expectation is taken over possible values of $\mathbf{w}$, $\{x_k\}_1^N$, and $\{y_k\}_1^N$ according to the density $\rho_{\mathbf{x}_1^N \mathbf{w} \mathbf{y}_1^N}$. This expected loss can be written as:

$$E_{\mathbf{Y}}\Big[E_{\mathbf{X}\mathbf{W}|\mathbf{Y}}\big[L(\{x_k - \hat{x}_k\}_1^N, \mathbf{w} - \hat{\mathbf{w}})\big|\{y_k\}_1^N\big]\Big], \tag{2.3}$$

This expression is often called the *Bayes risk*, and the values of $\{x_k\}_1^N$ and $\mathbf{w}$ for which it is minimized are the *Bayes estimates*. Clearly, what these estimates are will depend on the specific loss function $L(\cdot)$ that is chosen.

If a quadratic loss is chosen, the expected cost is minimized by the *minimum mean squared error* (MMSE) solution, given by the conditional mean $E[\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N]$. An absolute value loss function produces an estimate equal to the median value of the joint conditional density $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$, called the *minimax*[1] solution. A loss that is one everywhere and zero in a small region around the true values of $\{x_k\}_1^N$ and $\mathbf{w}$ corresponds to the *maximum a posteriori* (MAP) solution, which maximizes the posterior (or conditional) density $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$. For derivations and additional discussion on this topic see page 4 of [43].

When $\rho_{\mathbf{x}_1^N \mathbf{w}}$ is unimodal and symmetric about its mean, the Bayes estimate is the same for a broad class of loss functions (for details, see [32]). In particular, when $\rho_{\mathbf{x}_1^N \mathbf{w}}$ is Gaussian, the MAP, MMSE, and minimax estimates are all the same; this equivalence will hold when the noise processes are Gaussian *and* the system function $f(\cdot)$ is linear.

[1] The minimax estimate is so called because it minimizes the maximum value of the error.

2.2.3 MAP Approach to Dual Estimation

The MAP estimate is also sometimes used in applications where the choice of a suitable loss function is not clear [17]. Furthermore, the relationship between the various approaches in the literature is most apparent when viewed from a MAP perspective. In the dual estimation context, the MAP estimation approach consists of the following optimization problem:

$$(\hat{\mathbf{x}}_1^N, \hat{\mathbf{w}}) = \arg\max_{\mathbf{x}_1^N, \mathbf{w}} \rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N} \tag{2.4}$$

This formulation of the problem is the focus of this thesis.

By and large, the literature can be divided into two basic classes of algorithms. The first, referred to herein as *joint estimation methods*, attempt to maximize $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$ directly. This approach will be described in Section 2.3. The second class of methods, which will be referred to as *marginal estimation methods*, operate by expanding the joint density as:

$$\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N} = \rho_{\mathbf{x}_1^N | \mathbf{w} \mathbf{y}_1^N} \cdot \rho_{\mathbf{w} | \mathbf{y}_1^N} \tag{2.5}$$

and maximizing the two terms separately. The marginal estimation approach will be described in Section 2.4.

2.3 Joint Estimation of Signal and Weights

The MAP approach to the dual estimation problem is to maximize the joint conditional probability density $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$ of the signal $\{x_k\}_1^N$ and weights $\mathbf{w}$, given the noisy data $\{y_k\}_1^N$. Again, estimation schemes that deal with this quantity are referred to as *joint estimation* methods.

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

$$\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{\mathbf{x}_1^N \mathbf{w}}}{\rho_{\mathbf{y}_1^N}}. \tag{2.6}$$

Although $\{y_k\}_1^N$ is *statistically* dependent on $\{x_k\}_1^N$ and $\mathbf{w}$, the prior $\rho_{\mathbf{y}_1^N}$ is nonetheless *functionally* independent of $\{x_k\}_1^N$ 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. The first term $\rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}}$ represents the joint likelihood function of the signal and weights, while the second term $\rho_{\mathbf{x}_1^N \mathbf{w}}$ represents the prior information about the relationship between the signal and the weights. The numerator can be expanded further as:

$$\rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{\mathbf{x}_1^N \mathbf{w}} = \rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{\mathbf{x}_1^N | \mathbf{w}} \cdot \rho_{\mathbf{w}}. \tag{2.7}$$

If no prior information is available on the weights, $\rho_{\mathbf{w}}$ can be dropped, leaving the maximization of

$$\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \mathbf{w}} = \rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{\mathbf{x}_1^N | \mathbf{w}} \tag{2.8}$$

Let **a** and **b** represent two jointly distributed Gaussian random variables. Then:

$$\rho_{a|b}(\mathbf{a}=a|\mathbf{b}=b)=\frac{1}{\sqrt{2\pi\sigma_{a|b}^2}}\exp\left(-\frac{(a-E[a|b])^2}{2\sigma_{a|b}^2}\right),$$

where $E[a|b]$ is the conditional mean of **a** given $\mathbf{b}=b$, and $\sigma_{a|b}^2$ is the conditional variance.

Formula 2.1: The general form of a Gaussian conditional density $\rho_{\mathbf{A}|\mathbf{B}}$

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

2.3.1 White Noise Case

If v_k and n_k are both zero-mean white Gaussian noise processes, then the two terms of Equation 2.8 can be evaluated (as shown in Appendix A) to give:

$$\begin{aligned}\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}} =& \frac{1}{\sqrt{(2\pi)^N(\sigma_n^2)^N}}\exp\left(-\sum_{k=1}^{N}\frac{(y_k-x_k)^2}{2\sigma_n^2}\right)\\ &\cdot\frac{1}{\sqrt{(2\pi)^N(\sigma_v^2)^N}}\exp\left(-\sum_{k=1}^{N}\frac{(x_k-x_k^-)^2}{2\sigma_v^2}\right),\end{aligned} \tag{2.9}$$

$$\begin{aligned}\text{where}\quad x_k^- &\triangleq E[x_k|\{x_t\}_1^{k-1},\mathbf{w}]\\ &= f(x_{k-1},\ldots,x_{k-M},\mathbf{w}).\end{aligned}$$

Here we have used the structure given in Equation 1.1 to compute the prediction x_k^- using the model $f(\cdot,\mathbf{w})$ and only the past M values of the series.

After taking the logarithm, the corresponding cost function can be seen to be:

$$\begin{aligned}J=\sum_{k=1}^{N}\Bigg(&\log(2\pi\sigma_n^2)+\frac{(y_k-x_k)^2}{\sigma_n^2}+\\ &\log(2\pi\sigma_v^2)+\frac{(x_k-x_k^-)^2}{\sigma_v^2}\Bigg).\end{aligned} \tag{2.10}$$

This cost function can be minimized with respect to any of the unknown quantities. However, if the noise variances are known, then only $\{x_k\}_1^N$ and $\mathbf{w}$ need to be estimated. 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:

$$J^j(\mathbf{x}_1^N,\mathbf{w})=\sum_{k=1}^{N}\left(\frac{(y_k-x_k)^2}{\sigma_n^2}+\frac{(x_k-x_k^-)^2}{\sigma_v^2}\right). \tag{2.11}$$

The first term is a soft constraint keeping $\{x_k\}_1^N$ 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

signal estimates and model estimates mutually consistent with the AR structure. This constraint will be strong when the signal is highly deterministic (*i.e.*, σ_v^2 is small). Although the first term is a function of $\{x_k\}_1^N$ alone, the second term represents a strong coupling between $\{x_k\}_1^N$ and $\mathbf{w}$, through $x_k^- = f(x_{k-1} \ldots x_{k-M}, \mathbf{w})$.

$J^j(\mathbf{x}_1^N, \mathbf{w})$ should be minimized with respect to both $\{x_k\}_1^N$ and $\mathbf{w}$ to find the estimates which 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 $\{x_k\}_1^N$ and $\mathbf{w}$. As shown in Appendix G, the EIV approach tries to minimize this same cost in an iterative framework. The joint EKF algorithm mentioned in Section 1.4 attempts to *sequentially* estimate the signal and weights by combining them into a single (joint) state vector (see Chapter 3). However, the resulting system of state-space equations is highly nonlinear, even for linear models. Several authors have reported convergence problems with this approach [45, 61].

Decoupling with Direct Substitution

An alternative way of dealing with this sort of multivariate optimization problem is by optimizing one variable at a time while the other variable is fixed, and alternating. This approach, which effectively decouples the optimization problem, is exemplified by the (iterative) errors-in-variables algorithm, and by the (sequential) dual EKF family of algorithms.

Signal Estimation

To minimize $J^j(\mathbf{x}_1^N, \mathbf{w})$ with respect to the signal, the cost is evaluated using the current estimate $\hat{\mathbf{w}}$ of the weights to generate the predictions. The simplest approach is to substitute the predictions $\hat{x}_k^- \triangleq f(x_{k-1}, \ldots, x_{k-M}, \hat{\mathbf{w}})$ directly into Equation 2.11:

$$J^j(\mathbf{x}_1^N, \hat{\mathbf{w}}) = \sum_{k=1}^{N} \left(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - \hat{x}_k^-)^2}{\sigma_v^2} \right). \tag{2.12}$$

This cost function is then minimized with respect to $\{x_k\}_1^N$.

Weight Estimation

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 $\{\hat{x}_k\}_1^N$ and the associated (redefined) predictions $\hat{x}_k^- \triangleq f(\hat{x}_{k-1}, \ldots, \hat{x}_{k-M}, \mathbf{w})$. Again, this results in a straightforward substitution in Equation 2.11:

$$J^j(\hat{x}, \mathbf{w}) = \sum_{k=1}^{N} \left(\frac{(y_k - \hat{x}_k)^2}{\sigma_n^2} + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2} \right). \tag{2.13}$$

If the current signal estimate $\hat{x}_k$ is taken to be a recursive function of the weights, then both terms in the above cost are minimized with respect to $\mathbf{w}$.

Otherwise, however, the first term is independent of the weights $\mathbf{w}$, and only the second term is minimized. Here, only $\hat{x}_k^-$ is a function of the weights:

$$J_i^j(\hat{\mathbf{x}}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2}. \tag{2.14}$$

This is essentially a type of prediction error cost, where the model is trained to predict the *estimated* time-series. 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.

Variance Estimation

When the variances are unknown, they must be estimated as well. To minimize the joint cost function with respect to the noise variances, the full cost J^j is evaluated using the current signal estimate $\{\hat{x}_k\}_1^N$, weight estimates $\hat{\mathbf{w}}$, and the associated (redefined) predictions $\hat{x}_k^- \triangleq f(\hat{x}_{k-1}, ..., \hat{x}_{k-M}, \hat{\mathbf{w}})$. This results in a straightforward substitution in Equation 2.10:

$$\begin{aligned} J^j(\sigma^2) = \sum_{k=1}^{N} \Bigg(& \log(2\pi\sigma_n^2) + \frac{(y_k - \hat{x}_k)^2}{\sigma_n^2} \\ & + \log(2\pi\sigma_v^2) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2} \Bigg). \end{aligned} \tag{2.15}$$

This cost function can be minimized with respect to either σ_v^2 or σ_n^2 by using the current estimates of the signal and weights.

Notice that the log terms are necessary for keeping the variance estimates small, because the quadratic terms go to zero as the variances go to infinity. Also, the estimates $\hat{x}_k$ and predictions $\hat{x}_k^-$ are themselves functions of the variances, so the numerators in the quadratic terms are also minimized with respect to σ_v^2 and σ_n^2.

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. Sequential minimization of the costs in Equations 2.12 - 2.15 is performed by a two-observation form of the dual EKF algorithm (Nelson 1998 [60]); the errors-in-variables method performs a batch-style minimization. Details on these algorithms are provided in Chapter 3.

Error Coupling

While clearly justified, the above direct substitution approach fails to take advantage of the information which is available about the *errors* in the estimates at each step of the optimization.

Signal Estimation

From the standpoint of signal estimation alone, minimizing $J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$ in Equation 2.12 is not the best approach because the error associated with $\hat{\mathbf{w}}$ has not been taken into account. To see this, consider rewriting Equation 1.1 in terms of $\hat{\mathbf{w}}$:

$$\begin{aligned} x_k &= f\big(x_{k-1}, \ldots x_{k-M}, \hat{\mathbf{w}}\big) + \tilde{f}_k + v_k \\ y_k &= x_k + n_k, \qquad\qquad \forall k \in \{1 \ldots N\} \\ \text{where} \quad \tilde{f}_k &\triangleq f\big(x_{k-1}, \ldots x_{k-M}, \mathbf{w}\big) - f\big(x_{k-1}, \ldots x_{k-M}, \hat{\mathbf{w}}\big). \end{aligned} \tag{2.16}$$

Note that this representation is exactly equivalent to Equation 1.1, except that the time-series is now written as a function of $\hat{\mathbf{w}}$ instead of $\mathbf{w}$. The error due to $\hat{\mathbf{w}}$ is accounted for by introducing $\tilde{f}_k$. This reformulation allows for the evaluation of $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \hat{\mathbf{w}}}$ instead of $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \mathbf{w}}$. Because $\hat{\mathbf{w}}$ is available (and not $\mathbf{w}$), maximizing $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \hat{\mathbf{w}}}$ should produce better signal estimates; this in turn, should produce better weight estimates and allow for faster overall convergence. Furthermore, as the weights estimates converge, $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \hat{\mathbf{w}}}$ will converge to $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \mathbf{w}}$, giving the desired signal estimates.

The utility of this reformulation is realized by assuming that the dynamics error, $\tilde{f}_k$, is a zero-mean Gaussian process. When $f(\cdot)$ is a linear function, the dynamics error is a linear function of the error in the weight estimates: *i.e.*, $\tilde{f}_k = \mathbf{x}_{k-1}^T \cdot \tilde{\mathbf{w}}$, where $\tilde{\mathbf{w}} \triangleq \mathbf{w} - \hat{\mathbf{w}}$. If $\tilde{\mathbf{w}}$ is distributed as a zero-mean Gaussian, then so is $\tilde{f}_k$. If $f(\cdot)$ is nonlinear, this represents an approximation.

Assuming $\tilde{\mathbf{w}}$ is zero-mean is equivalent to assuming $\hat{\mathbf{w}}$ is an *unbiased* ($E[\tilde{\mathbf{w}}] = 0$) estimate. In order for $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \hat{\mathbf{w}}}$ to converge to $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \mathbf{w}}$ it is also necessary that $\hat{\mathbf{w}}$ be a *consistent* ($Cov[\tilde{\mathbf{w}}] \to 0$) estimator.

The cost associated with $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \hat{\mathbf{w}}}$ is derived using $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \hat{\mathbf{w}}} = \rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \hat{\mathbf{w}}} \cdot \rho_{\mathbf{x}_1^N | \hat{\mathbf{w}}}$ (*cf.* Equation 2.8) in conjunction with Equation 2.16. The cost is:

$$J^{ec}(\mathbf{x}_1^N) = \sum_{k=1}^{N} \left(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - \hat{x}_k^-)^2}{\sigma_{\tilde{f},k}^2 + \sigma_v^2} + \log(2\pi(\sigma_{\tilde{f},k}^2 + \sigma_v^2)) \right), \tag{2.17}$$

where $\sigma_{\tilde{f},k}^2$ is the variance of the error, $\tilde{f}_k$, in the prediction due to $\hat{\mathbf{w}}$. Predictions are given by $\hat{x}_k^- = f\big(x_{k-1}, \ldots x_{k-M}, \hat{\mathbf{w}}\big)$, and the prediction error $\tilde{x}_k^- = x_k - \hat{x}_k^-$ includes the usual process noise v_k as well as the error $\tilde{f}_k$, giving it a variance $\sigma_{\tilde{f}}^2 + \sigma_v^2$. The error in the weights is thus accounted

for by an adjustment in the prediction error variance. Note that because $\tilde{f}_k$ is independent of the *current* signal value x_k, this log term is neglected by algorithms that operate sequentially on the data, such as the Kalman filter.

In any case, all of the dual estimation algorithms in the literature that have a distinct signal estimation step (*e.g.*, EIV, EM, RPE) have until now minimized the cost in Equation 2.12. Some authors have suggested increasing the process noise variance σ_v^2 to account for model errors [78]. However, the more rigorous approach to incorporating the model error statistics – represented by Equation 2.17 – has not been investigated elsewhere.

Weight Estimation

For weight estimation, the cost functions in Equations 2.13 and 2.14 implicitly assume that the signal estimates $\{\hat{x}_k\}_1^N$ are exact. As in the above discussion on signal estimation, faster convergence might be obtained if the error associated with $\{\hat{x}_k\}_1^N$ is accounted for by maximizing the density $\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N|\mathbf{w}}$. Again, $\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N|\mathbf{w}}$ will converge to $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}}$ as the signal estimates converge.

In order to evaluate $\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N|\mathbf{w}}$, Equation 1.1 is again rewritten: this time as

$$\begin{aligned} \hat{x}_k &= f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \mathbf{w}\big) + \tilde{f}_k - \tilde{x}_k + v_k \\ y_k &= \hat{x}_k + \tilde{x}_k + n_k, \qquad\qquad \forall k \in \{1 \ldots N\} \\ &\text{where} \quad \tilde{f}_k \triangleq f\big(x_{k-1}, \ldots x_{k-M}, \mathbf{w}\big) - f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \mathbf{w}\big), \\ &\text{and} \quad \tilde{x}_k \triangleq x_k - \hat{x}_k. \end{aligned} \tag{2.18}$$

Here, $\tilde{f}_k$ and $\tilde{x}_k$ are assumed to be approximately Gaussian, and zero-mean under the assumption that $\hat{x}_k$ is unbiased. Convergence of $\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N|\mathbf{w}}$ to $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}}$ is provided by assuming that $\hat{x}_k$ is consistent. Using the above representation and the expansion $\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N|\mathbf{w}} = \rho_{\mathbf{y}_1^N|\hat{\mathbf{x}}_1^N \mathbf{w}} \cdot \rho_{\hat{\mathbf{x}}_1^N|\mathbf{w}}$ results in the following cost function:

$$\begin{aligned} J^{ec}(\mathbf{w}) = \sum_{k=1}^{N} \Bigg(& \log(2\pi\sigma_{e_k}^2) + \frac{(y_k - \hat{x}_k)^2}{\sigma_{e_k}^2} \\ & + \log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} \Bigg), \end{aligned} \tag{2.19}$$

where $\sigma_{e_k}^2$ is the variance of $e_k = (y_k - \hat{x}_k)$, which contains both n_k and the signal estimate error $\tilde{x}_k$. Hence, $\sigma_{e_k}^2 = \sigma_{\tilde{x}_k}^2 + \sigma_n^2$. Predictions are given by $\hat{x}_k^- = f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \mathbf{w}\big)$, and the variance of the prediction error $\tilde{\hat{x}}_k = (\hat{x}_k - \hat{x}_k^-)$ is denoted by g_k. The variances $\sigma_{e_k}^2$ and g_k replace σ_n^2 and σ_v^2, respectively, in Equation 2.13. Also, note that the log terms have not been dropped from this cost, because both $\sigma_{e_k}^2$ and g_k are functions of $\mathbf{w}$.

If the estimates $\{\hat{x}_k\}_1^N$ are taken to be independent of the weights, then the first two terms can be dropped, and the cost reduces to:

$$J_i^{ec}(\mathbf{w}) = \sum_{k=1}^{N}\left(\log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k}\right), \tag{2.20}$$

where only $\hat{x}_k^-$ is a function of $\mathbf{w}$. As in the cost of Equation 2.14, this cost is not directly dependent on the data $\{y_k\}_1^N$, and when used in the dual estimation setting, runs the risk of producing results that are consistent with $\{\hat{x}_k\}_1^N$, but not with the data. However, while the cost $J^j(\hat{\mathbf{x}}, \mathbf{w})$ in Equation 2.14 effectively treats the estimates $\hat{x}_k$ as if they are the clean signal, Equation 2.20 avoids this pitfall by accounting for the error in $\hat{x}_k$ through g_k.

Variance Estimation

The errors in the signal and weight estimates can also be accounted for during estimation of the noise variances. Here, the idea is to maximize $\rho_{\hat{\mathbf{x}}_1^N \hat{\mathbf{w}}|\mathbf{y}_1^N}$ under the assumption that it will converge to the density $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$ as the signal and weight estimates converge. The autoregression is written as in Equation 2.18, except that the dynamics error is defined differently, as

$$\tilde{f}_k \triangleq f\big(x_{k-1}, \ldots x_{k-M}, \mathbf{w}\big) - f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \hat{\mathbf{w}}\big). \tag{2.21}$$

The resulting cost function:

$$\begin{aligned} J^{ec}(\sigma^2) = \sum_{k=1}^{N}\Bigg(& \log(2\pi\sigma_{e_k}^2) + \frac{(y_k - \hat{x}_k)^2}{\sigma_{e_k}^2} \\ & + \log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k}\Bigg), \end{aligned} \tag{2.22}$$

is identical to that in Equation 2.19 except that predictions here are given by $\hat{x}_k^- = f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \hat{\mathbf{w}}\big)$ and g_k is adjusted accordingly. The argument σ^2 is used to indicate that the cost is minimized with respect to either σ_v^2 or σ_n^2.

Whether for signal estimation or weight estimation, the above "error coupling" cost functions have the potential of offering faster convergence to a maximum of $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$. However, they require the approximation of $\tilde{f}_k$ as a zero-mean Gaussian random variable whose variance goes to zero asymptotically. The effect of this approximation will depend on the specific data being considered; however, in Chapter 4, the error-coupled cost is shown not to perform as well, in general, as the other dual EKF costs. Algorithms for minimizing the costs in Equations 2.17, 2.19, and 2.20 are provided in Chapter 3.

The following section addresses the dual estimation problem when the measurement noise is *not* white. Noise with correlation between samples is usually referred to as *colored* noise. The development for the colored noise case mostly parallels the white noise case, although the resulting cost functions are somewhat different.

2.3.2 Colored Noise Case

The additive noise $\{n_k\}_1^N$ is generally assumed in this chapter to be Gaussian with an autocorrelation function that is known within a scalar multiple. The simplest case, addressed in the previous section, is when the noise is white with possibly unknown scalar variance, σ_n^2. When the noise is colored, the knowledge of its autocorrelation can be encoded by writing the noise as a linear AR process:

$$n_k = \sum_{i=1}^{M_n} w_n^{(i)} \cdot n_{k-i} + v_{n,k}, \tag{2.23}$$

where the parameters $w_n^{(i)}$ are assumed to be known, and $v_{n,k}$ 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. If the system (*i.e.*, $w_n^{(i)}$) were *not* known, the signal estimation problem would be equivalent to single-sensor blind signal separation. This remains a challenging area for future research, and will not be considered here.

Because n_k can be viewed as a second signal, it should be estimated on 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 comprised of elements in $\{n_k\}_1^N$) instead of $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$. We can write this as:

$$\rho_{\mathbf{x}_1^N \mathbf{n}_1^N \mathbf{w}|\mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N \mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{w}}}{\rho_{\mathbf{y}_1^N}}, \tag{2.24}$$

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.

However, writing the expansion

$$\rho_{\mathbf{y}_1^N \mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}} = \rho_{\mathbf{y}_1^N|\mathbf{x}_1^N \mathbf{n}_1^N \mathbf{w}} \cdot \rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}}, \tag{2.25}$$

exposes $\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N \mathbf{n}_1^N \mathbf{w}}$ as an impulse function at the constraint $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$. Hence, $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}}$ is singular, so maximizing it is equivalent to maximizing $\rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}}$ subject to $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$. Furthermore, $\rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}}$ can be written as:

$$\rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}} = \rho_{\mathbf{x}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{n}_1^N|\mathbf{w}}, \tag{2.26}$$

because the signal and noise are assumed to be mutually independent. If the process noise terms v_k and $v_{n,k}$ are zero-mean Gaussian white noise, then:

$$\begin{aligned}\rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}} = &\frac{1}{\sqrt{(2\pi)^N(\sigma_v^2)^N}} \exp\left(-\sum_{k=1}^{N} \frac{(x_k - x_k^-)^2}{2\sigma_v^2}\right) \\ &\cdot \frac{1}{\sqrt{(2\pi)^N(\sigma_{v_n}^2)^N}} \exp\left(-\sum_{k=1}^{N} \frac{(n_k - n_k^-)^2}{2\sigma_{v_n}^2}\right),\end{aligned} \tag{2.27}$$

where $n_k^- = \sum_{i=1}^{M_n} w_n^{(i)} \cdot n_{k-i}$. The corresponding cost is simply:

$$\begin{aligned}J_c^j = \sum_{k=1}^{N} \Bigg(&\log(2\pi\sigma_v^2) + \frac{(x_k - x_k^-)^2}{\sigma_v^2} \\ &\log(2\pi\sigma_{v_n}^2) + \frac{(n_k - n_k^-)^2}{\sigma_{v_n}^2}\Bigg),\end{aligned} \tag{2.28}$$

where, as before, $x_k^- = f(x_{k-1}, \ldots, x_{k-M}, \mathbf{w})$.

For estimation of $\{x_k\}_1^N$, $\{n_k\}_1^N$, or $\mathbf{w}$, the log terms can be dropped from the cost function, leaving:

$$J_c^j(\mathbf{x}_1^N, \mathbf{n}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \left(\frac{(x_k - x_k^-)^2}{\sigma_v^2} + \frac{(n_k - n_k^-)^2}{\sigma_{v_n}^2} \right), \tag{2.29}$$

minimized subject to $\mathbf{x}_1^N + \mathbf{n}_1^N = \mathbf{y}_1^N$. Comparing this cost function to that of Equation 2.11, note that a term involving the colored measurement noise has been included, while the term involving the error $(y_k - x_k)^2$ has been replaced by the hard constraint $y_k = x_k + n_k$.

The above cost should be minimized with respect to $\{x_k\}_1^N$, $\mathbf{w}$, as well as $\{n_k\}_1^N$ to solve the dual estimation problem. Again, one approach to this highly coupled optimization task is offered by the joint EKF algorithm, in which all of the unknowns are combined in a joint state vector. The colored noise version of the joint EKF will be shown in Chapter 3.

Decoupling with Direct Substitution – Colored Noise

As in the white noise case, the joint estimation can be decoupled by minimizing one variable at a time, while the other variables are fixed. Note, the colored noise case also requires the explicit estimation of the measurement noise $\mathbf{n}_1^N$, in addition to $\mathbf{x}_1^N$ and $\mathbf{w}$.

Signal and Colored Noise Estimation

Because of the hard constraint $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$, the signal and noise are tightly coupled. In fact, each one can be viewed as a function of the other. Therefore, $J_c^j(\mathbf{x}_1^N, \mathbf{n}_1^N, \mathbf{w})$ in Equation 2.29

should be minimized with respect to the signal and noise simultaneously, by evaluating it at the current weight estimate $\hat{\mathbf{w}}$. That is,

$$J_c^j(\mathbf{x}_1^N, \mathbf{n}_1^N, \hat{\mathbf{w}}) = \sum_{k=1}^{N} \left(\frac{(x_k - \hat{x}_k^-)^2}{\sigma_v^2} + \frac{(n_k - n_k^-)^2}{\sigma_{v_n}^2} \right), \tag{2.30}$$

where $\hat{x}_k^- = f(x_{k-1}, \ldots, x_{k-M}, \hat{\mathbf{w}})$, as before, and the predictions $n_k^- = \sum_{i=1}^{M_n} w_n^{(i)} \cdot n_{k-i}$ are made according to the known noise model. This cost function is minimized subject to the constraint, $y_k = x_k + n_k$, to produce signal and noise estimates.

Weight Estimation – Colored Noise

Similarly, the weights can be estimated by minimizing $J_c^j(\mathbf{x}_1^N, \mathbf{n}_1^N, \mathbf{w})$, evaluated using the current estimates, $\{\hat{x}_k\}_1^N$ and $\{\hat{n}_k\}_1^N$, of the signal and noise. The cost function is:

$$J_c^j(\hat{\mathbf{x}}_1^N, \hat{\mathbf{n}}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \left(\frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2} + \frac{(\hat{n}_k - \hat{n}_k^-)^2}{\sigma_{v_n}^2} \right), \tag{2.31}$$

where the predictions are defined as: $\hat{x}_k^- = f(\hat{x}_{k-1}, \ldots, \hat{x}_{k-M}, \mathbf{w})$, and $\hat{n}_k^- = \sum_{i=1}^{M_n} w_n^{(i)} \cdot \hat{n}_{k-i}$. If the signal estimates, $\hat{x}_k$, are recursive functions of the weights, then the noise estimates are as well, by way of the constraint $y_k = \hat{x}_k + \hat{n}_k$. Note, however, that if the signal estimates are not taken to depend on the weights, then the hard constraint becomes inconsequential, and only the first term in the above cost is used. The cost reduces to Equation 2.14 on page 25.

Variance Estimation – Colored Noise

To minimize the joint cost function with respect to the noise variances, J_c^j in Equation 2.28 on the preceding page is evaluated using the current signal and noise estimates, weight estimates $\hat{\mathbf{w}}$, and the associated (redefined) predictions $\hat{x}_k^- \triangleq f(\hat{x}_{k-1}, \ldots, \hat{x}_{k-M}, \hat{\mathbf{w}})$. Again, this results in a straightforward substitution:

$$\begin{aligned} J_c^j(\sigma^2) = \sum_{k=1}^{N} \Bigg(& \log(2\pi\sigma_v^2) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2} \\ & + \log(2\pi\sigma_{v_n}^2) + \frac{(\hat{n}_k - \hat{n}_k^-)^2}{\sigma_{v_n}^2} \Bigg). \end{aligned} \tag{2.32}$$

This cost function can be minimized with respect to σ_v^2, $\sigma_{v_n}^2$, or both.

All of the costs in Equations 2.30–2.32 can be minimized sequentially with the dual EKF algorithm, developed in Chapter 3, beginning on page 89.

Error Coupling – Colored Noise

As discussed in the white noise case, faster convergence to a minimum of J_c^j may be possible by using cost functions that take into account the errors in each of the estimates.

Signal and Colored Noise Estimation

The density $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N \mathbf{n}_1^N|\hat{\mathbf{w}}}$ provides an approach to estimating the signal and noise that takes into account the errors associated with $\hat{\mathbf{w}}$. This density is expanded as:

$$\rho_{\mathbf{y}_1^N \mathbf{x}_1^N \mathbf{n}_1^N|\hat{\mathbf{w}}} = \rho_{\mathbf{y}_1^N|\mathbf{x}_1^N \mathbf{n}_1^N \hat{\mathbf{w}}} \cdot \rho_{\mathbf{x}_1^N|\hat{\mathbf{w}}} \cdot \rho_{\mathbf{n}_1^N|\hat{\mathbf{w}}}, \tag{2.33}$$

where the first term is singular at the constraint $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$. Therefore, $\rho_{\mathbf{x}_1^N|\hat{\mathbf{w}}} \cdot \rho_{\mathbf{n}_1^N|\hat{\mathbf{w}}}$ can be maximized subject to the same constraint as before. This is evaluated using the alternative form of the AR model given in Equation 2.16 on page 26. Defining predictions $\hat{x}_k^- = f(x_{k-1}, ...x_{k-M}, \hat{\mathbf{w}})$ with error variance $(\sigma_{\hat{f}}^2 + \sigma_v^2)$ thus yields the cost function:

$$J_c^{ec}(\mathbf{x}_1^N, \mathbf{n}_1^N) = \sum_{k=1}^{N} \left(\frac{(x_k - \hat{x}_k^-)^2}{(\sigma_{\hat{f}}^2 + \sigma_v^2)} + \frac{(n_k - n_k^-)^2}{\sigma_{v_n}^2} + \log(2\pi(\sigma_{\hat{f}}^2 + \sigma_v^2)) \right), \tag{2.34}$$

which is minimized with respect to $\mathbf{x}_1^N$ and $\mathbf{n}_1^N$ subject to $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$. Note the similarity between Equations 2.34 and 2.17 on page 26, where the term involving the observation y_k has been replaced by the hard constraint, and an additional term for noise estimation is included. The statistics of the noise n_k are not affected by the weight estimates, $\hat{\mathbf{w}}$.

Weight Estimation – Colored Noise

For weight estimation, accounting for the error in the signal and noise estimates requires looking at the probability density function $\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N \hat{\mathbf{n}}_1^N|\mathbf{w}}$, which is expanded as:

$$\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N \hat{\mathbf{n}}_1^N|\mathbf{w}} = \rho_{\mathbf{y}_1^N|\hat{\mathbf{x}}_1^N \hat{\mathbf{n}}_1^N \mathbf{w}} \cdot \rho_{\hat{\mathbf{x}}_1^N|\hat{\mathbf{n}}_1^N \mathbf{w}} \cdot \rho_{\hat{\mathbf{n}}_1^N|\mathbf{w}}. \tag{2.35}$$

If the signal and noise were estimated subject to $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$, then the first term above will be singular at $\mathbf{y}_1^N = \hat{\mathbf{x}}_1^N + \hat{\mathbf{n}}_1^N$. The remaining two terms are evaluated using the model:

$$\hat{x}_k = f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \mathbf{w}\big) + \tilde{x}_k + \tilde{f}_k + v_k \tag{2.36}$$

$$\hat{n}_k = \sum_{i=1}^{M_n} \Big(w_n^{(i)} \cdot \hat{n}_{k-i}\Big) - \tilde{n}_k + \tilde{f}_{n,k} + v_{n,k} \tag{2.37}$$

$$y_k = \hat{x}_k + \hat{n}_k, \qquad \forall k \in \{1 \ldots N\} \tag{2.38}$$

$$\text{where} \quad \tilde{f}_k \triangleq f\big(x_{k-1}, \ldots x_{k-M}, \mathbf{w}\big) - f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \mathbf{w}\big), \tag{2.39}$$

$$\tilde{n}_k \triangleq n_k - \hat{n}_k, \quad \text{and} \quad \tilde{f}_{n,k} \triangleq \sum_{i=1}^{M_n} \Big(w_n^{(i)} \cdot \tilde{n}_{k-i}\Big). \tag{2.40}$$

Taking the negative log of $\rho_{\hat{\mathbf{x}}_1^N|\hat{\mathbf{n}}_1^N\mathbf{w}} \cdot \rho_{\hat{\mathbf{n}}_1^N|\mathbf{w}}$ produces the cost function:

$$J_c^{ec}(\mathbf{w}) = \sum_{k=1}^{N} \left(\log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} + \log(2\pi g_{n,k}) + \frac{(\hat{n}_k - \hat{n}_k^-)^2}{g_{n,k}} \right), \tag{2.41}$$

where $\hat{x}_k^- \triangleq f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \mathbf{w}\big)$, $\hat{n}_k^- \triangleq \sum_{i=1}^{M_n} \Big(w_n^{(i)} \cdot \hat{n}_{k-i}\Big)$, and g_k and $g_{n,k}$ are the variances of the errors $\tilde{\hat{x}}_k = (\hat{x}_k - \hat{x}_k^-)$ and $\tilde{\hat{n}}_k = (\hat{n}_k - \hat{n}_k^-)$, respectively. Notice the similarity between Equations 2.41 and 2.19 on page 27.

As before, if $\{\hat{x}_k\}_1^N$ is considered to be functionally independent of $\mathbf{w}$, then the cost function can be simplified. Here, the last two terms are dropped, yielding:

$$J_{ci}^{ec}(\mathbf{w}) = \sum_{k=1}^{N} \left(\log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} \right), \tag{2.42}$$

where only $\hat{x}_k^- = f(\hat{x}_{k-1}, \ldots, \hat{x}_{k-M}, \mathbf{w})$ and g_k are functions of the weights. As noted previously, this cost has the potential drawback of relying on the data $\{y_k\}_1^N$ only *indirectly*, through the estimates $\{\hat{x}_k\}_1^N$.

Variance Estimation – Colored Noise

Similarly, during estimation of the process noise variances σ_v^2 and $\sigma_{v_n}^2$, the errors in the estimates of all three quantities $\hat{\mathbf{x}}_1^N$, $\hat{\mathbf{n}}_1^N$, $\hat{\mathbf{w}}$ can be taken into account by maximizing the conditional density $\rho_{\hat{\mathbf{x}}_1^N \hat{\mathbf{n}}_1^N \hat{\mathbf{w}}|\mathbf{y}_1^N}$. The autoregression used for evaluating this density is written as in Equation 2.36–2.40, except that the dynamics error is defined as:

$$\tilde{f}_k \triangleq f\big(x_{k-1}, \ldots x_{k-M}, \mathbf{w}\big) - f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \hat{\mathbf{w}}\big). \tag{2.43}$$

The expansion

$$\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N \hat{\mathbf{n}}_1^N|\mathbf{w}} = \rho_{\mathbf{y}_1^N|\hat{\mathbf{x}}_1^N \hat{\mathbf{n}}_1^N \mathbf{w}} \cdot \rho_{\hat{\mathbf{x}}_1^N|\hat{\mathbf{n}}_1^N \mathbf{w}} \cdot \rho_{\hat{\mathbf{n}}_1^N|\mathbf{w}}, \tag{2.44}$$

produces the cost function:

$$J_c^{ec}(\sigma^2) = \sum_{k=1}^{N}\left(\log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} + \log(2\pi g_{n,k}) + \frac{(\hat{n}_k - \hat{n}_k^-)^2}{g_{n,k}}\right), \tag{2.45}$$

minimized with respect to σ_v^2 and $\sigma_{v_n}^2$, subject to $y_k = \hat{x}_k + \hat{n}_k$.

Algorithms for minimizing the error-coupled cost functions just described belong to the broad class of *joint estimation* approaches, and are explored along with the other dual EKF methods in Chapter 3. The next section presents cost functions for a second class of methods, referred to in this thesis as *marginal estimation* approaches.

2.4 Marginal Estimation

As described in the previous section, joint estimation methods are concerned with maximizing $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$ directly. A reasonable alternative to this approach can be found by separating the joint density function into two terms[2] as follows:

$$\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N} = \rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N} \cdot \rho_{\mathbf{w}|\mathbf{y}_1^N}, \tag{2.46}$$

or, in the case of colored measurement noise:

$$\rho_{\mathbf{x}_1^N \mathbf{n}_1^N \mathbf{w}|\mathbf{y}_1^N} = \rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{w}\mathbf{y}_1^N} \cdot \rho_{\mathbf{w}|\mathbf{y}_1^N}. \tag{2.47}$$

Often, $\hat{\mathbf{x}}_1^N$ is found by maximizing the first term on the right, and $\hat{\mathbf{w}}$ is found by maximizing the second term, $\rho_{\mathbf{w}|\mathbf{y}_1^N}$. This approach is referred to in this thesis as *marginal estimation.*

The second term, $\rho_{\mathbf{w}|\mathbf{y}_1^N}$, is independent of $\mathbf{x}_1^N$; only the first term ($\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}$ in Equation 2.46) is a function of the signal. Hence, maximizing the first term on the right with respect to $\mathbf{x}_1^N$ is the same as maximizing the joint density, $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$, on the left. However, both terms are functionally dependent on the weights, so the same is *not* true of maximizing the second term, $\rho_{\mathbf{w}|\mathbf{y}_1^N}$, with respect to $\mathbf{w}$. That is, because the first term depends on $\mathbf{w}$, maximizing $\rho_{\mathbf{w}|\mathbf{y}_1^N}$ 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, estimates $\hat{\mathbf{w}}$ found by maximizing the marginal density function $\rho_{\mathbf{w}|\mathbf{y}_1^N}$ are consistent and unbiased, if conditions of sufficient excitation are met [54]. The marginal estimation approach is exemplified by the maximum-likelihood approaches [26, 44] and EM approaches [58, 76]

[2] A second expansion $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N} = \rho_{\mathbf{w}|\mathbf{x}_1^N \mathbf{y}_1^N} \cdot \rho_{\mathbf{x}_1^N|\mathbf{y}_1^N}$ is also possible, but does not yield practical algorithms. This is discussed in more detail in Appendix B.

mentioned in the introduction. Prediction error algorithms (*e.g.*, RPE [47]) represent an approximation to the marginal estimation approach.

While the present claim is that the relationship of these algorithms to the joint estimation methods can be understood in terms of Equation 2.46 or 2.47, it is important to note that this equation does not represent their primary motivation. Rather, the motivation for marginal estimation methods comes from considering 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. Furthermore, these signal estimates are invariably produced by maximizing the first term $\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}$ (or $\rho_{\mathbf{x}_1^N\mathbf{n}_1^N|\mathbf{w}\mathbf{y}_1^N}$) of Equation 2.46 (2.47) *in some way*. This last fact justifies the use of Equation 2.46 (and 2.47) for understanding the relationship between marginal estimation and joint estimation approaches.

The pertinent cost functions for marginal estimation are laid out in the remainder of this section, while the algorithms are described in Chapter 3.

2.4.1 Marginal Weight Estimation

As just mentioned, marginal estimation methods find the weight estimates $\hat{\mathbf{w}}$ by maximizing the second term, $\rho_{\mathbf{w}|\mathbf{y}_1^N}$, in Equation 2.46. Applying Bayes' rule here produces:

$$\rho_{\mathbf{w}|\mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{w}}}{\rho_{\mathbf{y}_1^N}}. \tag{2.48}$$

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 as (see Appendix A):

$$\rho_{\mathbf{y}_1^N|\mathbf{w}} = \prod_{k=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{\varepsilon_k}^2}} \exp\left(-\frac{(y_k - \overline{y_{k|k-1}})^2}{2\sigma_{\varepsilon_k}^2}\right), \tag{2.49}$$

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

is the conditional mean (and optimal prediction), $\sigma_{\varepsilon_k}^2$ is the prediction error variance.Note that the assumption that $\rho_{\mathbf{y}_1^N|\mathbf{w}}$ is Gaussian is only true if the model $f(\cdot)$ is linear. For nonlinear models, the above form is an approximation made *in addition* to the Gaussian assumption on the noise terms, n_k and v_k.

Taking the log of this likelihood function gives:

$$\log \rho_{\mathbf{y}_1^N|\mathbf{w}} = -\frac{1}{2}\sum_{k=1}^{N}\left(\log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \overline{y_{k|k-1}})^2}{\sigma_{\varepsilon_k}^2}\right). \tag{2.50}$$

When the signal model is linear, $\overline{y_{k|k-1}}$ can be computed using an ordinary Kalman filter. For nonlinear models, however, the expectation can only be approximated by an extended Kalman filter (see Chapter 3).

Note that the log-likelihood function takes the same form whether the measurement noise is colored or white. The following paragraphs describe how the log-likelihood is the foundation for a few different marginal estimation methods.

Prediction Error Cost

Often the variance $\sigma^2_{\varepsilon_k}$ 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:

$$J^{pe}(\mathbf{w}) = \sum_{k=1}^{N} (y_k - \overline{y_{k|k-1}})^2. \tag{2.51}$$

Recursive prediction error algorithms [47, 87] minimize this simplified cost function with respect to the weights $\mathbf{w}$. 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.

Maximum-Likelihood Cost

When the dependence of the prediction error variance on the weights and time index is taken into account, the form of the cost function is:

$$J^{ml}(\mathbf{w}) = \sum_{k=1}^{N} \left(\log(2\pi\sigma^2_{\varepsilon_k}) + \frac{(y_k - \overline{y_{k|k-1}})^2}{\sigma^2_{\varepsilon_k}} \right). \tag{2.52}$$

Note, $J^{ml}(\mathbf{w})$ is the maximum likelihood cost, while the prediction error cost $J^{pe}(\mathbf{w})$ represents an approximation.

EM Algorithm

Another approach to maximizing $\rho_{\mathbf{w}|\mathbf{y}_1^N}$ is offered by the Expectation-Maximization (EM) algorithm [16, 76, 71]. The EM algorithm can be derived by first expanding the log likelihood as:

$$\log \rho_{\mathbf{y}_1^N|\mathbf{w}} = \log \rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}} - \log \rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}, \tag{2.53}$$

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

$$\log \rho_{\mathbf{y}_1^N|\mathbf{w}} = E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}}|\mathbf{y}_1^N, \hat{\mathbf{w}}] - E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}|\mathbf{y}_1^N, \hat{\mathbf{w}}], \tag{2.54}$$

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 [11], so it will decrease for *any* solution $\mathbf{w}$ moving away from the current estimate $\hat{\mathbf{w}}$ (and the negative will increase). Therefore, choosing $\mathbf{w}$ to maximize the first term on the right alone will always increase the log likelihood on the left hand side. In other words, in order to maximize $\rho_{\mathbf{y}_1^N|\mathbf{w}}$, the EM algorithm repeatedly maximizes $E_{\mathbf{X}|\mathbf{YW}}[\log \rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}}|\mathbf{y}_1^N, \hat{\mathbf{w}}]$ with respect to $\mathbf{w}$, each time setting $\hat{\mathbf{w}}$ to the new maximizing value. For the white noise case, then, the EM cost function is:

$$J^{em} = E_{\mathbf{X}|\mathbf{YW}}\left[\sum_{k=1}^{N}\left(\log(2\pi\sigma_n^2) + \frac{(y_k - x_k)^2}{\sigma_n^2} + \log(2\pi\sigma_v^2) + \frac{(x_k - x_k^-)^2}{\sigma_v^2}\right) \middle| \mathbf{y}_1^N, \hat{\mathbf{w}}\right], \tag{2.55}$$

where $x_k^- \triangleq f(x_{k-1}, \ldots, x_{k-M}, \mathbf{w})$, as before. The evaluation of the expectation in J^{em} is discussed in Appendix F.

Colored Noise EM

When the measurement noise is colored, $\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}}$ is no longer easy to evaluate. Instead, the following expansion is used:

$$\log \rho_{\mathbf{y}_1^N|\mathbf{w}} = \log \rho_{\mathbf{y}_1^N\mathbf{x}_1^N\mathbf{n}_1^N|\mathbf{w}} - \log \rho_{\mathbf{x}_1^N\mathbf{n}_1^N|\mathbf{w}\mathbf{y}_1^N}, \tag{2.56}$$

so that $E_{\mathbf{XN}|\mathbf{YW}}[\log \rho_{\mathbf{y}_1^N\mathbf{x}_1^N\mathbf{n}_1^N|\mathbf{w}}|\mathbf{y}_1^N, \hat{\mathbf{w}}]$ is the term to be maximized. Recall, however, that in:

$$\log \rho_{\mathbf{y}_1^N\mathbf{x}_1^N\mathbf{n}_1^N|\mathbf{w}} = \log \rho_{\mathbf{y}_1^N|\mathbf{x}_1^N\mathbf{n}_1^N\mathbf{w}} + \log \rho_{\mathbf{x}_1^N\mathbf{n}_1^N|\mathbf{w}}, \tag{2.57}$$

the first term is singular at $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$. Hence, one should instead maximize the expectation of the second term subject to $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$. This gives the EM cost for colored noise as:

$$J_c^{em} = E_{\mathbf{XN}|\mathbf{YW}}\left[\sum_{k=1}^{N}\left(\log(2\pi\sigma_v^2) + \frac{(x_k - x_k^-)^2}{\sigma_v^2} + \log(2\pi\sigma_{v_n}^2) + \frac{(n_k - n_k^-)^2}{\sigma_n^2}\right) \middle| \mathbf{y}_1^N, \hat{\mathbf{w}}\right]. \tag{2.58}$$

Details of the EM algorithms are provided in the next chapter.

2.4.2 Marginal Variance Estimation

If the noise variances unknown, they can be estimated along with the weights by including them in the log likelihood function. The resultant prediction error, maximum-likelihood, or EM cost

function can be minimized.

Prediction Error Variance Estimation

The prediction error cost is the same as for weight estimation:

$$J^{pe}(\sigma^2) = \sum_{k=1}^{N} (y_k - \overline{y_{k|k-1}})^2, \tag{2.59}$$

except now the prediction $\overline{y_{k|k-1}}$ is viewed as a recursive function of the unknown variance, σ^2. The form of this function is explored in Chapter 3.

Maximum-Likelihood Variance Estimation

The maximum-likelihood cost function for variance estimation is:

$$J^{ml}(\sigma^2) = \sum_{k=1}^{N} \left(\log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \overline{y_{k|k-1}})^2}{\sigma_{\varepsilon_k}^2} \right). \tag{2.60}$$

This is identical to the maximum-likelihood weight cost function, except that the argument has been changed to emphasize the estimation of the unknown variances. The specific ways in which $\overline{y_{k|k-1}}$ and $\sigma_{\varepsilon_k}^2$ depend on the variance terms are shown in Chapter 3.

EM Variance Estimation

Alternatively, the variances can be estimated within the EM framework. If the variances are unknown, then the expectation is conditioned on their estimated values during the E-step. For white noise, this means the expectation $E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}} | \mathbf{y}_1^N, \hat{\mathbf{w}}, \hat{\sigma_v^2}, \hat{\sigma_n^2}]$ is maximized with respect to $\mathbf{w}$, σ_v^2, and σ_n^2 during the M-step. For colored noise, $E_{\mathbf{X}\mathbf{N}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{x}_1^N \mathbf{n}_1^N} | \mathbf{y}_1^N, \hat{\mathbf{w}}, \hat{\sigma_v^2}, \hat{\sigma_{v_n}^2}]$ is maximized with respect to $\mathbf{w}$, σ_v^2, and $\sigma_{v_n}^2$ during the M-step. The forms of the cost functions are developed fully in Chapter 3, and Appendix F.

2.4.3 Marginal Signal Estimation

As noted at the beginning of this section, marginal estimation methods are motivated by the maximization of the marginal density $\rho_{\mathbf{y}_1^N|\mathbf{w}}$ alone. However, as shown above, in the maximum-likelihood cost the term $\overline{y_{k|k-1}}$ must be computed, and $E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}}]$ (or $E_{\mathbf{X}\mathbf{N}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{x}_1^N \mathbf{n}_1^N}]$) is required for the EM algorithm.

As shown in the next chapter, computing either of these quantities requires the computation of some form of signal estimate. In the white noise case, these estimates are invariably generated

by maximizing $\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}$, which is the first term in Equation 2.46. Because $\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}$ is being maximized, the interpretation of the marginal estimation methods given in Equation 2.46 is justified. The term can be written as:

$$\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}}}{\rho_{\mathbf{y}_1^N|\mathbf{w}}} \tag{2.61}$$

and the signal can be estimated by maximizing the numerator $\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}}$ with respect to $\{x_k\}_1^N$. This is equivalent to minimizing the joint cost $J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$ defined in Section 2.3 by Equation 2.12[3].

Similarly, for the colored noise case, both signal and noise estimates are required, and are generated by maximizing $\rho_{\mathbf{x}_1^N\mathbf{n}_1^N|\mathbf{w}\mathbf{y}_1^N}$, which is the first term in Equation 2.47. This can be shown to be equivalent to minimizing the joint cost $J_c^j(\mathbf{x}_1^N, \mathbf{n}_1^N, \hat{\mathbf{w}})$ defined by Equation 2.30 on page 31.

2.5 Discussion

The dual estimation problem is to find signal and weight estimates which are in some sense optimal. A sensible measure of optimality is given by the joint conditional density $\rho_{\mathbf{x}_1^N\mathbf{w}|\mathbf{y}_1^N}$, and the corresponding cost function $J^j(\mathbf{x}_1^N, \mathbf{w})$. The joint cost $J^j(\mathbf{x}_1^N, \mathbf{w})$ is essentially a two-argument function, with a fairly high degree of coupling between the arguments. Although $J^j(\mathbf{x}_1^N, \mathbf{w})$ can be minimized with respect to both the signal and weights simultaneously (*e.g.*, by the joint EKF, as shown in Chapter 3), another approach is to minimize the function by alternately minimizing with respect to one argument and then the other. This can be done either by substituting current estimates for one of the arguments in the cost function, or by deriving new cost functions that incorporate the statistics of these estimates. These alternative cost functions account for the errors in the estimates of each argument while the other is being estimated.

Still other costs can be found by expanding the joint density, and minimizing the terms separately. While these *marginal estimation* approaches fail to maximize the joint density, unbiased estimates of the parameters are produced. These methods, exemplified by maximum-likelihood, prediction error, and EM algorithms, have been shown to be quite useful in practice.

The various cost functions derived in this chapter are summarized in Table 2.1. For brevity, only the white noise forms of the costs and densities are shown; equation numbers for the colored noise case are shown in parentheses. Furthermore, no explicit signal estimation cost is given for the marginal estimation methods because signal estimation is only an *implicit* step of the marginal approach. Marginal signal estimation is performed using the joint cost $J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$.

[3]The error-coupled cost $J^{ec}(\mathbf{x}_1^N, \hat{\mathbf{w}})$ can also be used to generate the necessary signal estimates. However, this approach is not explored in this thesis.

Table 2.1: Summary of the cost functions derived in this Chapter. Some costs differ slightly for the colored noise case; their equation numbers are enclosed in parentheses.

	Symbol	Name of Cost	Density	Equation	Argument
Joint	$J^j(\mathbf{x}_1^N, \mathbf{w})$	joint	$\rho_{\mathbf{x}_1^N \mathbf{w}\|\mathbf{y}_1^N}$	2.11(2.29)	$\{x_k\}_1^N, \mathbf{w}$
	$J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$	joint signal	$\rho_{\mathbf{x}_1^N \mathbf{w}\|\mathbf{y}_1^N}$	2.12(2.30)	$\{x_k\}_1^N$
	$J^j(\hat{\mathbf{x}}_1^N, \mathbf{w})$	joint weight	$\rho_{\mathbf{x}_1^N \mathbf{w}\|\mathbf{y}_1^N}$	2.13(2.31)	$\mathbf{w}$
	$J_i^j(\hat{\mathbf{x}}_1^N, \mathbf{w})$	joint weight (indep.)	$\rho_{\mathbf{x}_1^N\|\mathbf{w}}$	2.14(2.31)	$\mathbf{w}$
	$J^j(\sigma^2)$	joint variance	$\rho_{\mathbf{x}_1^N \mathbf{w}\|\mathbf{y}_1^N}$	2.15(2.32)	$\sigma_v^2, \sigma_n^2(\sigma_{v_n}^2)$
	$J^{ec}(\mathbf{x}_1^N)$	error-coupled signal	$\rho_{\mathbf{x}_1^N \hat{\mathbf{w}}\|\mathbf{Y}}$	2.17(2.34)	$\{x_k\}_1^N$
	$J^{ec}(\mathbf{w})$	error-coupled weight	$\rho_{\hat{\mathbf{x}}_1^N \mathbf{w}\|\mathbf{y}}$	2.19(2.41)	$\mathbf{w}$
	$J_i^{ec}(\mathbf{w})$	e.-c. weight (indep.)	$\rho_{\hat{\mathbf{x}}_1^N\|\mathbf{w}}$	2.20(2.42)	$\mathbf{w}$
	$J^{ec}(\sigma^2)$	error-coupled variance	$\rho_{\hat{\mathbf{x}}_1^N \hat{\mathbf{w}}\|\mathbf{y}}$	2.22(2.45)	$\sigma_v^2, \sigma_n^2(\sigma_{v_n}^2)$
Marginal	$J^{pe}(\mathbf{w})$	prediction error	$\sim \rho_{\mathbf{w}\|\mathbf{y}_1^N}$	2.51	$\mathbf{w}$
	$J^{pe}(\sigma^2)$	prediction error	$\sim \rho_{\mathbf{w}\|\mathbf{y}_1^N}$	2.59	$\sigma_v^2, \sigma_n^2(\sigma_{v_n}^2)$
	$J^{ml}(\mathbf{w})$	max. likelihood	$\rho_{\mathbf{w}\|\mathbf{y}_1^N}$	2.52	$\mathbf{w}$
	$J^{ml}(\sigma^2)$	max. likelihood	$\rho_{\mathbf{w}\|\mathbf{y}_1^N}$	2.60	$\sigma_v^2, \sigma_n^2(\sigma_{v_n}^2)$
	$J^{em}(\mathbf{w})$	EM	$n.a.$	2.55(2.58)	$\mathbf{w}$
	$J^{em}(\sigma^2)$	EM	$n.a.$	2.55(2.58)	$\sigma_v^2, \sigma_n^2(\sigma_{v_n}^2)$

In other words, the signal estimation cost functions for joint and marginal estimation are the same. The two approaches primarily differ, then, in the form of the cost functions used for weight estimation. The following *qualitative* comments about the weight estimation part of the problem might therefore shed some light on the tradeoffs between the approaches:

- By maximizing $\rho_{\mathbf{y}_1^N|\mathbf{w}}$, marginal estimation methods ensure an unbiased estimate of the weights. However, the measurement noise in the data $\{y_k\}_1^N$ will affect this estimate by increasing its variance.

- For the joint estimation methods, lower variance weight estimates can be obtained by minimizing $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}}$ or $\rho_{\mathbf{y}_1^N \hat{\mathbf{x}}_1^N|\mathbf{w}}$, using signal estimates. However, the resulting weight estimates will only be unbiased if $\{\hat{x}_k\}_1^N$ has converged to $\{x_k\}_1^N$. In fact, the potential for lower variance, higher bias weight estimates is verified experimentally in Figure 4.25 and Figure 4.41 in Chapter 4.

- If the cost functions corresponding to $\rho_{\mathbf{x}_1^N|\mathbf{w}}$ and $\rho_{\hat{\mathbf{x}}_1^N|\mathbf{w}}$ are used, the variance of $\hat{\mathbf{w}}$ is further reduced, but at the expense of even greater bias if $\{\hat{x}_k\}_1^N$ has not converged to the true signal.

These comments are rather too vague to be of any immediate use, because the relative values

of "lower variance" and "unbiased" will ultimately depend on various properties of the actual data (such as its length and SNR), and the optimization procedure. Of much greater use would be a *quantitative* theoretical analysis of the bias/variance tradeoffs of the different cost functions. This, however, is a formidable task, and will not be attempted in this thesis.

Further complicating matters is the fact that different approximations are used to arrive at the various joint and marginal cost functions. The relative severity of these approximations is not inherently obvious, and will largely depend on the particular noisy time series at hand. The purpose of this chapter, rather, is to show what the approximations and assumptions are, and how they lead to different dual estimations methods.

However, experimental evaluations and comparisons of the variety of cost functions are provided in Chapter 4. In this context, the above comments will engender useful hypotheses for interpreting the results. For example, the joint cost tends to show its best performance on signals with lower effective noise levels, where the errors in $\hat{x}_k$ will tend to be less severe. These signal estimates are less likely to produce bias in the weights. For linear models – for which the weight errors can be computed – the bias-variance tradeoff can be observed even more directly, as mentioned above. Another outcome of the experiments in Chapter 4 is the similarity of performance of the prediction error and maximum-likelihood costs in various settings; this fact can be explained by their common theoretical underpinnings. In general, understanding the relationship between the various cost functions, and between joint and marginal estimation methods, provides a guiding principle for selecting an algorithm given a particular application.

Chapter 3 describes a family of algorithms, called the ***dual extended Kalman filters***, for minimizing the variety of cost functions just derived. Algorithmic issues, such as variance initialization and computational expense, are also discussed.

Chapter 3

Algorithms

3.1 Overview

In the preceding chapter, the dual estimation problem was considered from a probabilistic perspective in order to demonstrate the relationship between several different cost functions. These cost functions approach the *off-line* problem of estimating the parameters $\mathbf{w}$ and time-series $\{x_k\}_1^N$ from an entire sequence of noisy data $\{y_k\}_1^N$, which is available all at once. This procedure is necessarily noncausal because estimates of the signal at times $k = 1, 2, \ldots, N$ all depend on the measurement, y_N, at final time $k = N$. Off-line processing typically entails an *iterative* procedure of repeatedly minimizing the cost with respect to first the signal, and then the weights (see Figure 1.5(a) on page 10).

Although iterative algorithms are mentioned in the following sections whenever they are relevant, the focus of the current chapter is on the *sequential* estimation of the signal and parameters as the noisy measurements y_k become available, as shown in Figure 1.5(b). Iterative algorithms are most useful when the length N of the noisy time-series is *fixed.* Sequential algorithms are more appropriate for on-line applications, wherein new data arrive during processing; the length of the time-series continually increases with the time index, k.

While it is certainly possible to apply batch-style estimation to the on-line problem by using all the available data $\{y_t\}_1^k$ at every time step, this approach requires recomputing estimates of the entire trajectory $\{x_t\}_1^k$ and $\mathbf{w}$ from $\{y_t\}_1^k$ each time a new measurement y_k arrives. The expense of such an approach becomes prohibitive as $k \to \infty$.

Instead, *sequential* approaches have the property that their computational and memory requirements are constant in time. Such algorithms are typically recursive in nature, so that the new information in measurement y_k is combined with the existing estimates of the signal and weights. For weight estimation, the previous estimate of $\mathbf{w}$ (based on data $\{y_t\}_1^{k-1}$) is updated upon arrival of y_k. During signal estimation, rather than update the entire trajectory of estimates $\{\hat{x}_t\}_1^k$, only

an M-vector of lagged values $\mathbf{x}_k \triangleq [x_k, \ldots, x_{k-M+1}]^T$ is estimated using the new measurement. As mentioned in Chapter 1, sequential methods can easily be applied to off-line estimation; the algorithm is repeatedly passed over the same set of data, with the weights from one iteration used to initialize the next.

Even though they do not take advantage of data in the future, sequential algorithms should still have the property that the *current* estimates are optimal with respect to the corresponding batch cost function on the same data. For example, at time $k = N$, recursive least squares (RLS) produces an estimate $\hat{\mathbf{w}}_N$ that is equivalent to the batch least squares solution using $\{y_k\}_1^N$; similarly, the Kalman filter produces an estimate $\hat{\mathbf{x}}_N$ equal to that produced by the Kalman smoother. However, the sequential estimates of the entire series will not generally be the same as the non-causal estimates. The dual Kalman filter described later in this chapter is therefore not equivalent to a batch-style algorithm.

Before introducing the dual Kalman filter, the next two sections review some standard sequential estimation algorithms. First, Section 3.2 develops the sequential signal estimation problem assuming a known model, and provides a theoretical review of the Kalman filter (KF) and extended Kalman filter (EKF). Second, the application of Kalman filtering to weight estimation using a clean signal is shown in Section 3.3; an alternate form of the weight filter – which is useful for minimizing other cost functions – is also introduced in this section.

The remainder of the chapter considers the dual estimation problem. The joint EKF algorithm, in which the signal and weights are estimated in a combined state vector, is described in Section 3.4. Separate state-space representations are used in Section 3.5 to develop the family of algorithms called dual Kalman filters. Finally, additional issues relating to the practical implementation of the dual EKF are addressed in Section 3.6.

3.2 Signal Estimation

This section develops the use of Kalman filtering for MAP signal estimation when the model and noise statistics are *known*. The discussion shows the need for a state-space representation of the time-series. The unknown model problem is treated in Sections 3.4 and 3.5.

3.2.1 Batch Estimation

As just stated, a batch algorithm uses all of the available data to estimate the entire signal $\{x_t\}_1^k$. In the MAP context, this is stated formally as:

$$\mathbf{x}_1^k = \arg\max_{\mathbf{x}_1^k} \rho_{\mathbf{x}_1^k|\mathbf{y}_1^k\mathbf{w}}, \tag{3.1}$$

which gives the most probable estimate of the signal, given the noisy data up to the present time k. Bayes rule can be used to rewrite this density as:

$$\rho_{\mathbf{x}_1^k|\mathbf{y}_1^k\mathbf{w}} = \frac{\rho_{\mathbf{y}_1^k|\mathbf{x}_1^k\mathbf{w}} \cdot \rho_{\mathbf{x}_1^k|\mathbf{w}}}{\rho_{\mathbf{y}_1^k|\mathbf{w}}}. \tag{3.2}$$

Because the denominator does not depend on the signal, $\mathbf{x}_1^k$ can be estimated by minimizing the negative log of the numerator. As shown on page 23, when the measurement noise is a white Gaussian process, this gives the cost function:

$$J(\mathbf{x}_1^k) = \sum_{t=1}^{k} \left(\frac{(y_t - x_t)^2}{\sigma_n^2} + \frac{(x_t - x_t^-)^2}{\sigma_v^2} \right). \tag{3.3}$$

The optimal estimate $\hat{\mathbf{x}}_1^k$ can be found either by batch least-squares (as describe in the errors-in-variables context in Appendix G), or recursively, using a Kalman smoothing algorithm [68]. Both of these algorithms are necessarily *off-line*, and produce $\{\hat{x}_t\}_1^k$ as defined in Equation 3.1.

3.2.2 Kalman Filtering – White Noise Case

The preceding development made the assumption that the additive measurement noise is white. This assumption is also made in the subsequent paragraphs; the colored noise case is considered in Section 3.2.3 on page 51.

A *sequential* algorithm can be derived if only the MAP estimate of the *current* signal value x_k is desired, rather than an estimate of the entire signal $\{x_t\}_1^k$. As shown in the following development, this will require the introduction of a state-space representation of the system. Sequential MAP estimation seeks the current estimate $\hat{x}_k$ that is most probable given the model and all the measurements $\{y_t\}_1^k$ up to and including the present time. This goal is formally expressed as:

$$\hat{x}_k = \arg\max_{x_k} \rho_{x_k|\mathbf{y}_1^k\mathbf{w}}. \tag{3.4}$$

Note that the value $\hat{x}_k$ satisfying this equation is the *same* as the estimate of x_k found by minimizing $J(\mathbf{x}_1^k)$ in Equation 3.3. In the sequential framework, however, estimates of *all* the past values of the signal are not desired; only a limited number of values are needed, as shown in the next few pages.

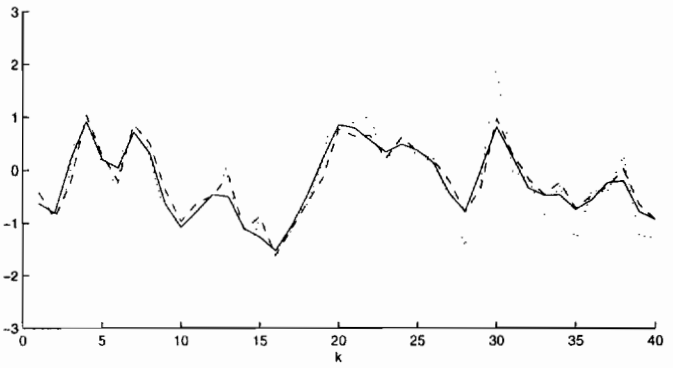


Figure 3.1: The Kalman filter (dashed line) and Kalman smoother (solid line) estimates are equivalent at the final time $k = N = 40$. The noisy data (dotted line) were generated by adding white noise to a linear AR signal.

The joint density to be maximized can be rewritten as:

$$\rho_{x_k|\mathbf{y}_1^k\mathbf{w}} = \frac{\rho_{x_k\mathbf{y}_1^k|\mathbf{w}}}{\rho_{\mathbf{y}_1^k|\mathbf{w}}} = \frac{\rho_{x_k y_k|\mathbf{y}_1^{k-1}\mathbf{w}} \cdot \rho_{\mathbf{y}_1^{k-1}|\mathbf{w}}}{\rho_{\mathbf{y}_1^k|\mathbf{w}}}. \tag{3.5}$$

Because $\rho_{\mathbf{y}_1^{k-1}|\mathbf{w}}$ and $\rho_{\mathbf{y}_1^k|\mathbf{w}}$ are functionally independent of x_k, the MAP estimate can be obtained by maximizing $\rho_{x_k y_k|\mathbf{y}_1^{k-1}\mathbf{w}}$ alone. This is expanded as:

$$\rho_{x_k y_k|\mathbf{y}_1^{k-1}\mathbf{w}} = \rho_{y_k|\mathbf{y}_1^{k-1}x_k\mathbf{w}} \cdot \rho_{x_k|\mathbf{y}_1^{k-1}\mathbf{w}} \tag{3.6}$$

Note that $\rho_{y_k|\mathbf{y}_1^{k-1}x_k\mathbf{w}} = \rho_{y_k|x_k}$. If the process noise v_k and measurement noise n_k are both zero-mean white Gaussian processes, then Equation 3.6 evaluates as:

$$\rho_{x_k y_k|\mathbf{y}_1^{k-1}\mathbf{w}} = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(y_k - x_k)^2}{2\sigma_n^2}\right) \cdot \frac{1}{\sqrt{2\pi p_k^-}} \exp\left(-\frac{(x_k - \hat{x}_k^-)^2}{2p_k^-}\right),$$

where $\hat{x}_k^- = E[x_k|\{y_t\}_1^{k-1}, \mathbf{w}]$ and $p_k^- = E[(x_k - \hat{x}_k^-)^2|\{y_t\}_1^{k-1}, \mathbf{w}]$ are the prior mean and variance of x_k given the data $\{y_t\}_1^{k-1}$, but *before* the measurement y_k has arrived. Taking the negative log gives the cost function:

$$J(x_k) = \frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - \hat{x}_k^-)^2}{p_k^-}, \tag{3.7}$$

which can be minimized with respect to x_k to produce the desired sequential MAP estimate of the signal. Note that both σ_n^2 and p_k^- are functionally independent of x_k.

Minimizing $J(x_k)$ with respect to x_k is equivalent to minimizing the batch cost $J(\mathbf{x}_1^k)$ of Equation 3.3 with respect to x_k, as illustrated in Figure 3.1. In this sense, the sequential estimates are optimal with respect to the batch cost function. However, minimizing the sequential cost $J(x_k)$ first requires determining the value of the prior mean (or prediction) $\hat{x}_k^-$ and its variance p_k^-. As

is shown in the following development, these prior statistics are a function of the statistics at the previous time step; calculating them requires a recursive estimation procedure derived within a state-space framework.

Linear Model

For the sake of simplicity, the calculation of priors is presented first for the linear-model case; the nonlinear AR model of Equation 1.1 is replaced by the following linear one:

$$\begin{aligned} x_k &= \sum_{i=1}^{M} w_i x_{k-i} + v_k \\ y_k &= x_k + n_k. \end{aligned} \tag{3.8}$$

By defining $\mathbf{x}_{k-1} = [x_{k-1}, \ldots, x_{k-M}]^T$, the first equation can be rewritten as $x_k = \mathbf{w}^T \mathbf{x}_{k-1} + v_k$.

Substituting this expression for x_k in $\hat{x}_k^- = E[x_k | \{y_t\}_1^{k-1}, \mathbf{w}]$ gives:

$$\hat{x}_k^- = E[\mathbf{w}^T \mathbf{x}_{k-1} + v_k | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{3.9a}$$

$$= \mathbf{w}^T E[\mathbf{x}_{k-1} | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{3.9b}$$

$$= \mathbf{w}^T \hat{\mathbf{x}}_{k-1}, \tag{3.9}$$

where $\hat{\mathbf{x}}_{k-1} \triangleq E[\mathbf{x}_{k-1} | \{y_t\}_1^{k-1}, \mathbf{w}]$ is the conditional (or *posterior*[1]) mean of $\mathbf{x}_{k-1}$ given the data $\{y_t\}_1^{k-1}$. Similarly, $p_k^- = E[(x_k - \hat{x}_k^-)^2 | \{y_t\}_1^{k-1}, \mathbf{w}]$ can be rewritten as:

$$p_k^- = E[(\mathbf{w}^T \mathbf{x}_{k-1} + v_k - \mathbf{w}^T \hat{\mathbf{x}}_{k-1})^2 | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{3.10a}$$

$$= \mathbf{w}^T \cdot E[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^2 | \{y_t\}_1^{k-1}, \mathbf{w}] \cdot \mathbf{w} + \sigma_v^2 \tag{3.10b}$$

$$= \mathbf{w}^T \mathbf{P}_{k-1} \mathbf{w} + \sigma_v^2, \tag{3.10}$$

where $\mathbf{P}_{k-1} \triangleq E[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^T | \{y_t\}_1^{k-1}, \mathbf{w}]$ is defined as the conditional (or *posterior*) covariance of $\mathbf{x}_{k-1}$ given the data $\{y_t\}_1^{k-1}$.

In summary, generating *posterior* estimates of the signal $\hat{x}_k$ requires computing the ***prior*** mean $\hat{x}_k^-$ and variance p_k^-. However, computing the *prior* mean and variance requires computing the *posterior* mean $\hat{\mathbf{x}}_{k-1}$ and covariance $\mathbf{P}_{k-1}$ at the previous time step. The situation is depicted in Figure 3.2. At the next time step, to compute $\hat{x}_{k+1}$ when y_{k+1} arrives will ultimately require the vector estimate $\hat{\mathbf{x}}_k$ (not just $\hat{x}_k$), as well as the error covariance $\mathbf{P}_k$. Therefore, sequential estimation of the signal $\{x_k\}_1^N$ requires computing $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ *recursively* for all $k \in [1, \ldots, N]$.

[1]The use of the term *posterior* herein applies to the statistic of $\mathbf{x}_t$ given data up to time t, whereas the term *prior* applies to the statistic of $\mathbf{x}_t$ given data up to time $(t-1)$.

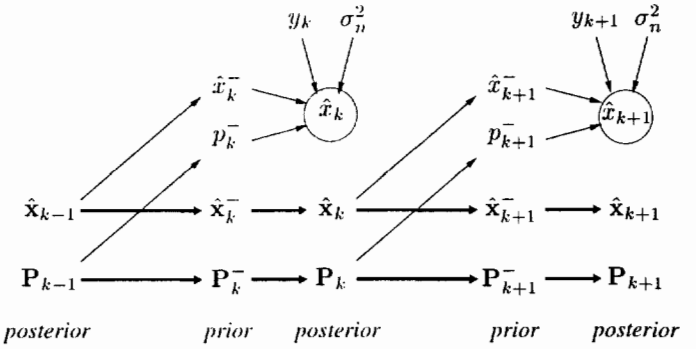


Figure 3.2: The dependence of signal estimate $\hat{x}_k$ on state estimate $\hat{\mathbf{x}}_{k-1}$ and covariance $\mathbf{P}_{k-1}$.

Note also that when the conditional density $\rho_{x_k|\mathbf{y}_1^k\mathbf{w}}$ is Gaussian, then the MAP estimate $\hat{x}_k$ (which maximizes this density) is the same as the conditional mean $E[x_k|\{y_k\}_1^N, \mathbf{w}]$. Hence, $\hat{x}_k$ can be taken directly from the first element of the conditional mean $\hat{\mathbf{x}}_k = E[\mathbf{x}_k|\{y_k\}_1^N, \mathbf{w}]$.

State-Space Representation

The vector $\mathbf{x}_k$ which must be estimated is usually referred to as a *state vector*. The current state of the system is defined as the minimal amount of information such that all future behavior of the system can be determined from the future inputs to the system and the current system state. For a more formal and complete discussion of *state* and state-space representations of systems, see [6, 8].

The linear AR process of Equation 3.8 can be equivalently described by the following state-space equations:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A} \cdot \mathbf{x}_{k-1} + \mathbf{B} \cdot v_k \\ \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M+1} \end{bmatrix} &= \begin{bmatrix} w_1 & w_2 & \cdots & w_M \\ 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ x_{k-2} \\ \vdots \\ x_{k-M} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot v_k, \end{aligned} \tag{3.11}$$

$$\begin{aligned} y_k &= \mathbf{C} \cdot \mathbf{x}_k + n_k \\ &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \cdot \mathbf{x}_k + n_k. \end{aligned} \tag{3.12}$$

The equivalence with Equation 3.8 is seen by looking at the top row of the matrix equation 3.11. An infinite variety of state-space representations can be found for a linear AR model by projecting $\mathbf{x}_k$ onto an alternate basis (*via* a linear transformation). This transformation will of course change

the form of the system matrices **A**, **B**, and **C**. The particular form shown here is called the *control canonical representation* [19], as determined by the special structure of the **A** and **B** matrices.

Linear Kalman Filter

The desired conditional mean $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ in Equations 3.9 and 3.10 are calculated by the Kalman filter algorithm when the known model is linear with Gaussian statistics. Because the mean and covariance completely specify a Gaussian density function, the Kalman filter effectively estimates the entire conditional density $\rho_{\mathbf{x}_k|\mathbf{y}_1^k\mathbf{w}}$ at each time step.

The mean and mode of a Gaussian probability density function are identical, so calculating the conditional mean $\hat{\mathbf{x}}_k$ is equivalent to calculating the MAP estimate; *i.e.*:

$$\hat{\mathbf{x}}_k = E[\mathbf{x}_k|\{y_t\}_1^k, \mathbf{w}] = \arg\max_{\mathbf{x}_k} \rho_{\mathbf{x}_k|\mathbf{y}_1^k,\mathbf{w}}. \tag{3.13}$$

The first element of this state estimate satisfies Equation 3.4. Also, note that the covariance $\mathbf{P}_k$ of the state can equivalently be interpreted as the *error covariance* of the MAP estimate:

$$\mathbf{P}_k = Cov[\mathbf{x}_k|\{y_t\}_1^k, \mathbf{w}] = E[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T|\{y_t\}_1^k, \mathbf{w}]. \tag{3.14}$$

Hence, the Kalman filter equations can be derived from either a minimum mean squared error (MMSE) approach (yielding the conditional mean) or from a MAP perspective. There are numerous textbooks [43, 79] on the subject of Kalman filtering; most of these explain the topic from the MMSE perspective. A derivation of the Kalman filter from MAP principles is provided in Appendix C.1 of this thesis.

The Kalman filter equations are shown in Formula 3.1. For a linear model and Gaussian noise statistics, the Kalman filter produces the *optimal* causal estimates $\hat{x}_k$ that minimize $J(x_k)$ in Equation 3.7. These estimates are optimal in both the MMSE and MAP senses. Maximum likelihood signal estimates are obtained by letting the initial covariance $\mathbf{P}_0$ approach infinity, thus causing the filter to ignore the value of the initial state $\hat{\mathbf{x}}_0$.

Nonlinear Models

When the autoregressive function $f(x_{k-1}, x_{k-2}, \ldots, x_{k-M}, \mathbf{w})$ in Equation 1.1 is *nonlinear*, then the Kalman filter equations can no longer be applied directly. The nonlinearity disrupts the Gaussianity of the statistics, making it impossible to obtain optimal estimates merely by propagating the mean and covariance of the posterior density.

For general (*i.e.*, non-Gaussian) densities, an optimal MMSE or MAP estimate can only be obtained by calculating the entire density function $\rho_{x_k\mathbf{y}_1^k}$ at each time step: a computationally

Initialize with:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0] \tag{3.15}$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \tag{3.16}$$

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

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} \tag{3.17}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{B}\sigma_v^2\mathbf{B}^T \tag{3.18}$$

and the measurement update equations:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1} \tag{3.19}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \tag{3.20}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^-. \tag{3.21}$$

Formula 3.1: The linear Kalman filter equations.

intractable task. Various approximations to the density can be calculated, however, with varying degrees of computational expense.

One of the more costly (and more exact) approaches are the sequential Monte Carlo algorithms, which sample many points from the posterior density function. The expense of these approaches comes largely from the need to propagate "clouds" of samples through the nonlinear function. A review is provided in [14].

Another approach to the nonlinear estimation problem is to approximate the conditional density with a Gaussian, and calculate only the covariance and mean, as before. While clearly inexact, these methods have a greatly reduced computational cost in comparison to the Monte Carlo sampling approach. Furthermore, the suboptimal solutions they generate are perfectly acceptable in many situations, particularly when the density remains unimodal, or when the nonlinearity is not severe.

The extended Kalman filter (EKF) is the most commonly used of these Gaussian-approximation methods. Under the Gaussian assumption, the estimation criterion is the same as expressed in Equation 3.7:

$$\hat{x}_k = \arg\min_{x_k}\left(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - \hat{x}_k^-)^2}{p_k^-}\right).$$

As before, calculating $\hat{x}_k^-$ and p_k^- requires $\hat{\mathbf{x}}_{k-1}$ and $\mathbf{P}_{k-1}$ from the previous time-step (the situation

of Figure 3.2 also holds for the nonlinear case). However, generating these statistics is problematic in the nonlinear case. The EKF produces *approximate* conditional means $\hat{\mathbf{x}}_{k-1}$, $\hat{\mathbf{x}}_k^-$, and covariances $\mathbf{P}_{k-1}$, and $\mathbf{P}_k^-$ by linearizing a set of nonlinear state-space equations[2].

Nonlinear State-Space Representation

A state-space representation for the more general nonlinear AR process is given by:

$$\mathbf{x}_k = \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w}) + \mathbf{B} \cdot v_k \tag{3.22}$$

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

$$\begin{aligned} y_k &= \mathbf{C} \cdot \mathbf{x}_k + n_k \\ &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \cdot \mathbf{x}_k + n_k \end{aligned} \tag{3.23}$$

where $\mathbf{F}(\cdot)$ has been introduced as a vector-valued function whose first element given by $f(\cdot)$, and whose remaining elements take on shifted values of the previous state.

Extended Kalman Filter

Under the Gaussian assumption, estimation of the posterior mean $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ from the prior statistics $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$ (*i.e.*, the measurement equations of Formula 3.1) is the same as in the linear case. However, generating prior mean $\hat{\mathbf{x}}_k^-$ and covariance $\mathbf{P}_k^-$ through the nonlinear function requires an approximation, as shown in Appendix D.

Defining $\mathbf{A}_k$ as:

$$\mathbf{A}_k \triangleq \frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{w})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_k} = \begin{bmatrix} \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{x}}^T \\ \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{matrix} \end{bmatrix}, \tag{3.24}$$

the EKF is obtained merely by replacing the KF time-update equations (3.17 and 3.18) with Formula 3.2. Note that this definition of $\mathbf{A}_k$ generalizes the definition of $\mathbf{A}$ in the linear case.

[2] A recently published algorithm called the *unscented filter* (UF) [35] offers a higher-order approximation to the mean and covariance. The UF is not discussed in this thesis.

$$\hat{\mathbf{x}}_k^- = \mathbf{F}(\hat{\mathbf{x}}_{k-1}, \mathbf{w}) \tag{3.25}$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{B}\sigma_v^2\mathbf{B}^T \tag{3.26}$$

Formula 3.2: The extended Kalman filter time-update equations.

3.2.3 Kalman Filter – Colored Noise Case

When the measurement noise, n_k, is *colored*, the KF and EKF equations require some modification. As discussed in Section 2.3.2, colored noise can be thought of as a second signal added to the first. In fact, because the weights $\mathbf{w}$ are assumed to be known in the present section, the only real distinction between x_k and n_k is that the signal x_k might be generated by nonlinear model, whereas n_k is assumed to be generated by a linear AR model.

Hence, the colored noise n_k ought to be estimated on equal footing with the signal. In the context of sequential MAP estimation, this means:

$$(\hat{x}_k, \hat{n}_k) = \arg\max_{x_k, n_k} \rho_{x_k n_k | \mathbf{y}_1^k \mathbf{w}}, \tag{3.27}$$

where the joint density can be expanded as:

$$\rho_{x_k n_k | \mathbf{y}_1^k \mathbf{w}} = \frac{\rho_{y_k | \mathbf{y}_1^{k-1} x_k n_k \mathbf{w}} \cdot \rho_{x_k n_k | \mathbf{y}_1^{k-1} \mathbf{w}} \cdot \rho_{\mathbf{y}_1^{k-1} | \mathbf{w}}}{\rho_{\mathbf{y}_1^k | \mathbf{w}}}. \tag{3.28}$$

However, due to the constraint $y_k = x_k + n_k$, the density $\rho_{y_k | \mathbf{y}_1^{k-1} x_k n_k \mathbf{w}}$ is therefore a Dirac delta function. Also, the densities $\rho_{\mathbf{y}_1^{k-1} | \mathbf{w}}$ and $\rho_{\mathbf{y}_1^k | \mathbf{w}}$ are functionally independent of x_k and n_k. Hence, maximizing $\rho_{x_k n_k | \mathbf{y}_1^k \mathbf{w}}$ is equivalent to maximizing $\rho_{x_k n_k | \mathbf{y}_1^{k-1} \mathbf{w}}$ subject to the constraint $y_k = x_k + n_k$. Furthermore, $\rho_{x_k n_k | \mathbf{y}_1^{k-1} \mathbf{w}}$ can be written as:

$$\rho_{x_k n_k | \mathbf{y}_1^{k-1} \mathbf{w}} = \rho_{x_k | \mathbf{y}_1^{k-1} \mathbf{w}} \cdot \rho_{n_k | \mathbf{y}_1^{k-1} \mathbf{w}} \tag{3.29}$$

under the assumption that the signal and noise are statistically independent. If the process noise terms v_k and $v_{n,k}$ are zero-mean Gaussian white noise, then:

$$\rho_{x_k n_k | \mathbf{y}_1^{k-1} \mathbf{w}} = \frac{\rho_{\mathbf{x}_0}}{\sqrt{2\pi p_k^-}} \exp\left(-\frac{(x_k - \hat{x}_k^-)^2}{2p_k^-}\right) \cdot \frac{1}{\sqrt{2\pi p_{n,k}^-}} \exp\left(-\frac{(n_k - \hat{n}_k^-)^2}{2p_{n,k}^-}\right),$$

where $\hat{n}_k^- = E[n_k | \{y_t\}_1^{k-1}, \mathbf{w}]$ and $p_{n,k}^- = E[(n_k - \hat{n}_k^-)^2 | \{y_t\}_1^{k-1}, \mathbf{w}]$. Taking the negative log of this expression, the corresponding cost is simply:

$$J(x_k, n_k) = \frac{(x_k - \hat{x}_k^-)^2}{p_k^-} + \frac{(n_k - \hat{n}_k^-)^2}{p_{n,k}^-}, \tag{3.30}$$

minimized subject to $y_k = x_k + n_k$. As in the white noise case, in order to generate the desired MAP estimates $\hat{x}_k$ and $\hat{n}_k$, it is first necessary to compute the prior statistics $\hat{x}_k$, p_k^-, $\hat{n}_k^-$ and $p_{n,k}^-$.

Linear Model

Starting with the linear-model case, the signal is assumed to be generated by the AR process of Equation 3.8, and the measurement noise is generated by a similar AR process (given in Equation 2.23). By defining vectors $\mathbf{x}_{k-1} = [x_{k-1}, \ldots, x_{k-M}]^T$ and $\mathbf{n}_{k-1} = [n_{k-1}, \ldots, n_{k-M_n}]^T$, the prior means and variances can be computed as:

$$\hat{x}_k^- = \mathbf{w}^T \hat{\mathbf{x}}_{k-1} \qquad p_k^- = \mathbf{w}^T \mathbf{P}_{k-1} \mathbf{w}$$
$$\hat{n}_k^- = \mathbf{w}_n^T \hat{\mathbf{n}}_{k-1} \qquad p_{n,k}^- = \mathbf{w}_n^T \mathbf{P}_{n,k-1} \mathbf{w}_n.$$

The derivation of these equations is directly analogous to that presented for the white noise case in Equations 3.9- 3.10 on page 46.

Hence, in addition to estimating the signal state $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ recursively, estimation of the noise state $\hat{\mathbf{n}}_k$ and covariance $\mathbf{P}_{n,k}$ is also required. This is achieved by formulating a state-space representation of the system, and applying a Kalman filter.

State-Space Representation – Colored Noise

Note that the constraint $y_k = x_k + n_k$ has some peculiar effects. Namely, the estimates $\hat{x}_k$ and $\hat{n}_k$ must also sum to y_k, and the variance p_k must be equal to the variance $p_{n,k}$.

To enforce these constraints, both the signal and noise are incorporated into a combined state-space representation:

$$\boldsymbol{\xi}_k = \mathbf{A}_c \cdot \boldsymbol{\xi}_{k-1} + \mathbf{B}_c \cdot \mathbf{v}_{c,k} \tag{3.31}$$
$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A}_n \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{n}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{B} & 0 \\ 0 & \mathbf{B}_n \end{bmatrix} \cdot \begin{bmatrix} v_k \\ v_{n,k} \end{bmatrix}$$

$$y_k = \mathbf{C}_c \cdot \boldsymbol{\xi}_k \tag{3.32}$$
$$y_k = [\mathbf{C} \quad \mathbf{C}_n] \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{bmatrix},$$

where

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

The *effective* measurement noise is zero, and the process noise $\mathbf{v}_{c,k}$ is white, as required, with covariance $\mathbf{V}_c = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_{v_n}^2 \end{bmatrix}$.

Kalman Filter – Colored Noise

The Kalman filter equations for the colored measurement noise case are shown in Formula 3.3. A potential problem with the algorithm is that the zero effective measurement noise can adversely effect the stability of the Kalman filter under some circumstances. Hence, adding a small positive value to the noise variance may be necessary in Equation 3.37.

Initialize with:

$$\hat{\boldsymbol{\xi}}_0 = E[\boldsymbol{\xi}_0] \tag{3.33}$$

$$\mathbf{P}_0 = E[(\boldsymbol{\xi}_0 - \hat{\boldsymbol{\xi}}_0)(\boldsymbol{\xi}_0 - \hat{\boldsymbol{\xi}}_0)^T] \tag{3.34}$$

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

$$\hat{\boldsymbol{\xi}}_k^- = \mathbf{A}_c \hat{\boldsymbol{\xi}}_{k-1} \tag{3.35}$$

$$\mathbf{P}_k^- = \mathbf{A}_c \mathbf{P}_{k-1} \mathbf{A}_c^T + \mathbf{B}_c \mathbf{V}_c \mathbf{B}_c^T \tag{3.36}$$

and the measurement update equations:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_c^T (\mathbf{C}_c \mathbf{P}_k^- \mathbf{C}_c^T + 0)^{-1} \tag{3.37}$$

$$\hat{\boldsymbol{\xi}}_k = \hat{\boldsymbol{\xi}}_k^- + \mathbf{K}_k (y_k - \mathbf{C}_c \hat{\boldsymbol{\xi}}_k^-) \tag{3.38}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_c) \mathbf{P}_k^- . \tag{3.39}$$

Formula 3.3: The linear Kalman filter equations for colored measurement noise.

An alternative approach suggested by Bryson and Henrikson [7] avoids this problem and also maintains the dimension of the original signal-state vector. Although this can improve the computational efficiency of the filter, the order of the noise model is restricted to be the same as the dimension of the measurement. In the context of time-series, this means the noise can only be modeled by an AR(1) process (recalling $\dim(y_k) = 1$), making this approach impractical for colored noise with higher-order correlations.

Nonlinear Model

As in the white noise case, the statistics of the signal are no longer Gaussian when the signal model is nonlinear, so an approximate solution is required. The EKF approach is essentially identical to the white noise case, except that the combined state-space representation is used to include the colored noise.

Nonlinear State-Space Representation – Colored Noise

$$\begin{aligned} \boldsymbol{\xi}_k &= \mathbf{F}_c(\boldsymbol{\xi}_{k-1}, \mathbf{w}, \mathbf{w}_n) + \mathbf{B}_c \cdot \mathbf{v}_{c,k} \\ \begin{bmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{bmatrix} &= \begin{bmatrix} \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w}) \\ \mathbf{A}_n \cdot \mathbf{n}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{B} & 0 \\ 0 & \mathbf{B}_n \end{bmatrix} \cdot \begin{bmatrix} v_k \\ v_{n,k} \end{bmatrix} \end{aligned} \tag{3.40}$$

$$\begin{aligned} y_k &= \mathbf{C}_c \cdot \boldsymbol{\xi}_k \\ y_k &= [\mathbf{C} \quad \mathbf{C}_n] \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{bmatrix} \end{aligned} \tag{3.41}$$

Extended Kalman Filter – Colored Noise

Seeing that the noise is still assumed to be generated by a linear AR process, the nonlinearity only affects the part of the state which contains the signal. For the signal component, the same approximations are made as in the white noise case. Using $\mathbf{A}_k$ as defined in Equation 3.24:

$$\mathbf{A}_k \triangleq \frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{w})}{\partial \mathbf{w}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_k},$$

the combined state-transition matrix:

$$\mathbf{A}_{c,k} \triangleq \begin{bmatrix} \mathbf{A}_k & 0 \\ 0 & \mathbf{A}_n \end{bmatrix} \tag{3.42}$$

is defined. The EKF can now be found for the combined state-space representation by replacing the time-update equations in Formula 3.3 with Formula 3.4.

$$\hat{\boldsymbol{\xi}}_k^- = \mathbf{F}_c(\hat{\boldsymbol{\xi}}_{k-1}, \mathbf{w}, \mathbf{w}_n) \tag{3.43}$$

$$\mathbf{P}_k^- = \mathbf{A}_{c,k-1}\mathbf{P}_{k-1}\mathbf{A}_{c,k-1}^T + \mathbf{B}_c\mathbf{V}_c\mathbf{B}_c^T \tag{3.44}$$

Formula 3.4: The extended Kalman filter time-update equations for colored measurement noise.

3.3 Weight Estimation

The estimation of model parameters from noisy data is a fairly difficult task, and is discussed later in this chapter. However, just as the signal can be estimated when the weights are assumed to be known, the weights can be easily estimated when the signal is known. This standard weight estimation problem is useful for introducing the Kalman and extended Kalman weight filters, as

well as several key concepts which a central to the development of the dual Kalman filter. In a MAP estimation context, the weight estimate $\hat{\mathbf{w}}_k$ is desired which is most probable given the signal $\{x_t\}_1^k$ up to and including the present time k. This is formally expressed as:

$$\hat{\mathbf{w}}_k = \arg\max_{\mathbf{w}} \rho_{\mathbf{w}|\mathbf{x}_1^k}. \tag{3.45}$$

3.3.1 Batch Estimation

Applying Bayes rule: $\rho_{\mathbf{w}|\mathbf{x}_1^k} = \frac{\rho_{\mathbf{x}_1^k|\mathbf{w}} \cdot \rho_{\mathbf{w}}}{\rho_{\mathbf{x}_1^k}}$, and assuming the prior $\rho_{\mathbf{w}}$ is uninformative, indicates that maximizing $\rho_{\mathbf{w}|\mathbf{x}_1^k}$ is equivalent to maximizing $\rho_{\mathbf{x}_1^k|\mathbf{w}}$. According to Appendix A, the density $\rho_{\mathbf{x}_1^k|\mathbf{w}}$ can be expanded as:

$$\rho_{\mathbf{x}_1^k|\mathbf{w}} = \frac{1}{\sqrt{(2\pi)^k(\sigma_v^2)^k}} \exp\left(-\sum_{t=1}^{k} \frac{(x_t - x_t^-)^2}{2\sigma_v^2}\right), \tag{3.46}$$

where $x_t^- = f(x_{t-1}, \ldots, x_{t-M}, \mathbf{w})$. Taking the negative log gives the *batch* cost function:

$$J(\mathbf{w}) = \sum_{t=1}^{k} \left(\log(2\pi\sigma_v^2) + \frac{(x_t - x_t^-)^2}{\sigma_v^2}\right), \tag{3.47}$$

where the log term can be dropped because σ_v^2 is assumed independent of $\mathbf{w}$. This leaves the sum of the squared prediction errors, normalized by the process noise variance σ_v^2:

$$J(\mathbf{w}) = \sum_{t=1}^{k} \frac{(x_t - x_t^-)^2}{\sigma_v^2}. \tag{3.48}$$

Minimizing this batch cost with respect to the weights produces such algorithms as least squares [29] in the linear-model case. and batch back-propagation [72, 95] for neural network models.

However, these learning algorithms are not appropriate for use in on-line applications. Although sequential approaches can be derived as variations of the batch algorithms (*e.g.*, recursive least squares (RLS) and "stochastic" backpropagation), a more rigorous derivation of a sequential algorithm within the MAP framework is provided next.

3.3.2 Sequential Weight Estimation

To develop a *sequential* MAP learning procedure, the density to be maximized is expanded as:

$$\rho_{\mathbf{w}|\mathbf{x}_1^k} = \frac{\overbrace{\rho_{x_k|\mathbf{x}_1^{k-1}\mathbf{w}} \cdot \rho_{\mathbf{w}|\mathbf{x}_1^{k-1}}}^{\rho_{x_k\mathbf{w}|\mathbf{x}_1^{k-1}}} \cdot \rho_{\mathbf{x}_1^{k-1}}}{\rho_{\mathbf{x}_1^k}}. \tag{3.49}$$

Because $\rho_{\mathbf{x}_1^k}$ is not a function of $\mathbf{w}$, the MAP estimate can be obtained by maximizing the first two terms in the numerator.

If the process noise v_k is a white Gaussian zero-mean process (assume no measurement noise in this section), then these terms evaluate to:

$$\rho_{x_k \mathbf{w}|\mathbf{x}_1^{k-1}} = \frac{1}{\sqrt{2\pi\sigma_v^2}} \exp\left(-\frac{(x_k - x_k^-)^2}{2\sigma_v^2}\right) \cdot \frac{1}{\sqrt{2\pi|\mathbf{Q}_k^-|}} \exp\left(-\frac{1}{2}(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T(\mathbf{Q}_k^-)^{-1}(\mathbf{w} - \hat{\mathbf{w}}_k^-)\right), \tag{3.50}$$

where $x_k^- = E[x_k|\{x_t\}_1^{k-1}, \mathbf{w}]$ and $\hat{\mathbf{w}}_k^- \triangleq E[\mathbf{w}|\{x_t\}_1^{k-1}]$,

and where $\mathbf{Q}_k^- \triangleq E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T|\{x_t\}_1^{k-1}]$.

Taking the negative log gives the following cost function:

$$J(\mathbf{w}) = \frac{(x_k - x_k^-)^2}{\sigma_v^2} + (\mathbf{w} - \hat{\mathbf{w}}_k^-)^T(\mathbf{Q}_k^-)^{-1}(\mathbf{w} - \hat{\mathbf{w}}_k^-), \tag{3.51}$$

which can be minimized with respect to $\mathbf{w}$ to produce the desired MAP estimate. The first term is the instantaneous squared prediction error. The second term in the cost keeps the new estimate close to the prior estimate $\hat{\mathbf{w}}_k^-$, which is based on the previous data. The prior covariance $\mathbf{Q}_k^-$ determines the distance metric used to define "close".

Note that the cost in Equation 3.51 is *equivalent* to the batch prediction error cost in Equation 3.48. However, by reformulating the cost in terms of the prior statistics $\hat{\mathbf{w}}_k^-$ and $\mathbf{Q}_k^-$, a recursive procedure can be derived: the prior weight estimate $\hat{\mathbf{w}}_k^-$ and covariance $\mathbf{Q}_k^-$ must be determined from the posterior statistics at the previous time step.

Linear Model

Measurement Equation for Weights

The case of a linear model is considered first: $x_k = \mathbf{w}_k^T\mathbf{x}_{k-1} + v_k$.

State-Space Representation for Weights

When developing a recursive weight estimation procedure, it is convenient to give the weights their own state-space representation. This is done by modeling the weights as a stationary process:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{u}_k \tag{3.52}$$

$$x_k = \mathbf{w}_k^T\mathbf{x}_{k-1} + v_k. \tag{3.53}$$

Note that the state transition matrix is identity, and that a Gaussian process noise vector $\mathbf{u}_k$ has been added with covariance $E[\mathbf{u}_k\mathbf{u}_k^T] = \mathbf{U}_k$ and cross-covariance $E[\mathbf{u}_k\mathbf{u}_j] = 0 \quad \forall k \neq j$. When $\mathbf{U}_k = 0$, the weight process is a constant deterministic process; otherwise, it is a random walk.

Even though the weights are not typically believed to undergo this sort of stochastic variation, the process noise $\mathbf{u}_k$ can be useful for increasing the tracking ability of the weight estimation filter. Roughly speaking, the larger the covariance $\mathbf{U}_k$, the more quickly older data is discarded; this relationship is described more fully beginning on the current page.

Using this model for $\mathbf{w}_k$ gives:

$$\hat{\mathbf{w}}_k^- = E[\mathbf{I} \cdot \mathbf{w}_{k-1} + \mathbf{u}_k | \{x_t\}_1^{k-1}] \tag{3.54a}$$

$$= \hat{\mathbf{w}}_{k-1} \tag{3.54}$$

$$\mathbf{Q}_k^- = E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T | \{x_t\}_1^{k-1}]. \tag{3.55a}$$

$$= E[(\mathbf{w}_{k-1} + \mathbf{u}_k - \hat{\mathbf{w}}_{k-1})(\mathbf{w}_{k-1} + \mathbf{u}_k - \hat{\mathbf{w}}_{k-1})^T | \{x_t\}_1^{k-1}]. \tag{3.55b}$$

$$= \mathbf{Q}_{k-1} + \mathbf{U}_k, \tag{3.55}$$

which gives the prior mean and covariance in terms of the posterior mean and covariance from the previous time step. To complete the recursive procedure, $\hat{\mathbf{w}}_k$ and $\mathbf{Q}_k$ must also be calculated from $\hat{\mathbf{w}}_k^-$, $\mathbf{Q}_k^-$, and the new measurement x_k.

The measurement equation (3.53) expresses the known signal x_k as an observation on the unknown weights $\mathbf{w}_k$. Note also that the known signal state $\mathbf{x}_{k-1}$ can be interpreted here as a time-varying parameter vector. Together, Equations 3.52 and 3.53 constitute a state-space representation for the weights.

Kalman Weight Filter

Using this state-space representation, a Kalman weight filter can be derived from the MAP perspective to minimize the cost $J(\mathbf{w})$ in Equation 3.51. The derivation is provided in Appendix C.2 and closely parallels that given in Appendix C.1 for signal estimation. The Kalman filter equations for recursively generating prior and posterior estimates and error covariances for the weights are compiled in Formula 3.5 on the following page. The algorithm can be viewed as a generalization of the popular *recursive least squares* (RLS) algorithm for linear parameter estimation [29].

Recursive Least Squares

More precisely, RLS is a special case of the Kalman weight filter when the covariance, $\mathbf{U}_k$, of the process noise is constrained in a certain way. Specifically,

$$\mathbf{U}_k = (\lambda^{-1} - 1)\mathbf{Q}_{k-1}, \quad \text{where } \lambda \in (0, 1], \tag{3.63}$$

Initialize with:

$$\hat{\mathbf{w}}_0 = E[\mathbf{w}] \tag{3.56}$$

$$\mathbf{Q}_0 = E[(\mathbf{w} - \hat{\mathbf{w}}_0)(\mathbf{w} - \hat{\mathbf{w}}_0)^T] \tag{3.57}$$

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

$$\hat{\mathbf{w}}_k^- = \hat{\mathbf{w}}_{k-1} \tag{3.58}$$

$$\mathbf{Q}_k^- = \mathbf{Q}_{k-1} + \mathbf{U}_k \tag{3.59}$$

and the measurement update equations:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^- \mathbf{x}_{k-1} (\mathbf{x}_{k-1}^T \mathbf{Q}_k^- \mathbf{x}_{k-1} + \sigma_v^2)^{-1} \tag{3.60}$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} (x_k - \mathbf{x}_{k-1}^T \hat{\mathbf{w}}_k^-) \tag{3.61}$$

$$\mathbf{Q}_k = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \mathbf{x}_{k-1}^T) \mathbf{Q}_k^-. \tag{3.62}$$

Formula 3.5: The linear Kalman weight filter equations.

causes Equation 3.59 to be replaced by

$$\mathbf{Q}_k^- = \lambda^{-1} \mathbf{Q}_{k-1}, \tag{3.64}$$

which prescribes that the prior covariance should be larger than the posterior covariance by a certain *percentage*, rather than by an additive amount $\mathbf{U}_k$. By defining:

$$\mathbf{\Sigma}_k \triangleq (\frac{1}{\sigma_v^2} \mathbf{Q}_k)^{-1} = (\frac{\lambda}{\sigma_v^2} \mathbf{Q}_{k+1}^-)^{-1}, \tag{3.65}$$

it is shown in Appendix C.2 that Equations 3.60-3.62 are equivalent to the RLS equations:

$$\mathbf{\Sigma}_k = \lambda \mathbf{\Sigma}_{k-1} + \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T \tag{3.66}$$

$$\boldsymbol{\beta}_k = \lambda \boldsymbol{\beta}_{k-1} + \mathbf{x}_{k-1} x_k \tag{3.67}$$

$$\hat{\mathbf{w}}_k = \mathbf{\Sigma}_k^{-1} \boldsymbol{\beta}_k. \tag{3.68}$$

These equations imply $\mathbf{\Sigma}_k = \sum_{t=1}^k \lambda^{k-t} \mathbf{x}_{t-1} \mathbf{x}_{t-1}^T$ and $\boldsymbol{\beta}_k = \sum_{t=1}^k \lambda^{k-t} \mathbf{x}_{t-1} x_t$. In RLS, λ is often called the *forgetting factor* because it controls the time constant of an exponential window over the data (see Figure 3.3). When $\lambda = 1$, all of the past data is weighted equally. The same effect is produced by $\mathbf{U}_k = 0$ in the Kalman weight filter (no process noise for the weights). This relationship motivates the use of process noise; non-zero $\mathbf{u}_k$ conveys the idea that the data in the distant past is no longer relevant for modeling the current dynamics. This enables the algorithm to effectively "forget" data in the past, and increases the algorithm's ability to track a changing

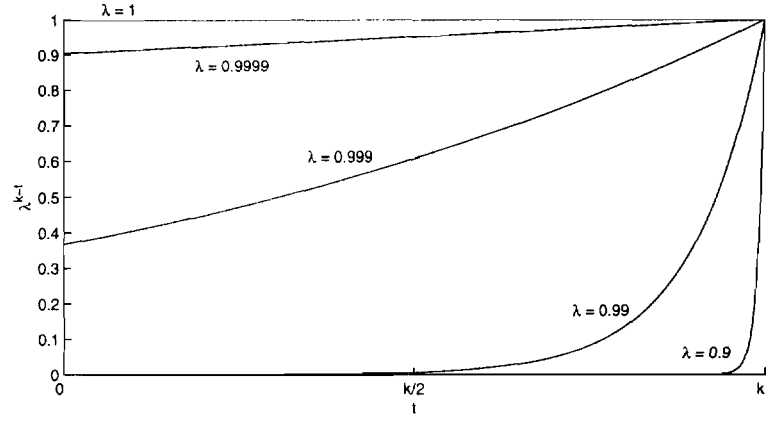


Figure 3.3: The gain produced for the data at times $t \leq k$ by various forgetting factors λ. Values displayed for $k = 1000$. The time constants can be computed as $\tau = -1/\log(\lambda)$.

system. However, this will also increase the variance of the weight estimates, because less data is being used.

The RLS algorithm is only equivalent to LS as $k \to \infty$, because RLS must be initialized with a positive-definite matrix $\mathbf{\Sigma}_0 = \sigma_v^2 \mathbf{Q}_0^{-1}$. The determinant of $\mathbf{\Sigma}_0$ must be large enough to produce a well-conditioned inversion in Equation 3.68, but small enough so as not to bias the result. From Equation 3.65, these comments also shed light on the role of σ_v^2 and $\mathbf{Q}_0^{-1}$ in the Kalman weight filter: their product should be chosen to give stable weight updates during the first few time-steps, without unduly biasing the estimates $\hat{\mathbf{w}}_k$. This issue is explored experimentally in the context of dual estimation in Chapter 4.

In the general context of parameter estimation, using $\lambda < 1$ is appropriate whenever the data exhibits some amount of nonstationarity. In this situation, either the weights, $\mathbf{w}$, or variances are drifting with time in some unspecified manner, so that older data do not accurately reflect the current model parameters. This parameter movement is appropriately modeled by a process noise term in the state-space equations for the parameters. However, too small a value of λ limits the amount of data being used to estimate the parameters. This increases the variance of the parameter estimates, making them less accurate.

In the context of linear parameter estimation on clean data, an analytic expression can be derived for the optimal value of λ, given information about the degree of nonstationarity [29]. This expression trades off error due to variance in the parameter estimates (called *noise misadjustment*) and error due to insufficient tracking (called *lag misadjustment*). However, it requires knowledge of the rate at which the system is changing.

Alternatively, rules for adapting λ can be derived. Other approaches in the literature are to

define $\mathbf{U}_k$ as a constant diagonal matrix [67], to estimate it from a moving average of the prediction errors [81], or to change it according to an annealing schedule [18]. However, these alternatives are not considered in the context of this thesis.

Nonlinear Model

Nonlinear State-Space Representation for Weights

For nonlinear models of the data, the state-space equations for $\mathbf{w}$ become:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{u}_k \tag{3.69}$$

$$x_k = f(\mathbf{x}_{k-1}, \mathbf{w}_k) + v_k. \tag{3.70}$$

where the measurement equation is expressed in terms of a *nonlinear* observation on $\mathbf{w}_k$, parameterized by the signal-state $\mathbf{x}_{k-1}$.

Extended Kalman Weight Filter

The weights of the nonlinear model can be estimated by applying an EKF to the nonlinear state equations (3.69-3.70). This requires linearizing the model with respect to the weights:

$$\mathbf{H}_k \triangleq \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{w})}{\partial \mathbf{w}}^T \Big|_{\mathbf{w}=\hat{\mathbf{w}}_k^-} \tag{3.71}$$

in order to calculate the Kalman gain and update the covariance. The measurement update equations in Formula 3.5 are replaced by Formula 3.6. The EKF for training neural networks was initially proposed by Singhal and Wu ([77],1989), and has been successfully applied and enhanced by numerous authors.

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{Q}_k^- \mathbf{H}_k^T + \sigma_v^2)^{-1} \tag{3.72}$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} (x_k - f(\hat{\mathbf{w}}_k^-, \mathbf{x}_{k-1})) \tag{3.73}$$

$$\mathbf{Q}_k = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \mathbf{H}_k) \mathbf{Q}_k^-. \tag{3.74}$$

Formula 3.6: The extended Kalman weight filter measurement-update equations.

Modified-Newton Method

The weight EKF can be interpreted as a modified-Newton optimization method [48], which performs an approximate second-order search over the surface of the squared-prediction-error cost

function. To see this, note that the weight update

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}}(x_k - f(\hat{\mathbf{w}}_k^-, \mathbf{x}_{k-1})) \tag{3.75}$$

can be rewritten (using Equation C.49 in Appendix C.2) as:

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{Q}_k \mathbf{H}_k^T \sigma_v^{-2}(x_k - f(\hat{\mathbf{w}}_k^-, \mathbf{x}_{k-1})). \tag{3.76}$$

Also, an alternative form for the covariance recursion is derived in Appendix C.2, as:

$$\mathbf{Q}_k^{-1} = ((\mathbf{Q}_k^-)^{-1} + \mathbf{H}_k^T \sigma_v^{-2} \mathbf{H}_k) \tag{3.77a}$$

$$= (\lambda^{-1}\mathbf{Q}_{k-1})^{-1} + \mathbf{H}_k^T \sigma_v^{-2} \mathbf{H}_k. \tag{3.77}$$

In a modified-Newton algorithm, the weight update takes the form:

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- - \mathbf{S}_k \nabla_{\mathbf{w}} J(\hat{\mathbf{w}}_k^-)^T \tag{3.78}$$

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 typically approximates the inverse Hessian of the cost. Both the gradient and Hessian are of course evaluated at the previous value of the weight estimate, $\hat{\mathbf{w}}_k^-$.

If J is the batch form of the squared-prediction-error cost in Equation 3.48 on page 55:

$$J = \sum_{t=1}^{k} \frac{(x_t - f(\mathbf{x}_{t-1}, \mathbf{w}))^2}{\sigma_v^2} \tag{3.79}$$

then the gradient and Hessian are given as:

$$\nabla_{\mathbf{w}} J = -2 \sum_{t=1}^{k} \mathbf{H}_t^T \sigma_v^{-2}(x_t - f(\mathbf{w}, \mathbf{x}_{t-1})), \tag{3.80}$$

$$\text{and} \qquad \nabla_{\mathbf{w}}^2 J = 2 \sum_{t=1}^{k} \mathbf{H}_t^T \sigma_v^{-2} \mathbf{H}_t + o(2), \tag{3.81}$$

where $o(2)$ represents terms involving the second derivative of the cost with respect to $\mathbf{w}$. For a linear model and the prediction-error cost, $o(2) = 0$.

Equation 3.77 can be rewritten in closed form as:

$$\mathbf{Q}_k^{-1} = \lambda \mathbf{Q}_0^{-1} + 2 \sum_{t=1}^{k} \lambda^{k-t} \mathbf{H}_t^T \sigma_v^{-2} \mathbf{H}_t, \tag{3.82}$$

to express $\mathbf{Q}_k^{-1}$ as a first-order approximation to (one-half) the Hessian. Furthermore, the expression $\mathbf{H}_k^T \sigma_v^{-2}(x_k - f(\hat{\mathbf{w}}_k^-, \mathbf{x}_{k-1}))$ in Equation 3.76 is an instantaneous (or stochastic) approximation

to (one-half) the negative gradient; equivalently, it is the negative gradient of the ***instantaneous cost***: $J_k = \frac{1}{2}\sigma_v^{-2}(x_k - f(\mathbf{w}, \mathbf{x}_{k-1}))^2$.

The EKF weight-update expressed in Equation 3.76 can therefore be interpreted as an on-line form of the modified-Newton optimization scheme for minimizing the batch prediction error cost. The scale factors of $\frac{1}{2}$ cancel out in the weight update. Note that whereas the vectors $\mathbf{H}_t$ used to build up the inverse covariance in Equation 3.82 are evaluated using a different value of $\hat{\mathbf{w}}_t^-$ at each time step, $\mathbf{Q}_k$ is therefore only an *approximation* to the inverse of the first-order Hessian. All of the values of $\mathbf{H}_t$ in the true Hessian expressed in Equation 3.81 should be calculated using the *same* value of $\hat{\mathbf{w}}_k^-$.

Typically, the weight covariance is initialized as $\mathbf{Q}_0 = q \cdot \mathbf{I}$, where q is a positive scalar reflecting the expected numerical range taken by the parameters, and $\mathbf{I}$ is the identity matrix. If the time-series data has been normalized to unit variance, then it is usually reasonable to assume unit variance on the parameter values ($q = 1$). However, the value of σ_v^2 can influence the choice of q.

The prediction error variance σ_v^2 scales *both* the approximate gradient and approximate Hessian terms, so it effectively cancels out of the weight update. However, because it acts as a scaling term, σ_v^2 will determine the relative influence of the initial covariance $\mathbf{Q}_0$ on later covariances $\mathbf{Q}_k$, for $k > 0$. When $\lambda = 1$, Equation 3.77 can be written equivalently as $\sigma_v^2\mathbf{Q}_k^{-1} = \sigma_v^2(\mathbf{Q}_{k-1})^{-1} + \mathbf{H}_k^T\mathbf{H}_k$, so that when σ_v^2 is small, $(\mathbf{Q}_0)^{-1}$ should be large to keep $\sigma_v^2(\mathbf{Q}_1)^{-1}$ invertible. The situation is similar to the RLS initialization in the linear case, where the condition number of $\sigma_v^2(\mathbf{Q}_0)^{-1}$ is also crucial. A small value of q will cause $\mathbf{Q}_0^{-1}$ to have large diagonal values and produce more stable (lower variance) behavior, but this will bias the estimates $\hat{\mathbf{w}}_k$ for small times k. Ultimately, then, the choice of q will depend on the variance of the process noise σ_v^2, and the variance of data.

Observed-Error Form

The preceding paragraphs demonstrate that the weight EKF provides a modified-Newton optimization algorithm for the squared-prediction-error cost function given in Equation 3.79. This result can be readily generalized to produce recursive algorithms for minimizing *other* batch cost functions, by simply rewriting the observation equation for the weights. The basic idea is presented by Puskorius and Feldkamp [67] for minimizing an entropic cost function.

From the standpoint of the modified-Newton update, the exact choice of state-space representation for the weights is irrelevant, so long as good approximations to the Hessian and gradient

are generated. Therefore, consider reformulating the state-space representation for the weights as:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{u}_k \tag{3.83}$$

$$0 = -\mathbf{e}_k + r_k. \tag{3.84}$$

where r_k is a measurement noise term with variance $\sigma_r^2 = \frac{1}{2}$, and the target "observation" is fixed at zero. The measurement function $-\mathbf{e}_k$ is chosen according to the cost function to be minimized, such that $\mathbf{e}_k^T \mathbf{e}_k = J_k$. However, this alone does not uniquely specify $\mathbf{e}_k$, which can be vector-valued.

Applying the alternative form of the Kalman weight filter update (in Equations 3.76-3.77) to the observed-error state-space representation, gives:

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{Q}_{o,k}\mathbf{H}_{o,k}^T \sigma_r^{-2}(0 + \mathbf{e}_k), \tag{3.85}$$

$$\text{where} \quad \mathbf{H}_{o,k} \triangleq \left.\frac{\partial(-\mathbf{e}_k)}{\partial \mathbf{w}}\right|_{\mathbf{w}=\hat{\mathbf{w}}_k^-}^T \tag{3.86}$$

$$\text{and} \quad \mathbf{Q}_{o.k}^{-1} = (\lambda^{-1}\mathbf{Q}_{o,k-1})^{-1} + \mathbf{H}_{o,k}^T \sigma_r^{-2}\mathbf{H}_{o,k}. \tag{3.87}$$

The error $\mathbf{e}_k$ is chosen such that $\mathbf{H}_{o,k}^T \sigma_r^{-2}\mathbf{e}_k$ and $\mathbf{Q}_{o.k}$ produce the negative gradient and inverse Hessian of the desired batch cost[3].

In [67], Puskorius and Feldkamp propose using this observed-error filter to minimize an entropic cost function[4]. The instantaneous cost is:

$$J_k = 2\log\frac{2}{1 + y_k\hat{y}_k}, \tag{3.88}$$

where $\hat{y}_k$ represents the output of the model at time k. The observed-error and its negative derivative are defined as $\mathbf{e}_k = \sqrt{J_k}$, and

$$\mathbf{H}_{o,k} = \frac{y_k}{J_k^{\frac{1}{2}}(1 + y_k\hat{y}_k)} \cdot \nabla_{\mathbf{w}}\hat{y}_k \tag{3.89}$$

to perform minimization of the entropic cost.

Returning to the prediction-error cost of Equation 3.48, let $\mathbf{e}_k = \sqrt{J_k} = (x_k - f(\mathbf{x}_{k-1}, \mathbf{w}_k))/\sigma_v$, so that

$$\mathbf{H}_{o,k} = \frac{-1}{2\sqrt{J_k}}\nabla_{\mathbf{w}}^T J_k(\hat{\mathbf{w}}_k) \quad = \frac{1}{\sigma_v}\nabla_{\mathbf{w}}^T f(\mathbf{x}_{k-1}, \hat{\mathbf{w}}_k) \quad = \frac{1}{\sigma_v}\mathbf{H}_k, \tag{3.90}$$

and $\mathbf{Q}_{o,k}^{-1}$ approximates $2\sum_{t=1}^{N}\mathbf{H}_t^T\sigma_v^{-2}\mathbf{H}_t$. Hence, $\mathbf{Q}_{o,k} = 2\mathbf{Q}_k$, giving the same approximation to the inverse Hessian as before, and $\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{e}_k = 2\mathbf{H}_k^T\sigma_v^{-2}(x_k - f(\mathbf{x}_{k-1}, \mathbf{w}_k))$ is the negative of

[3]While $-\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{e}_k$ gives the exact gradient of J_k, $\mathbf{Q}_{o.k}^{-1}$ is a recursive approximation to the first-order part of the Hessian.

[4]In [67], the outputs are constrained to be ± 1, and the cost allows for models with multiple outputs, but this is not important here.

the instantaneous gradient. The observed-error formulation for the prediction-error cost function therefore gives the same update as the standard Kalman weight filter in Formulae 3.5 and 3.6.

Moreover, by using the observed-error representation of the weights, a sequential weight estimation procedure can be designed for any cost J that can be written as a sum of instantaneous costs J_k. The only changes to the standard weight EKF are a redefinition of the error as $\mathbf{e}_k$ such that $\mathbf{e}_k^T \mathbf{e}_k = J_k$, a corresponding use of the output error derivative, $\mathbf{H}_{o,k}$, and the replacement of σ_v^2 with $\sigma_r^2 = \frac{1}{2}$. This last substitution means the initial weight covariance $\mathbf{Q}_0$ can now be chosen independently of σ_v^2. However, the primary advantage of the observed-error form is that any cost J_k can be minimized, so long as it is differentiable and nonnegative. As will be shown in Section 3.5, the dual Kalman filter relies on this form of the weight filter for minimizing many of the cost functions derived in Chapter 2.

3.4 Joint Estimation

Section 3.2 considered the problem of estimating the signal from noisy data when the model is known, and Section 3.3 considered the problem of estimating the model when the signal is known. The present section addresses the more complex problem of estimating *both* the signal and the model from noisy data, when neither one is known. Here, the two unknown quantities are estimated by combining them in a joint state-space representation; the dual Kalman filtering approach, which treats them separately, is described in Section 3.5.

3.4.1 Joint Kalman Filtering – White Noise Case

Recall the joint cost function from Equation 2.11 on page 23. derived in Section 2.3:

$$J(\mathbf{x}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \Big(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - \hat{x}_k^-)^2}{\sigma_v^2} \Big),$$

which can be minimized to produce the most probable estimates of the signal $\{x_k\}_1^N$ and weights $\mathbf{w}$ given the noisy data $\{y_k\}_1^N$.

However, in a sequential MAP estimation context, only the *current* estimates $\hat{x}_k$ and $\hat{\mathbf{w}}$ are desired, rather than an estimate of the entire signal. This goal is formally written as:

$$\big(\hat{x}_k, \hat{\mathbf{w}}_k\big) = \arg\max_{x_k, \mathbf{w}} \rho_{x_k \mathbf{w} | \mathbf{y}_1^k}. \tag{3.91}$$

The sequential estimates $\hat{x}_N$ and $\hat{\mathbf{w}}_N$ defined in this way will also be optimal with respect to the above batch cost function, as desired.

The previous two sections showed that sequential estimation procedures can be produced independently for x_k and $\mathbf{w}$ by first creating a state-space representation for each. To generate MAP estimates of x_k and $\mathbf{w}$ *simultaneously*, it is useful to define a new joint state-space representation. Defining:

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \end{bmatrix}, \tag{3.92}$$

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 $J(\mathbf{x}_1^k, \mathbf{w})$. Furthermore, the resulting state-space representation for $\mathbf{z}_k$ enables the development of a sequential estimation procedure.

Linear Model

To develop the joint state-space representation, first assume a linear model of the data:

$$x_k = \mathbf{w}_k^T \mathbf{x}_{k-1} + v_k \tag{3.93}$$

$$y_k = x_k + n_k, \tag{3.94}$$

where, as before, the weights are modeled as a stationary stochastic process:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{u}_k. \tag{3.95}$$

Joint State-Space Representation

The state-space equations for the joint state are:

$$\begin{aligned} \mathbf{z}_k &= \overline{\mathbf{F}}(\mathbf{z}_{k-1}) + \overline{\mathbf{B}}(v_k, \mathbf{u}_k) \\ \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{k-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{w}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \cdot v_k \\ \mathbf{u}_k \end{bmatrix} \end{aligned} \tag{3.96}$$

$$\begin{aligned} y_k &= \overline{\mathbf{C}} \cdot \mathbf{z}_k + n_k \\ y_k &= \begin{bmatrix} \mathbf{C} & 0 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \end{bmatrix} + n_k \end{aligned} \tag{3.97}$$

where, as before: $\mathbf{A}_k \triangleq \begin{bmatrix} \mathbf{w}_k^T \\ \mathbf{I} \quad 0 \end{bmatrix}$. While the above system *looks* linear in form, the multiplication $\mathbf{A}_{k-1} \cdot \mathbf{x}_{k-1}$ represents a nonlinear (or more precisely, *bilinear*) function of the joint state, $\mathbf{z}_{k-1}$.

This precludes the use of the Kalman filter for state estimation, even though the form of the model was assumed linear. However, an EKF can be applied to generate approximate MAP estimates of the signal and weights. This approach seems to have been developed first in [38, 12].

Joint Extended Kalman Filter

To apply the EKF, $\overline{\mathbf{F}}(\mathbf{z}_k)$ must be linearized with respect to the joint state $\mathbf{z}_k$, evaluated at the estimate $\hat{\mathbf{z}}_k$. Using the definition:

$$\overline{\mathbf{A}}_k \triangleq \frac{\partial \overline{\mathbf{F}}(\mathbf{z})}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\hat{\mathbf{z}}_k} = \begin{bmatrix} \mathbf{A}_k & \begin{bmatrix} \hat{\mathbf{x}}_k^T \\ 0 \quad 0 \end{bmatrix} \\ 0 & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \hat{\mathbf{w}}_k^T \\ \mathbf{I} \quad 0 \end{bmatrix} & \begin{bmatrix} \hat{\mathbf{x}}_k^T \\ 0 \quad 0 \end{bmatrix} \\ 0 & \mathbf{I} \end{bmatrix}, \tag{3.98}$$

and introducing the joint noise covariance:

$$\overline{\mathbf{V}}_k \triangleq Cov \begin{bmatrix} \mathbf{B}v_k \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{B}\sigma_v^2\mathbf{B}^T & 0 \\ 0 & \mathbf{U} \end{bmatrix} \tag{3.99}$$

the derivation of the joint EKF is exactly analogous to that of the standard Kalman filter given in Appendix C. The equations are given in Formula 3.7.

Initialize with:

$$\hat{\mathbf{z}}_0 = E[\mathbf{z}_0] \tag{3.100}$$

$$\mathbf{P}_0 = E[(\mathbf{z}_0 - \hat{\mathbf{z}}_0)(\mathbf{z}_0 - \hat{\mathbf{z}}_0)^T] \tag{3.101}$$

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

$$\hat{\mathbf{z}}_k^- = \overline{\mathbf{F}}(\hat{\mathbf{z}}_{k-1}) \tag{3.102}$$

$$\mathbf{P}_k^- = \overline{\mathbf{A}}_{k-1}\mathbf{P}_{k-1}\overline{\mathbf{A}}_{k-1}^T + \overline{\mathbf{V}}_k \tag{3.103}$$

and the measurement update equations:

$$\overline{\mathbf{K}}_k = \mathbf{P}_k^-\overline{\mathbf{C}}^T(\overline{\mathbf{C}}\mathbf{P}_k^-\overline{\mathbf{C}}^T + \sigma_n^2)^{-1} \tag{3.104}$$

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_k^- + \overline{\mathbf{K}}_k(y_k - \overline{\mathbf{C}}\hat{\mathbf{z}}_k^-) \tag{3.105}$$

$$\mathbf{P}_k = (\mathbf{I} - \overline{\mathbf{K}}_k\overline{\mathbf{C}})\mathbf{P}_k^-. \tag{3.106}$$

Formula 3.7: The joint extended Kalman filter equations.

Nonlinear Model

When the time-series is generated by a nonlinear AR process, the only change in the joint EKF comes from redefining $\overline{\mathbf{F}}(\mathbf{z}_{k-1})$ in Equation 3.96 as:

$$\overline{\mathbf{F}}(\mathbf{z}_{k-1}) \triangleq \begin{bmatrix} \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{I} \cdot \mathbf{w}_{k-1} \end{bmatrix} \tag{3.107}$$

and consequently letting:

$$\overline{\mathbf{A}}_k \triangleq \frac{\partial \overline{\mathbf{F}}(\mathbf{z})}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\hat{\mathbf{z}}_k} = \begin{bmatrix} \mathbf{A}_k & \begin{bmatrix} \mathbf{H}_k \\ 0 \end{bmatrix} \\ 0 & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{x}}^T \end{bmatrix} & \begin{bmatrix} \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{w}}^T \end{bmatrix} \\ \begin{matrix} \mathbf{I} & 0 \end{matrix} & \begin{matrix} 0 & 0 \end{matrix} \\ 0 & \mathbf{I} \end{bmatrix}. \tag{3.108}$$

Each of these definitions is consistent with the linear-model case; using them therefore allows the joint EKF algorithm in Formula 3.7 to be used for both linear and nonlinear models.

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 $\mathbf{w}$ (see Section 3.6.1 on page 102). This fact is cited in [45, 47] as a potential source of convergence problems for the joint EKF. Additional results and citations in [61] corroborate the difficulties of the approach, although the cause of divergence is linked therein to the linearization of the coupled system, rather than the lack of recurrent derivatives. Although the use of recurrent derivatives is suggested in [45, 47], there is no justification for this from the standpoint of minimizing the joint cost function. Furthermore, no divergence problems were encountered by this author during preparation of the experimental results in Chapter 4 when using non-recursive derivatives.

3.4.2 Joint Kalman Filtering – Colored Noise Case

As discussed in Section 3.2, when the measurement noise is colored, it must be estimated as though it were a second signal. The joint cost function for colored noise was given in Equation 2.29 on page 30 as:

$$J(\mathbf{x}_1^N, \mathbf{n}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \left(\frac{(x_k - x_k^-)^2}{\sigma_v^2} + \frac{(n_k - n_k^-)^2}{\sigma_{v_n}^2} \right), \tag{3.109}$$

which when minimized subject to the constraint $\{y_k\}_1^N = \{x_k\}_1^N + \{n_k\}_1^N$, produces the most probable estimates of the signal, noise, and weights given the data. However, the goal of *sequential* estimation is to find current estimates $\hat{x}_k, \hat{n}_k$, and $\hat{\mathbf{w}}_k$ such that:

$$(\hat{x}_k, \hat{n}_k, \hat{\mathbf{w}}_k) = \arg \max_{x_k, n_k, \mathbf{w}} \rho_{x_k, n_k, \mathbf{w} | \mathbf{y}_1^k}. \tag{3.110}$$

These estimates are optimal with respect to the batch cost function $J(\mathbf{x}_1^k, \mathbf{n}_1^k, \mathbf{w})$ used with data up to time k.

As in the white noise case, a state-space representation facilitates the development of a sequential algorithm. This time, define the joint state vector as:

$$\boldsymbol{\zeta}_k \triangleq \begin{bmatrix} \mathbf{x}_k \\ \mathbf{n}_k \\ \mathbf{w}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{\xi}_k \\ \mathbf{w}_k \end{bmatrix}. \tag{3.111}$$

Maximizing the density $\rho_{\boldsymbol{\zeta}_k|\mathbf{y}_1^k}$ with respect to $\boldsymbol{\zeta}_k$ will produce the desired $\hat{x}_k$, $\hat{n}_k$, and $\hat{\mathbf{w}}_k$ that minimize $J(\mathbf{x}_1^k, \mathbf{n}_1^k, \mathbf{w})$.

Linear Model

Starting with the linear-model case, the signal is assumed to be generated by the AR process of Equation 3.8, and the measurement noise is generated by a similar AR process (given in Equation 2.23).

Joint State-Space Representation · Colored Noise

The estimation of $\boldsymbol{\zeta}_k$ can be done recursively with an extended Kalman filter by writing the state-space equations for the joint state:

$$\begin{aligned} \boldsymbol{\zeta}_k &= \overline{\mathbf{F}}_c(\boldsymbol{\zeta}_{k-1}) + \overline{\mathbf{B}}_c(v_k, v_{n,k}, \mathbf{u}_k) \\ \begin{bmatrix} \boldsymbol{\xi}_k \\ \mathbf{w}_k \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{c,k-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\xi}_{k-1} \\ \mathbf{w}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_c \cdot \mathbf{v}_{c,k} \\ \mathbf{u}_k \end{bmatrix} \end{aligned} \tag{3.112}$$

$$\begin{aligned} y_k &= \overline{\mathbf{C}}_c \cdot \boldsymbol{\zeta}_k \\ y_k &= \begin{bmatrix} \mathbf{C} & \mathbf{C}_n & 0 \cdots 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\xi}_k \\ \mathbf{w}_k \end{bmatrix} \end{aligned} \tag{3.113}$$

where, as before: $\mathbf{A}_{c,k} \triangleq \left[\begin{smallmatrix} \mathbf{A}_k & 0 \\ 0 & \mathbf{A}_n \end{smallmatrix}\right]$, where $\mathbf{A}_k$ is given in Equation 3.24 on page 50. Hence, as in the white noise case, the multiplication $\mathbf{A}_{k-1} \cdot \mathbf{x}_{k-1}$ represents a bilinear function of the state, $\boldsymbol{\zeta}_{k-1}$. The state equations are therefore nonlinear, and an extended Kalman filter is needed for estimation of the signal, noise, and weights.

Joint Extended Kalman Filter -- Colored Noise

To apply the EKF, $\overline{\mathbf{F}}_c(\boldsymbol{\zeta}_k)$ must be linearized with respect to the joint-state $\boldsymbol{\zeta}_k$, evaluated at the estimate $\hat{\boldsymbol{\zeta}}_k$. Using the definition:

$$\overline{\mathbf{A}}_{c,k} \triangleq \frac{\partial \overline{\mathbf{F}}_c(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Big|_{\boldsymbol{\zeta}=\hat{\boldsymbol{\zeta}}_k} = \begin{bmatrix} \mathbf{A}_{c,k} & \begin{bmatrix} \hat{\mathbf{x}}_k^T \\ 0 \quad 0 \end{bmatrix} \\ 0 & \mathbf{I} \end{bmatrix} \tag{3.114}$$

and introducing the joint noise covariance:

$$\overline{\mathbf{V}}_{c,k} \triangleq Cov \begin{bmatrix} \mathbf{B}_c \mathbf{v}_{c,k} \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{B}\sigma_v^2 \mathbf{B}^T & 0 & 0 \\ 0 & \mathbf{B}_n \sigma_{v_n}^2 \mathbf{B}_n^T & 0 \\ 0 & 0 & \mathbf{U} \end{bmatrix} \tag{3.115}$$

allows the derivation of the colored noise joint EKF shown in Formula 3.8.

Initialize with:

$$\hat{\boldsymbol{\zeta}}_0 = E[\boldsymbol{\zeta}_0] \tag{3.116}$$

$$\mathbf{P}_0 = E[(\boldsymbol{\zeta}_0 - \hat{\boldsymbol{\zeta}}_0)(\boldsymbol{\zeta}_0 - \hat{\boldsymbol{\zeta}}_0)^T] \tag{3.117}$$

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

$$\hat{\boldsymbol{\zeta}}_k^- = \overline{\mathbf{F}}_c(\hat{\boldsymbol{\zeta}}_{k-1}) \tag{3.118}$$

$$\mathbf{P}_k^- = \overline{\mathbf{A}}_{c,k-1} \mathbf{P}_{k-1} \overline{\mathbf{A}}_{c,k-1}^T + \overline{\mathbf{V}}_{c,k} \tag{3.119}$$

and the measurement update equations:

$$\overline{\mathbf{K}}_k = \mathbf{P}_k^- \overline{\mathbf{C}}_c^T (\overline{\mathbf{C}}_c \mathbf{P}_k^- \overline{\mathbf{C}}_c^T + 0)^{-1} \tag{3.120}$$

$$\hat{\boldsymbol{\zeta}}_k = \hat{\boldsymbol{\zeta}}_k^- + \overline{\mathbf{K}}_k (y_k - \overline{\mathbf{C}}_c \hat{\boldsymbol{\zeta}}_k^-) \tag{3.121}$$

$$\mathbf{P}_k = (\mathbf{I} - \overline{\mathbf{K}}_k \overline{\mathbf{C}}_c) \mathbf{P}_k^- . \tag{3.122}$$

Formula 3.8: The joint extended Kalman filter equations for colored measurement noise.

Nonlinear Model – Colored Noise

When the time-series is generated by a nonlinear AR process, the only change in the joint EKF comes from redefining $\overline{\mathbf{F}}_c(\boldsymbol{\zeta}_{k-1})$ in Equation 3.96 as:

$$\overline{\mathbf{F}}_c(\boldsymbol{\zeta}_{k-1}) \triangleq \begin{bmatrix} \mathbf{F}_c(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{I} \cdot \mathbf{w}_{k-1} \end{bmatrix} \tag{3.123}$$

and consequently letting:

$$\overline{\mathbf{A}}_{c,k} \triangleq \frac{\partial \overline{\mathbf{F}}_c(\zeta)}{\partial \zeta}\Big|_{\zeta=\hat{\zeta}_k} = \begin{bmatrix} \mathbf{A}_{c,k} & \begin{bmatrix} \mathbf{H}_k \\ 0 \end{bmatrix} \\ 0 & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \mathbf{A}_k & 0 \\ 0 & \mathbf{A}_{n,k} \end{bmatrix} & \begin{bmatrix} \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{w}}^T \\ 0 \quad 0 \end{bmatrix} \\ 0 & \mathbf{I} \end{bmatrix}. \tag{3.124}$$

Both of these definitions are consistent with the linear-model case.

3.5 Dual Kalman Filtering

In the previous section, the joint EKF algorithm was described as a method for sequentially estimating both the signal and the model from noisy data. Because the joint cost function is a highly coupled function of its arguments, the joint EKF estimates the signal and weights simultaneously by combining them in a joint state-space representation. In this section, an alternative algorithm called the *dual extended Kalman filter* is developed by decomposing the problem into separate signal-estimation and weight-estimation components.

One powerful advantage of the dual EKF is that it can be applied to a variety of estimation cost functions. That is, the various costs derived in Chapter 2 can all be minimized sequentially by dual EKF algorithms. Although some costs do not require it, this is accomplished most generally through the observed-error form of the weight filter, described in Section 3.3.2. The joint EKF presented in the previous section lacks this flexibility, and can only minimize the joint cost $J^j(\mathbf{x}_k, \mathbf{w})$. The various cost functions and their observed-error variable definitions are presented throughout this section; Table 3.1 provides a summary. The form of the algorithms differ slightly for white noise and colored noise cases, so they are treated separately, in Sections 3.5.1 and 3.5.2, respectively.

3.5.1 White Noise Case

Section 2.3 derived the joint cost function for estimating $\{x_k\}_1^N$ and $\mathbf{w}$ in the presence of white Gaussian measurement noise as (from Equation 2.11):

$$J^j(\mathbf{x}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \Big(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - x_k^-)^2}{\sigma_v^2} \Big),$$

where $x_k^- = E[x_k | \{x_t\}_1^{k-1}, \mathbf{w}]$ is the optimal prediction, and is a function of both the signal and weights: $x_k^- = f(x_{k-1}, \ldots, x_{k-M}, \mathbf{w})$.

Table 3.1: Summary of the observed-error formulae for the various weight and variance cost functions minimized by the dual Kalman filter. When equations differ for the colored noise case, formula numbers are enclosed in parentheses.

	Name of Cost	Symbol	Formula	Page
Joint	joint weight	$J^j(\hat{\mathbf{x}}_1^N, \mathbf{w})$	3.9(3.22)	73(91)
	joint variance	$J^j(\sigma^2)$	3.11(3.23)	76(92)
	error-coupled weight	$J^{ec}(\mathbf{w})$	3.14(3.24)	81(94)
	error-coupled variance	$J^{ec}(\sigma^2)$	3.15(3.25)	81(95)
Marginal	prediction error	$J^{pe}(\mathbf{w})$	3.16(3.27)	84(98)
	prediction error	$J^{pe}(\sigma^2)$	n.a.	84(98)
	max. likelihood	$J^{ml}(\mathbf{w})$	3.17	84
	max. likelihood	$J^{ml}(\sigma^2)$	3.18	85
	EM weight	$J^{em}(\mathbf{w})$	3.19(3.28)	88(100)
	EM variance	$J^{em}(\sigma^2)$	3.20(n.a.)	89(100)

Decoupling with Direct Substitution

As discussed in Chapter 2, a *common* approach to minimizing a multivariate cost function is to optimize one argument at a time while the other argument is fixed. This can be done in an iterative framework (see Figure 1.5(a) on page 10) by first minimizing $J^j(\hat{\mathbf{x}}_1^N, \mathbf{w})$ with respect to $\mathbf{w}$ to produce $\hat{\mathbf{w}}$, and then minimizing $J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$ with respect to $\mathbf{x}_1^N$ to produce $\hat{\mathbf{x}}_1^N$, and repeating until the algorithm converges to a final set of estimates. Denoting the iteration index as i, the iterative approach can be viewed as a minimization of two sequences of cost functions, $\{J^j(\hat{\mathbf{x}}_1^N, \mathbf{w})\}_{i=1}^{\infty}$ and $\{J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})\}_{i=1}^{\infty}$, each of which converges to the cost $J^j(\mathbf{x}_1^N, \mathbf{w})$ as the estimates $\hat{\mathbf{x}}_1^N$ and $\hat{\mathbf{w}}$ converge to their true values. The errors-in-variables (EIV) framework in the statistics literature [75, 87] is an example of this iterative approach, and is described briefly in Appendix G.

In sequential dual estimation, on the other hand, a different cost function is effectively used at each *time*-step k. For a sequential approach, only the current state $\hat{\mathbf{x}}_k$ is optimized with respect to the current cost $J^j(\mathbf{x}_1^k, \hat{\mathbf{w}}_k)$; the sequence of costs $\{J^j(\mathbf{x}_1^k, \hat{\mathbf{w}}_k)\}_{k=1}^{\infty}$ is used to generate the sequence of signal-state estimates $\{\hat{\mathbf{x}}_k\}_{k=1}^{\infty}$. Meanwhile, these signal estimates are used to generate a sequence of weight estimates $\{\hat{\mathbf{w}}_k\}_{k=1}^{\infty}$ from the sequence of costs $\{J^j(\hat{\mathbf{x}}_1^k, \mathbf{w}_k)\}_{k=1}^{\infty}$.

Hence, the estimates of the signal-state and weights are generated simultaneously, with the estimation of each quantity depending on the estimate of the other, as shown in Figure 3.4. As discussed in the following paragraphs, Kalman filters can be used for both the signal estimation and weight estimation components, resulting in the *dual extended Kalman filter* family of algorithms.

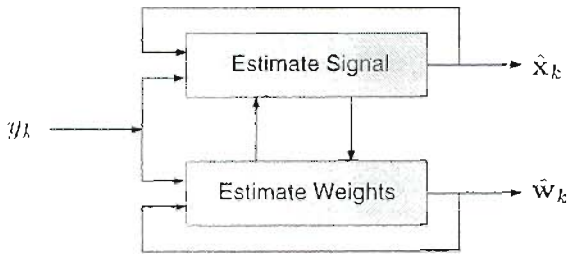


Figure 3.4: In sequential dual estimation, the signal and weights estimates are produced concurrently, with the estimation of each quantity depending on the other.

Signal Estimation

To estimate $\mathbf{x}_k$, the cost $J^I(\mathbf{x}_1^k, \mathbf{w})$ is evaluated using the weight estimates $\{\hat{\mathbf{w}}_t^-\}_1^k$. This is similar to the batch cost in Equation 2.12 on page 24, except that a *sequence* of weight estimates is used, rather than a single $\hat{\mathbf{w}}$:

$$J^I(\mathbf{x}_1^k, \mathbf{w}_k^-) = \sum_{t=1}^{k} \left(\frac{(y_t - x_t)^2}{\sigma_n^2} + \frac{(x_t - \hat{x}_t^-)^2}{\sigma_v^2} \right),$$

where the prediction is: $\hat{x}_t^- = f(\mathbf{x}_{t-1}, \hat{\mathbf{w}}_t^-)$. This is also identical to the signal-estimation cost $J(\mathbf{x}_1^k)$ given on page 44, except that the known, fixed weight vector $\mathbf{w}$ has been replaced here by the time-varying sequence of weight estimates, $\{\hat{\mathbf{w}}_t^-\}_1^k$. Section 3.2 showed that a Kalman filter produces sequential estimates $\hat{\mathbf{x}}_k$ that minimize $J(\mathbf{x}_1^k)$. Hence, given weight estimates $\{\hat{\mathbf{w}}_t^-\}_1^k$, a Kalman filter (or EKF) will produce the state-estimate $\hat{\mathbf{x}}_k$ that is optimal with respect to the above cost.

Weight Estimation

The weights $\mathbf{w}$ are estimated by minimizing the joint cost $J^I(\mathbf{x}_1^k, \mathbf{w})$, evaluated using the signal estimates $\{\hat{x}_t\}_1^k$. This is given by Equation 2.13 on page 24, restated here for data up to the current time k:

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

where $\hat{x}_t^- = f(\hat{\mathbf{x}}_{t-1}, \mathbf{w})$. There is no immediate restriction here on how the signal estimates are found; however, this cost will generally only be useful for weight estimation if $\{\hat{x}_t\}_1^k$ are chosen to be a function of $\mathbf{w}$. If the estimates $\hat{x}_t$ are not considered to be a function of $\mathbf{w}$, then the cost function reduces to the second term alone, and is essentially a prediction-error cost on the signal estimates. This simplified joint cost is expressed as $J_t^I(\hat{\mathbf{x}}_1^k, \mathbf{w})$ in Equation 2.14 on page 25, and is also identical to the weight-estimation cost $J(\mathbf{w})$ given in Equation 3.48, except that the clean

signal x_k has been replaced by estimates. The Kalman weight filter of Formulae 3.5 and 3.6 can be directly applied using $\{\hat{x}_l\}_1^k$. However, this procedure is somewhat risky, as there is no guarantee that $\hat{x}_k$ is at all related to the data.

On the other hand, if $\hat{x}_k$ and $\hat{x}_k^-$ are produced by a linear or extended Kalman filter, as described on the previous page, then both terms in the cost function are used. In this case, both $\hat{x}_k$ and $\hat{x}_k^-$ are *recursive* functions of the weights. To minimize the full cost function $J^j(\hat{\mathbf{x}}_1^k, \mathbf{w})$, a special *two-observation* form of the weight filter is used. An equivalent version of this filter appears in [60]; however, the observed-error form is shown here for consistency with dual EKF variations throughout this section.

The observed-error form of the Kalman weight filter – described on page 62 – can be used by defining the instantaneous cost as:

$$J_k = \frac{(y_k - \hat{x}_k)^2}{\sigma_n^2} + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2} \quad , \text{or} \quad J_k = \frac{e_k^2}{\sigma_n^2} + \frac{\tilde{\hat{x}}_k^2}{\sigma_v^2}, \tag{3.125}$$

where $e_k \triangleq (y_k - \hat{x}_k)$ and $\tilde{\hat{x}}_k \triangleq (\hat{x}_k - \hat{x}_k^-)$. Hence, $\sum_1^N J_k = J^j(\hat{\mathbf{x}}_1^N, \mathbf{w})$. The gradient and Hessian are shown in Appendix E, and can be approximated as described in Section 3.3 by defining a vector form of the observed-error. This is shown along with its negative derivative in Formula 3.9. This gives $\boldsymbol{\epsilon}_k^T \boldsymbol{\epsilon}_k = J_k$, as required. Letting $\sigma_r^2 = \frac{1}{2} \cdot \mathbf{I}$, the negative gradient is produced by

$$\boldsymbol{\epsilon}_k \triangleq \begin{bmatrix} \sigma_n^{-1} e_k \\ \sigma_v^{-1} \tilde{\hat{x}}_k \end{bmatrix}, \quad \text{with negative Jacobian} \quad \mathbf{H}_{o,k} = -\begin{bmatrix} \sigma_n^{-1} \nabla_{\mathbf{w}}^T e_k \\ \sigma_v^{-1} \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k \end{bmatrix}$$

Formula 3.9: Joint cost function observed-error terms for dual EKF weight filter.

$\mathbf{H}_{o,k}^T \sigma_r^{-2} \boldsymbol{\epsilon}_k = -\nabla_{\mathbf{w}} J_k$, as shown in Appendix E, and a first-order approximation to the instantaneous Hessian $\nabla_{\mathbf{w}}^2 J_k$ is given by: $\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k}$. Although alternative formulations of the observed-error (such as $\boldsymbol{\epsilon}_k = \sqrt{J_k}$) will produce the correct gradient, they will not produce a good approximation to the Hessian. The derivatives contained in $\mathbf{H}_{o,k}$ evaluate as:

$$\nabla_{\mathbf{w}} e_k = -\nabla_{\mathbf{w}} \hat{x}_k, \quad \text{and} \quad \nabla_{\mathbf{w}} \tilde{\hat{x}}_k = (\nabla_{\mathbf{w}} \hat{x}_k - \nabla_{\mathbf{w}} \hat{x}_k^-), \tag{3.126}$$

and so must be computed recursively; the derivatives of $\hat{x}_k$ and $\hat{x}_k^-$ are computed through the recurrent Kalman filter structure. Because these computations are the same for any of the dual Kalman filter variations, this procedure is described in Section 3.6.1 on page 102.

Combining the signal estimation filter and weight estimation filter produces the dual Kalman filter, presented in Formula 3.10. The algorithm is shown schematically in Figure 3.5. As described in the rest of this chapter, the algorithm can be applied to other cost functions by redefining J_k, $\boldsymbol{\epsilon}_k$, and $\mathbf{H}_{o,k}$ as needed.

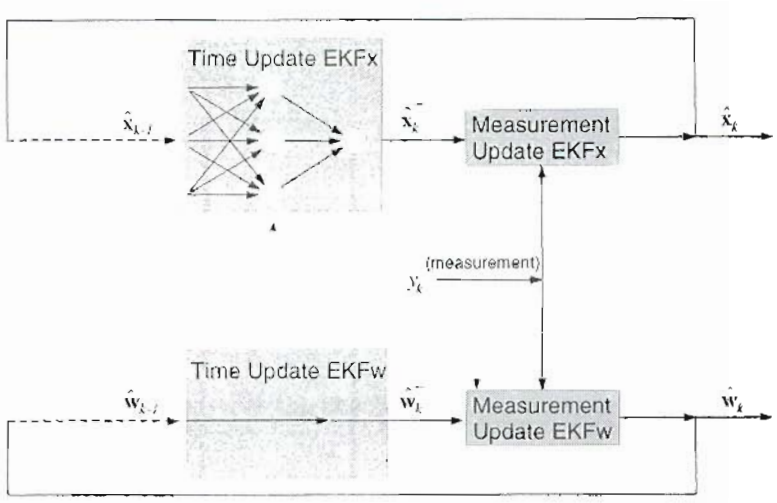


Figure 3.5: The dual extended Kalman filter. The algorithm consists of two EKFs run concurrently. The top EKF generates signal estimates, and requires $\hat{\mathbf{w}}_{k-1}$ for the time-update. The bottom EKF generates weight estimates, and requires $\hat{\mathbf{x}}_{k-1}$ for the measurement update.

Variance Estimation

When the variance terms σ_n^2 and σ_ϵ^2 are not known, they can be estimated by minimizing the cost function given in Equation 2.15 on page 25, repeated below:

$$J^j(\sigma^2) = \sum_{t=1}^{k} \left(\log(2\pi\sigma_n^2) + \frac{(y_t - \hat{y}_t)^2}{\sigma_n^2} + \log(2\pi\sigma_\epsilon^2) + \frac{(\hat{x}_t - \hat{x}_t^-)^2}{\sigma_\epsilon^2} \right).$$

If the dependence of the signal estimates $\hat{x}_t$ and predictions $\hat{x}_t^-$ on the noise variances is ignored, then either of the variances ($\sigma^2 = \sigma_n^2$ or $\sigma^2 = \sigma_\epsilon^2$) can be found by minimizing only the terms in which it appears. In either case, $\hat{\sigma}^2$ is the average of the quadratic term in the appropriate numerator. This ad hoc approach to estimating the noise variances from the average of squared error terms has been reported elsewhere [62, 81], but is not regarded in the literature as a reliable method for variance estimation.

In reality, *both* the signal estimates and predictions will be functions of the noise variances, so the cost function cannot be minimized so easily. 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:

$$\sigma_{k+1}^2 = \sigma_k^2 + u_k, \tag{3.137}$$

$$0 = \epsilon_k + r_k \tag{3.138}$$

Initialize with:

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

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

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

$$\hat{\mathbf{w}}_k^- = \hat{\mathbf{w}}_{k-1} \tag{3.127}$$

$$\mathbf{Q}_k^- = \mathbf{Q}_{k-1} + \mathbf{U}_k \qquad = \lambda^{-1}\mathbf{Q}_{k-1} \tag{3.128}$$

and for the signal filter are:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{w}}_k^-) \tag{3.129}$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{B}\sigma_v^2\mathbf{B}^T \tag{3.130}$$

The measurement update equations for the signal filter are:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1} \tag{3.131}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \tag{3.132}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^- \tag{3.133}$$

and for the weight filter are:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^-\mathbf{H}_{o,k}^T(\mathbf{H}_{o,k}\mathbf{Q}_k^-\mathbf{H}_{o,k}^T + \sigma_r^2)^{-1} \tag{3.134}$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} \cdot \boldsymbol{\epsilon}_k \tag{3.135}$$

$$\mathbf{Q}_k = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}}\mathbf{H}_{o,k})\mathbf{Q}_k^-. \tag{3.136}$$

Formula 3.10: The dual extended Kalman filter equations. The definitions of $\boldsymbol{\epsilon}_k$ and $\mathbf{H}_{o,k}$ will depend on the particular form of the weight filter being used. See the text for details.

which gives a one-dimensional state-space representation. Introducing the notation $\ell_n = \log(2\pi\sigma_n^2)$ and $\ell_v = \log(2\pi\sigma_v^2)$, the condition $\check{\boldsymbol{\epsilon}}_k^T\check{\boldsymbol{\epsilon}}_k = J_k$ is satisfied by defining the observed-error as in Formula 3.11[5]. The derivatives $\frac{\partial\sigma_n^2}{\partial\sigma^2}$ and $\frac{\partial\sigma_v^2}{\partial\sigma^2}$ evaluate to either 0 or 1, depending on whether $\sigma^2 = \sigma_n^2$, or $\sigma^2 = \sigma_v^2$. The other derivatives:

$$\frac{\partial e_k}{\partial\sigma^2} = -\frac{\partial\hat{x}_k}{\partial\sigma^2}, \quad \text{and} \quad \frac{\partial\tilde{\hat{x}}_k}{\partial\sigma^2} = (\frac{\partial\hat{\boldsymbol{x}}_k}{\partial\sigma^2} - \frac{\partial\hat{\boldsymbol{x}}_k^-}{\partial\sigma^2}), \tag{3.139}$$

[5]Note that some elements of $\check{\boldsymbol{\epsilon}}_k$ will generally take on complex values because the log terms that appear in the square root can be less than zero. However, the gradient and approximate Hessian will be real.

$$\breve{\mathbf{e}}_k \triangleq \begin{bmatrix} (\ell_n)^{\frac{1}{2}} \\ \sigma_n^{-1} e_k \\ (\ell_v)^{\frac{1}{2}} \\ \sigma_v^{-1} \tilde{\tilde{x}}_k \end{bmatrix}, \quad \text{with negative derivative} \quad \breve{\mathbf{H}}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_n)^{-\frac{1}{2}}}{\sigma_n^2}\frac{\partial \sigma_n^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_n}\frac{\partial e_k}{\partial \sigma^2} + \frac{e_k}{2(\sigma_n^2)^{(3/2)}}\frac{\partial \sigma_n^2}{\partial \sigma^2} \\ -\frac{1}{2}\frac{(\ell_v)^{-\frac{1}{2}}}{\sigma_v^2}\frac{\partial \sigma_v^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_v}\frac{\partial \tilde{\tilde{x}}_k}{\partial \sigma^2} + \frac{\tilde{\tilde{x}}_k}{2(\sigma_v^2)^{(3/2)}}\frac{\partial \sigma_v^2}{\partial \sigma^2} \end{bmatrix}$$

Formula 3.11: Joint cost function observed-error terms for dual EKF variance filter.

must be computed recursively, as described in Section 3.6.1. If these recursive derivatives are ignored (set to zero), the algorithm minimizes the ad hoc cost described above, instead of the full cost of Equation 2.15.

As shown in Appendix E, $\breve{\mathbf{H}}_{o,k}\sigma_r^{-2}\breve{\mathbf{e}}_k = -\frac{\partial J_k}{\partial \sigma^2}$ produces the exact negative of the derivative. The second derivative is approximated by $\breve{\mathbf{H}}_{o,k}^T\sigma_r^{-2}\breve{\mathbf{H}}_{o,k}$; this gives nearly the exact first-order part of the Hessian as long as $\ell_n = \sigma_n^2/(3e_k^2 - 2\sigma_n^2)$ and $\ell_v = \sigma_v^2/(3\tilde{\tilde{x}}_k^2 - 2\sigma_v^2)$. These values can be substituted directly in the expression for $\mathbf{e}_k$ and $\mathbf{H}_{o,k}$ in Formula 3.11; while this seems to contradict the earlier definitions of ℓ_v and ℓ_n, the situation is not so bleak. Consider for a moment adding an offset $\log(\alpha_k) + \log(\gamma_k)$ to the cost J_k; this will have no effect on the optimization process. Such a constant might be added by changing the base of the log functions, or equivalently, by making the following redefinitions:

$$\ell_n \triangleq \log(\alpha_k \cdot 2\pi\sigma_n^2) \qquad \ell_v \triangleq \log(\gamma_k \cdot 2\pi\sigma_v^2), \tag{3.140}$$

In principle, α_k and γ_k can be chosen arbitrarily at each time k, so values can be selected such that the required conditions are met. Fortunately, actual values for α_k and γ_k need not be computed. Instead, the required values can be directly substituted for ℓ_n and ℓ_v in the expressions for $\mathbf{e}_k$ and $\mathbf{H}_{o,k}$.

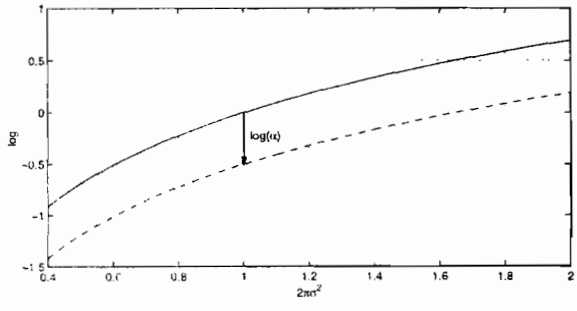


Figure 3.6: Effect of scaling parameter α on the log function. The solid curve shows $\log(2\pi\sigma^2)$, while the dotted curve shows $\log(\alpha \cdot 2\pi\sigma^2)$, when $\alpha \approx 0.6$. When $2\pi\sigma^2 = 1$, the log is effectively moved from a value of 0 to $-\frac{1}{2}$, with no change in the slope.

Initialize with:

$$\hat{\sigma}_0^2 = E[\sigma^2], \qquad q_0 = E[(\sigma^2 - \hat{\sigma}_0^2)(\sigma^2 - \hat{\sigma}_0^2)^T]$$

The variance estimation filter is shown in Formula 3.12. For $k \in \{1, \ldots, \infty\}$, the time update equations for the variance filter are:

$$\hat{\sigma^2}_k^- = \hat{\sigma}_{k-1}^2 \tag{3.141}$$

$$q_k^- = q_{k-1} + \sigma_u^2, \qquad \sigma_u^2 = \left(\frac{1}{\lambda} - 1\right) q_{k-1} \tag{3.142}$$

and the measurement equations are:

$$q_k = \left((q_k^-)^{-1} + \breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{H}}_{o,k} \right)^{-1} \tag{3.143}$$

$$\hat{\sigma}_k^2 = \hat{\sigma^2}_k^- + q_k \cdot \breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{e}}_k \tag{3.144}$$

Formula 3.12: The variance update equations. The definitions of $\breve{\mathbf{e}}_k$ and $\breve{\mathbf{H}}_{o,k}$ will depend on the particular form of the weight filter being used. See the text for details.

Note that the dimension of the state-space is 1 in the case of variance estimation, while the observation $\breve{\mathbf{e}}_k$ 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.

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, or into the filter in Formula 3.12, it is conceivable that negative values can be obtained. One solution to this problem (inspired by [74]) is to estimate $\ell \triangleq \log(\sigma^2)$ instead. Negative values of ℓ map to small positive values of σ^2, and $\ell = -\infty$ maps to $\sigma^2 = 0$. The log 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 log 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, rather than σ^2. Fortunately, the change is fairly straightforward. If the cost is a differentiable function of σ^2, then it is equivalently a differentiable function of e^ℓ. The first

derivative of the cost with respect to ℓ is:

$$\frac{\partial J}{\partial \ell} = \frac{\partial J}{\partial \sigma^2} \cdot \frac{\partial \sigma^2}{\partial \ell} \tag{3.145}$$

$$= \frac{\partial J}{\partial \sigma^2} \cdot e^{\ell} \tag{3.146}$$

$$= \frac{\partial J}{\partial \sigma^2} \cdot \sigma^2, \tag{3.147}$$

and the second derivative is:

$$\frac{\partial^2 J}{(\partial \ell)^2} = \frac{\partial(\partial J)}{\partial \sigma^2 \partial \ell} \cdot \sigma^2 + \frac{\partial J}{\partial \sigma^2} \cdot \frac{\partial \sigma^2}{\partial \ell} \tag{3.148}$$

$$= \frac{\partial^2 J}{(\partial \sigma^2)^2} \cdot \frac{\partial \sigma^2}{\partial \ell} \cdot \sigma^2 + \frac{\partial J}{\partial \sigma^2} \cdot \frac{\partial e^{\ell}}{\partial \ell} \tag{3.149}$$

$$= \frac{\partial^2 J}{(\partial \sigma^2)^2} \left(\sigma^2\right)^2 + \frac{\partial J}{\partial \sigma^2} \cdot \sigma^2. \tag{3.150}$$

These expressions are simple functions of the derivatives with respect to σ^2, which are approximated by $\breve{\mathbf{H}}_{o,k}^T \frac{1}{\sigma_r^2} \breve{\mathbf{e}}_k$ and $\breve{\mathbf{H}}_{o,k}^T \frac{1}{\sigma_r^2} \breve{\mathbf{H}}_{o,k}$. Hence, an alternative variance estimation filter is obtained by replacing the measurement update in Formula 3.12 with the alternative update in Formula 3.13. Strictly speaking, this no longer takes the form of a Kalman filter; it should instead be interpreted

$$q_k = \left((q_k^-)^{-1} + \breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{H}}_{o,k} \cdot (\hat{\sigma^2}_k^-)^2 + \breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{e}}_k \cdot \hat{\sigma^2}_k^- \right)^{-1} \tag{3.151}$$

$$\hat{\ell}_k^- = \log(\hat{\sigma^2}_k^-) \tag{3.152}$$

$$\hat{\ell}_k = \hat{\ell}_k^- + q_k \cdot \breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{e}}_k \cdot \hat{\sigma^2}_k^- \tag{3.153}$$

$$\hat{\sigma}_k^2 = e^{\hat{\ell}_k} \tag{3.154}$$

Formula 3.13: Alternative variance update using the log of the variance.

as a modified Newton learning rule. This form of the variance filter is used in the experiments in Chapter 4, with $\breve{\mathbf{e}}_k$ and $\breve{\mathbf{H}}_{o,k}$ defined according to the cost-function that is chosen.

Error Coupling

Although it is a reasonable approach, the direct substitution (in each filter) of estimated values for true ones fails to account for the errors in those estimates. As discussed in Section 2.3, these errors can be taken into consideration by making adjustments to the cost functions.

These changes amount to replacing the sequence of cost functions described on page 71 with the alternative sequences: $\{J^{ec}(\mathbf{x}_1^k)\}_1^\infty$ and $\{J^{ec}(\mathbf{w}_k)\}_1^\infty$. These sequences will also converge to

$J^j(\mathbf{x}_1^k, \mathbf{w})$ as the signal and weight estimates converge to their true values. However, as described in Section 2.3, the alternative costs have the potential to promote faster convergence in some cases.

Signal Estimation

The error in the weights is accounted for by modeling the resultant error in the dynamics, $\tilde{f}_k$ as a white Gaussian noise process. The batch form of the cost function was given in Equation 2.17 on page 26 as:

$$J^{ec}(\mathbf{x}_1^k) = \sum_{k=1}^{N} \left(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - \hat{x}_k^-)^2}{\sigma_{\tilde{f},k}^2 + \sigma_v^2} + \log(2\pi(\sigma_{\tilde{f},k}^2 + \sigma_v^2)) \right).$$

The variance $\sigma_{\tilde{f},k}^2$ of the dynamics error can be computed by approximating the dynamics to first order as $f(\mathbf{x}_{k-1}, \mathbf{w}) \approx \mathbf{H}_k \mathbf{w}$, where $\mathbf{H}_k \triangleq \nabla_{\mathbf{w}}^T f(\mathbf{x}_{k-1}, \mathbf{w})$, so that:

$$\begin{aligned}
\sigma_{\tilde{f},k}^2 &= E[(f(\mathbf{x}_{k-1}, \mathbf{w}) - f(\mathbf{x}_{k-1}, \hat{\mathbf{w}}_k^-))^2 | \{x_t\}_1^{k-1}, \hat{\mathbf{w}}_k^-] && \text{(3.155a)} \\
&\approx E[\mathbf{H}_k(\mathbf{w} - \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T \mathbf{H}_k^T | \{x_t\}_1^{k-1}, \hat{\mathbf{w}}_k^-] && \text{(3.155b)} \\
&= \mathbf{H}_k E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T] \mathbf{H}_k^T && \text{(3.155c)} \\
&= \mathbf{H}_k \mathbf{Q}_k^- \mathbf{H}_k^T. && \text{(3.155)}
\end{aligned}$$

Note that $\sigma_{\tilde{f},k}^2$ is independent of the current state, $\mathbf{x}_k$. Hence, in sequential estimation (wherein only the current state is estimated) the log term can be ignored. The error in the weight estimates can then be accounted for by simply replacing the process noise variance, σ_v^2, with $(\sigma_v^2 + \sigma_{\tilde{f},k}^2)$ in the signal filter portion of Formula 3.10.

In the sequential estimation case, the derivative, $\mathbf{H}_k$, of $f(\cdot)$ with respect to $\mathbf{w}$, is evaluated at the previous estimate, $\hat{\mathbf{x}}_{k-1}$. Because $\hat{\mathbf{x}}_{k-1}$ is itself a recursive function of $\mathbf{w}$, this suggests that $\mathbf{H}_k$ should be computed as a recurrent derivative, as shown in Section 3.6.1.

Weight Estimation

For weight estimation, taking the error in the signal estimates into account is somewhat more complicated. Here, the relevant cost function (Equation 2.19 on page 27) is:

$$\begin{aligned}
J^{ec}(\mathbf{w}) = \sum_{k=1}^{N} \Bigg(& \log(2\pi\sigma_{e_k}^2) + \frac{(y_k - \hat{x}_k)^2}{\sigma_{e_k}^2} \\
& + \log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} \Bigg),
\end{aligned}$$

where the variance terms are calculated as:

$$\sigma_{e_k}^2 = E[(y_k - \hat{x}_k)^2] \tag{3.156a}$$

$$= E[(n_k + x_k - \hat{x}_k)^2] \tag{3.156b}$$

$$= \sigma_n^2 + \mathbf{C}\mathbf{P}_k\mathbf{C}^T, \tag{3.156}$$

and

$$g_k = E[(\hat{x}_k - \hat{x}_k^-)^2] \tag{3.157a}$$

$$= E[\mathbf{C}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T\mathbf{C}^T] \tag{3.157b}$$

$$= E[\mathbf{C}\mathbf{K}_k(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-)^2\mathbf{K}_k^T\mathbf{C}^T] \tag{3.157c}$$

$$= E[\mathbf{C}\mathbf{K}_k(n_k + x_k - \mathbf{C}\hat{\mathbf{x}}_k^-)^2\mathbf{K}_k^T\mathbf{C}^T] \tag{3.157d}$$

$$= \mathbf{C}\mathbf{K}_k(\sigma_n^2 + \mathbf{C}\mathbf{P}_k^-\mathbf{C}^T)\mathbf{K}_k^T\mathbf{C}^T, \tag{3.157}$$

respectively[6]. Seeing that both $\mathbf{P}_k$ and $\mathbf{P}_k^-$ are functions of $\mathbf{w}$, so are $\sigma_{e_k}^2$ and g_k. The gradients of the variances are:

$$\nabla_{\mathbf{w}}\sigma_{e_k}^2 = \nabla_{\mathbf{w}}\mathbf{P}_k^{(1,1)}, \tag{3.158}$$

and

$$\nabla_{\mathbf{w}}g_k = 2\mathbf{K}_k^{(1)}\nabla_{\mathbf{w}}\mathbf{K}_k^{(1)}(\sigma_n^2 + \mathbf{C}\mathbf{P}_k^-\mathbf{C}^T) + (\mathbf{K}_k^{(1)})^2\nabla_{\mathbf{w}}(\mathbf{P}_k^-)^{(1,1)}, \tag{3.159}$$

where the gradients of the elements of $\mathbf{K}_k$ and $\mathbf{P}_k$ must be computed recursively, using the equations of the Kalman signal filter (see Section 3.6.1).

Sequential minimization of $J^{ec}(\mathbf{w})$ is provided by an observed-error weight filter. Here the instantaneous error is:

$$J_k = \log(2\pi\sigma_{e_k}^2) + \frac{e_k^2}{\sigma_{e_k}^2} + \log(2\pi g_k) + \frac{(\tilde{\hat{x}}_k)^2}{g_k}, \tag{3.160}$$

with the gradient and Hessian shown in Appendix E. The gradient and Hessian of J_k are approximated by defining the observed-error term and its negative derivative as in Formula 3.14. where $\ell_{e,k} \triangleq \log(2\pi\sigma_{e_k}^2)$ and $\ell_{g,k} \triangleq \log(2\pi g_k)$. This satisfies $\mathbf{e}_k^T\mathbf{e}_k = J_k$, and $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{e}_k = -\nabla_{\mathbf{w}}J_k$ gives the negative gradient, as shown in Appendix E. The Hessian is approximated by $\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{H}_{o,k}$; this gives nearly the exact first-order part of the Hessian when $\ell_{e,k} = \sigma_{e_k}^2/(3e_k^2 - 2\sigma_{e_k}^2)$ and $\ell_{g,k} = g_k/(3\tilde{\hat{x}}_k^2 - 2g_k)$.

[6]Note that g_k should always be at least as large as σ_r^2.

$$\mathbf{e}_k = \begin{bmatrix} (\ell_{e,k})^{\frac{1}{2}} \\ \sigma_{e_k}^{-1} e_k \\ (\ell_{g,k})^{\frac{1}{2}} \\ g_k^{(-1/2)} \tilde{\hat{x}}_k \end{bmatrix}, \quad \text{and} \quad \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_{e,k})^{-\frac{1}{2}}}{\sigma_{e_k}^2} \nabla_{\mathbf{w}}^T(\sigma_{e_k}^2) \\ -\frac{1}{\sigma_{e_k}} \nabla_{\mathbf{w}}^T e_k + \frac{e_k}{2(\sigma_{e_k}^2)^{(3/2)}} \nabla_{\mathbf{w}}^T(\sigma_{e_k}^2) \\ -\frac{1}{2}\frac{(\ell_{g,k})^{-\frac{1}{2}}}{g_k} \nabla_{\mathbf{w}}^T(g_k) \\ -\frac{1}{g_k^{(1/2)}} \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k + \frac{\tilde{\hat{x}}_k}{2g_k^{(3/2)}} \nabla_{\mathbf{w}}^T(g_k) \end{bmatrix}$$

Formula 3.14: Error-coupled cost function observed-error terms for dual EKF weight filter.

Variance Estimation

The variance terms σ_n^2 and σ_v^2 can also be estimated with information about the errors in both $\hat{\mathbf{x}}_1^N$ and $\hat{\mathbf{w}}$. This is accomplished by minimizing the cost function given in Equation 2.22:

$$J^{ec}(\sigma^2) = \sum_{k=1}^{N} \Bigg(\log(2\pi\sigma_{e_k}^2) + \frac{(y_k - \hat{x}_k)^2}{\sigma_{e_k}^2} + \log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} \Bigg),$$

which is identical to the cost given for weight estimation, except that the predictions here are given by $\hat{x}_k^- = f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \hat{\mathbf{w}}\big)$. The error-variance terms are $\sigma_{e_k}^2 = \sigma_n^2 + \mathbf{C}\mathbf{P}_k\mathbf{C}^T$ and $g_k = \mathbf{C}\mathbf{K}_k(\sigma_n^2 + \mathbf{C}\mathbf{P}_k^-\mathbf{C}^T)\mathbf{K}_k^T\mathbf{C}^T$, as shown in Equation 3.156 and 3.157 on the preceding page. The redefinition of $\hat{x}_k^-$ is reflected in $\mathbf{P}_k^-$, since this is produced by the error-coupled signal filter (which makes use of the statistics of $\hat{\mathbf{w}}$).

The instantaneous cost, J_k, is the quantity inside the above summation. The variance terms are estimated by defining the observed-error terms as in Formula 3.15. From Equations 3.156 and 3.157, the derivatives are

$$\breve{\mathbf{e}}_k = \begin{bmatrix} (\ell_{e,k})^{\frac{1}{2}} \\ \sigma_{e_k}^{-1} e_k \\ (\ell_{g,k})^{\frac{1}{2}} \\ g_k^{(-1/2)} \tilde{\hat{x}}_k \end{bmatrix}, \quad \text{and} \quad \breve{\mathbf{H}}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_{e,k})^{-\frac{1}{2}}}{\sigma_{e_k}^2} \frac{\partial \sigma_{e_k}^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_{e_k}} \frac{\partial e_k}{\partial \sigma^2} + \frac{e_k}{2(\sigma_{e_k}^2)^{(3/2)}} \frac{\partial \sigma_{e_k}^2}{\partial \sigma^2} \\ -\frac{1}{2}\frac{(\ell_{g,k})^{-\frac{1}{2}}}{g_k} \frac{\partial g_k}{\partial \sigma^2} \\ -\frac{1}{g_k^{(1/2)}} \frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} + \frac{\tilde{\hat{x}}_k}{2g_k^{(3/2)}} \frac{\partial g_k}{\partial \sigma^2} \end{bmatrix}$$

Formula 3.15: Error-coupled cost function observed-error terms for dual EKF variance filter.

$$\frac{\partial \sigma_{e_k}^2}{\partial \sigma^2} = \frac{\partial \sigma_n^2}{\partial \sigma^2} + \frac{\partial \mathbf{P}_k^{(1,1)}}{\partial \sigma^2}, \tag{3.161}$$

$$\text{and} \quad \frac{\partial g_k}{\partial \sigma^2} = 2\mathbf{K}_k^{(1)} \frac{\partial \mathbf{K}_k^{(1)}}{\partial \sigma^2} (\sigma_n^2 + \mathbf{C}\mathbf{P}_k^-\mathbf{C}^T) + (\mathbf{K}_k^{(1)})^2 \Big(\frac{\partial \sigma_n^2}{\partial \sigma^2} + \frac{\partial (\mathbf{P}_k^-)^{(1,1)}}{\partial \sigma^2} \Big), \tag{3.162}$$

where both $\frac{\partial \mathbf{P}_k^{(1,1)}}{\partial \sigma^2}$ and $\frac{\partial \mathbf{K}_k^{(1)}}{\partial \sigma^2}$ are computed recursively, as shown in Section 3.6.1.

With these quantities in hand, the variances can be estimated as described before, with a Kalman or extended Kalman filter. The variance estimation equations are given in Formulae 3.12 on page 77 and 3.13 on page 78.

Marginal Estimation

Section 2.4 describes a different approach to dual estimation, wherein the joint density function $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$ is expanded into two factors: $\rho_{\mathbf{x}_1^N | \mathbf{y}_1^N \mathbf{w}}$ and $\rho_{\mathbf{w} | \mathbf{y}_1^N}$. Marginal estimation methods maximize the first factor with respect to the signal, and the second factor with respect to the weights.

As in the decoupled joint estimation methods, the current weight estimates are used during signal estimation, and *vice versa*. Again, this can either be done in an iterative or sequential framework. Examples of iterative approaches include Lim and Oppenheim's well-known approach to speech enhancement [44], which alternates between noncausal Wiener filtering of the signal, and a least-squares solution of the weights; and the batch EM algorithm [76], which is described in more detail beginning on page 85.

Sequential Signal Estimation

For sequential estimation, maximizing $\rho_{\mathbf{x}_1^N | \mathbf{y}_1^N \mathbf{w}}$ with respect to the signal is done using a Kalman signal filter by substituting the current weight estimates $\hat{\mathbf{w}}_k^-$, for $\mathbf{w}$. See Formula 3.1 on page 49, and Formula 3.2 for the EKF. As with the decoupled joint estimation approaches, the hope is that as the weight estimates converge to $\mathbf{w}$, the signal estimates will tend toward their true values.

Sequential Weight Estimation

Maximizing $\rho_{\mathbf{w} | \mathbf{y}_1^N}$ with respect to the weights can be done by minimizing the negative log:

$$-\log \rho_{\mathbf{y}_1^N | \mathbf{w}} = \frac{1}{2} \sum_{k=1}^{N} \left(\log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \overline{y_{k|k-1}})^2}{\sigma_{\varepsilon_k}^2} \right). \tag{3.163}$$

The conditional mean is the prediction $\overline{y_{k|k-1}}$, with error-variance $\sigma_{\varepsilon_k}^2$. The mean is calculated as:

$$\overline{y_{k|k-1}} = E[y_k | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{3.164a}$$

$$= E[x_k + n_k | \{y_t\}_1^{k-1}, \mathbf{w}] \qquad = \hat{x}_k^- + 0 \tag{3.164b}$$

$$= \mathbf{C} \cdot \hat{\mathbf{x}}_k^- . \tag{3.164}$$

and variance is given by:

$$\sigma^2_{\varepsilon_k} = E[(y_k - \overline{y_{k|k-1}})^2 | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{3.165a}$$

$$= E[(n_k + x_k - \hat{x}_k^-)^2 | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{3.165b}$$

$$= \sigma_n^2 + \mathbf{C}\mathbf{P}_k^- \mathbf{C}^T, \tag{3.165}$$

so that both are computed by the Kalman signal filter, and thus are recursive functions of the weights $\mathbf{w}$. Their gradients are

$$\nabla_{\mathbf{w}}(\overline{y_{k|k-1}}) = \mathbf{C} \cdot \nabla_{\mathbf{w}} \hat{\mathbf{x}}_k^- \tag{3.166}$$

$$\nabla_{\mathbf{w}}(\sigma^2_{\varepsilon_k}) = \nabla_{\mathbf{w}}(\mathbf{P}_k^{-(1,1)}) \tag{3.167}$$

which must be computed recursively, as shown in Section 3.6.1.

Sequential Variance Estimation

If the noise variances represent addition unknown parameters, they can also be estimated by minimizing the cost in Equation 3.163. The mean and variance are computed the same as in Equations 3.164 and 3.165, except that the unknown variance σ^2 is now an additional conditioning argument in the expectations. Hence, the derivatives are:

$$\frac{\partial \overline{y_{k|k-1}}}{\partial \sigma^2} = -\frac{\partial \hat{x}_k^-}{\partial \sigma^2} \tag{3.168}$$

$$\frac{\partial \sigma^2_{\varepsilon_k}}{\partial \sigma^2} = \frac{\partial \sigma_n^2}{\partial \sigma^2} + \frac{\partial \mathbf{P}_k^{-(1,1)}}{\partial \sigma^2}. \tag{3.169}$$

Prediction Error Weight Estimation

If $\sigma^2_{\varepsilon_k}$ is assumed to be independent of $\mathbf{w}$, then the log term can be dropped from the cost function, leaving the squared prediction error cost:

$$J^{pe}(\mathbf{w}) = \sum_{k=1}^{N} \varepsilon_k^2, \tag{3.170}$$

where $\varepsilon_k \triangleq (y_k - \hat{x}_k^-)$. This cost corresponds to the simplest form of the dual EKF, developed in [87], which is equivalent to the *recursive prediction error* (RPE) method in [47, 52]. In [47, 52, 87], the weight filter is designed using the standard observation equation:

$$y_k = \overbrace{f(\mathbf{x}_{k-1}, \mathbf{w}_k) + v_k}^{x_k} + n_k, \tag{3.171}$$

which creates a filter of the same form as for the known-signal case in Formulae 3.5 and 3.6 in Section 3.3, except the clean target x_k is replaced by y_k, and the noise variance σ_v^2 is replaced by $(\sigma_v^2 + \sigma_n^2)$.

Equivalently, the observed-error form of weight filter is found by defining the instantaneous cost as $J_k = (y_k - \hat{x}_k^-)^2 = \varepsilon_k^2$, and letting the observed error be as in Formula 3.16, so that

$$\mathbf{e}_k \triangleq \varepsilon_k, \quad \text{and} \quad \mathbf{H}_{o,k} = -\nabla_{\mathbf{w}}\varepsilon_k \quad = \nabla_{\mathbf{w}}(\overline{y_{k|k-1}})$$

Formula 3.16: Prediction-error cost function observed-error terms for dual EKF weight filter.

the negative gradient is given by $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{e}_k = -2(\nabla_{\mathbf{w}}\varepsilon_k)\varepsilon_k$, and the Hessian is approximated to first-order by $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{H}_{o,k}^T = 2(\nabla_{\mathbf{w}}\varepsilon_k)(\nabla_{\mathbf{w}}\varepsilon_k)^T$. Note the similarity between these definitions of the observed-error weight filter and those presented for the known-signal case in Section 3.3.

Prediction Error Variance Estimation

The noise variances, σ_v^2 and σ_n^2, can also be estimated by minimizing the prediction error cost of Equation 3.170. The observed error $\breve{\mathbf{e}}_k$ is simply ε_k, as before, and $\breve{\mathbf{H}}_{o,k}$ is given by $-\frac{\partial \varepsilon_k}{\partial \sigma^2} = \frac{\partial \hat{x}_k^-}{\partial \sigma^2}$. Hence, prediction error variance estimation is entirely dependent on computing the derivatives of $\hat{x}_k^-$ recurrently.

Maximum Likelihood Weight Estimation

Taking the dependence of $\sigma_{\varepsilon_k}^2$ on $\mathbf{w}$ into account requires the minimization of everything in Equation 3.163. This means minimizing the full maximum-likelihood cost function given in Equation 2.52 on page 36, restated here as:

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

Here, the instantaneous cost J_k is the quantity in the summation. Defining $\ell_{\varepsilon,k} \triangleq \log(2\pi\alpha_k \cdot \sigma_{\varepsilon_k}^2)$, the appropriate weight filter is found by letting the observed-error and its negative derivative be as in Formula 3.17. and letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$. These terms are used in the dual EKF equations in

$$\mathbf{e}_k \triangleq \begin{bmatrix} \sqrt{\ell_{\varepsilon,k}} \\ \sigma_{\varepsilon_k}^{-1}\varepsilon_k \end{bmatrix}, \quad \text{and} \quad \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_{\varepsilon,k})^{-\frac{1}{2}}}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}^T(\sigma_{\varepsilon_k}^2) \\ -\frac{1}{\sigma_{\varepsilon_k}}\nabla_{\mathbf{w}}^T\varepsilon_k + \frac{\varepsilon_k}{2(\sigma_{\varepsilon_k}^2)^{(3/2)}}\nabla_{\mathbf{w}}^T(\sigma_{\varepsilon_k}^2) \end{bmatrix}$$

Formula 3.17: Maximum-likelihood cost function observed-error terms for dual EKF weight filter.

Formula 3.10. As shown in Appendix E, α_k should be chosen such that $\ell_{\varepsilon,k} = \sigma_{\varepsilon,k}^2/(3\varepsilon_k^2 - 2\sigma_{\varepsilon,k}^2)$ (as described on page 76). The negative gradient and Hessian are then approximated by $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{e}_k$ and $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{H}_{o,k}^T$.

Maximum Likelihood Variance Estimation

The maximum-likelihood cost function also offers a mechanism for estimating the variance terms, σ_n^2 and σ_v^2, when they are not known *a priori*. The instantaneous cost and observed-error terms are identical to those just given for weight estimation. The derivative of $-\breve{\boldsymbol{\epsilon}}_k$ is given in Formula 3.18,

$$\breve{\boldsymbol{\epsilon}}_k \triangleq \begin{bmatrix} \sqrt{\ell_{\varepsilon,k}} \\ \sigma_{\varepsilon_k}^{-1}\varepsilon_k \end{bmatrix}, \quad \text{and} \quad \breve{\mathbf{H}}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_{\varepsilon,k})^{-\frac{1}{2}}}{\sigma_{\varepsilon_k}^2}\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_{\varepsilon_k}}\frac{\partial \varepsilon_k}{\partial \sigma^2} + \frac{\varepsilon_k}{2(\sigma_{\varepsilon_k}^2)^{(3/2)}}\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} \end{bmatrix}$$

Formula 3.18: Maximum-likelihood cost function observed-error terms for dual EKF variance filter.

where σ^2 represents either σ_n^2 or σ_v^2, and where $\frac{\partial \varepsilon_k}{\partial \sigma^2} = -\frac{\partial \overline{y_{k|k-1}}}{\partial \sigma^2}$. The desired variance is estimated according to the equations in Formulae 3.12 and 3.13 on page 78.

Expectation Maximization

Although the EM algorithm is a marginal estimation method (see page 36), its general character is different enough from the maximum-likelihood and prediction-error methods to warrant separate treatment.

The EM algorithm has received a fair amount of attention recently in the context of estimating nonlinear dynamic systems [3, 5, 23]. Typically, the algorithm is used in an iterative framework, wherein the entire signal $\{x_k\}_1^N$ is estimated using the current weight estimates during the E-step, and the weights are estimated during the M-step using the entire trajectory of signal estimates $\{\hat{x}_k\}_1^N$.

As stated in Equation 2.55 on page 37, the EM cost is:

$$J^{em} = E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}\left[\sum_{k=1}^{N}\Big(\log(2\pi\sigma_n^2) + \frac{(y_k - x_k)^2}{\sigma_n^2} + \log(2\pi\sigma_v^2) + \frac{(x_k - x_k^-)^2}{\sigma_v^2}\Big) \;\Big|\; \{y_k\}_1^N, \hat{\mathbf{w}}\right],$$

which is minimized with respect to $\mathbf{w}$ during the M-step. As before, $x_k^- = f(\mathbf{x}_{k-1}, \mathbf{w})$. An important distinction exists between this value of $\mathbf{w}$ being estimated, and the previous estimate of the weights, $\hat{\mathbf{w}}$, used in the conditional expectation. As shown in Appendix F, the expectation evaluates to:

$$J^{em} = N\log(4\pi^2\sigma_v^2\sigma_n^2) + \sum_{k=1}^{N}\Big(\frac{(y_k - \hat{x}_{k|N})^2 + p_{k|N}}{\sigma_n^2} + \frac{(x_{k|N} - \hat{x}_{k|N}^-)^2 + p_{k|N} - 2p_{k|N}^\dagger + p_{k|N}^-}{\sigma_v^2}\Big) \tag{3.172}$$

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.

These conditional expectations are computed during the E-step, typically with a *Kalman smoother* algorithm [43, 68]. A Kalman smoother combines the results of both a forward and a backward pass over the data to produce the smoothed estimates[7]. When the system is nonlinear, the classical approach is to use an *extended* Kalman smoother (*e.g.*, in [23]), or use Gibbs sampling [83].

As discussed in Appendix F, only $\hat{x}_{k|N}^-$, $p_{k|N}^-$, and $p_{k|N}^\dagger$ are functions of $\mathbf{w}$. Hence, the portion of the cost relevant to weight estimation is:

$$J^{em}(\mathbf{w}) = \sum_{k=1}^{N} \left(\frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 - 2p_{k|N}^\dagger + p_{k|N}^-}{\sigma_v^2} \right). \tag{3.173}$$

Likewise, the portion of the cost in Equation 2.55 that depends on σ_v^2 is:

$$J^{em}(\sigma_v^2) = N \log(2\pi\sigma_v^2) + \sum_{k=1}^{N} \left(\frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 + p_{k|N} - 2p_{k|N}^\dagger + p_{k|N}^-}{\sigma_v^2} \right), \tag{3.174}$$

while the portion relevant to estimating σ_n^2 is:

$$J^{em}(\sigma_n^2) = N \log(2\pi\sigma_n^2) + \sum_{k=1}^{N} \left(\frac{(y_k - \hat{x}_{k|N})^2 + p_{k|N}}{\sigma_n^2} \right). \tag{3.175}$$

Note that in these last two expressions, the numerators inside the sums are dependent on the previous variance estimates, $\hat{\sigma}_v^2$ and $\hat{\sigma}_n^2$, but not on the value (σ_v^2 or σ_n^2) being estimated. In Equation 3.173, $\hat{x}_{k|N}^-$, $p_{k|N}^\dagger$, and $p_{k|N}^-$ are recursive functions of $\hat{\mathbf{w}}$, but not of $\mathbf{w}$. Hence, it happens that no recurrent derivative computations are required for the EM algorithm.

For the M-step, closed-form solutions are possible with linear models (and radial basis functions, as in [23]) using either a least-squares or RLS procedure. Expressions are provided in Appendix F. Typically, nonlinear models will require a *generalized* M-step, in which the cost function is *decreased* (but not necessarily minimized) at each iteration. This generalized M-step is often done with a gradient-descent method such as backpropagation, which can either be used in batch or pattern mode.

Regardless of the M-step, however, the Kalman smoothing for the E-step must be done off-line, as it makes noncausal use of the data. The EM algorithm is necessarily an *iterative* approach (see page 10) to dual estimation.

[7] The cross-covariance $p_{k|N}^\dagger$ is not calculated by the standard Kalman smoother, but can be included in the algorithm at small additional cost [76].

Sequential EM Cost

However, a fully sequential EM algorithm can be found by computing the expectations in the E-step with a Kalman filter rather than a Kalman smoother (*e.g.*, in [93]). This is equivalent to replacing the off-line means and covariances in Equations 3.172-3.175 with their on-line equivalents: $\hat{x}_{k|k}$, $p_{k|k}$, $\hat{x}^-_{k|k}$, $p^-_{k|k}$, and $p^\dagger_{k|k}$. The first two quantities are the usual estimate $\hat{x}_k = \mathbf{C}\hat{\mathbf{x}}_k$ and variance $\mathbf{C}\mathbf{P}_k\mathbf{C}^T$ computed by the Kalman signal filter. The remaining terms require special consideration.

The noncausal prediction $\hat{x}^-_{k|k}$ is defined as $E[f(\mathbf{x}_{k-1}, \mathbf{w})|\{y_t\}_1^k, \hat{\mathbf{w}}]$, which is difficult to compute in general because of its dependence on future data. As in the EKF, this expectation can be approximated by taking the function of the expected value: $f(E[\mathbf{x}_{k-1}|\{y_t\}_1^k, \hat{\mathbf{w}}], \mathbf{w})$, or equivalently, $f(\hat{\mathbf{x}}_{k-1|k}, \mathbf{w})$. Unfortunately, $\hat{\mathbf{x}}_{k-1|k}$ is not computed by the standard Kalman signal filter described in Section 3.2. However, a slight modification allows the KF to compute this quantity, in addition to the desired variance terms, $p^-_{k|k}$, and $p^\dagger_{k|k}$.

Specifically, the state-vector is augmented by one additional lagged value of the signal :

$$\mathbf{x}_k^+ = \begin{bmatrix} \mathbf{x}_k \\ x_{k-M} \end{bmatrix} = \begin{bmatrix} x_k \\ \mathbf{x}_{k-1} \end{bmatrix}, \tag{3.176}$$

so that the estimate $\hat{\mathbf{x}}_k^+$ produced by a Kalman filter will contain $\hat{\mathbf{x}}_{k-1|k}$ in its last M elements. Furthermore, the covariance $\mathbf{P}_k^+$ of $\mathbf{x}_k^+$ produced by the KF allows for approximate calculation of the variances $p^-_{k|k}$ and $p^\dagger_{k|k}$. Following the derivations in Appendix F:

$$p^-_{k|k} = \mathbf{C}\mathbf{A}_{k-1|k}(\mathbf{P}_{k-1|k})\mathbf{A}^T_{k-1|k}\mathbf{C}^T \quad \text{and} \quad p^\dagger_{k|k} = \mathbf{C}(\mathbf{P}^\natural_k)\mathbf{A}^T_{k-1|k}\mathbf{C}^T, \tag{3.177}$$

where the covariance $\mathbf{P}_{k-1|k}$ is provided as the lower right block of the augmented covariance $\mathbf{P}_k^+$, and $\mathbf{P}^\natural_k$ 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 Kalman filter requires only a couple of modifications to estimate $\mathbf{x}_k^+$:

1. A final zero element is included in the vectors $\mathbf{B}$ and $\mathbf{C}$.
2. The matrix $\mathbf{A}_k \triangleq \begin{bmatrix} \nabla_{\mathbf{x}}^T f \\ \mathbf{I} \quad 0 \end{bmatrix}$ is modified by increasing the dimension of $\mathbf{I}$ and adding a final column of zeros.

Note, the function $f(\mathbf{x}_k, \mathbf{w}_k)$ ignores the additional lagged element x_{k-M}. The overall dimension of the state-space representation is increased from M to $1+M$.

EM via the Dual EKF

With a sequential E-step provided by the Kalman signal filter on the augmented state $\mathbf{x}_k^+$, a sequential (generalized) M-step is needed for estimating the weights.

The observed-error weight filter can be easily applied for this purpose by defining the instantaneous error as:

$$J_k^{em}(\mathbf{w}) = \frac{(\hat{x}_k - \hat{x}_{k|k}^-)^2 - 2p_{k|k}^\dagger + p_{k|k}^-}{\sigma_v^2}. \tag{3.178}$$

The appropriate observed-error vector and its negative Jacobian matrix are given in Formula 3.19 where $\tilde{\hat{x}}_{k|k} = (\hat{x}_k - \hat{x}_{k|k}^-)$. Letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$, the negative gradient and Hessian of J_k^{em} are approxi-

$$\boldsymbol{\epsilon}_k = \begin{bmatrix} \sigma_v^{-1}\tilde{\hat{x}}_{k|k} \\ \sqrt{-2}\sigma_v^{-1}(p_{k|k}^\dagger)^{\frac{1}{2}} \\ \sigma_v^{-1}(p_{k|k}^-)^{\frac{1}{2}} \end{bmatrix} \text{ and } \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{\sigma_v}\nabla_{\mathbf{w}}^T\tilde{\hat{x}}_{k|k} \\ -\frac{\sqrt{-2}(p_{k|k}^\dagger)^{-\frac{1}{2}}}{2\sigma_v}\nabla_{\mathbf{w}}^T p_{k|k}^\dagger \\ -\frac{(p_{k|k}^-)^{-\frac{1}{2}}}{2\sigma_v}\nabla_{\mathbf{w}}^T p_{k|k}^- \end{bmatrix}$$

Formula 3.19: EM cost function observed-error terms for dual EKF weight filter.

mated by $\mathbf{H}_{o,k}\sigma_r^{-2}\boldsymbol{\epsilon}_k$ and $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{H}_{o,k}^T$, respectively, as shown in Appendix E. Note that the error variances $p_{k|k}^-$ and $p_{k|k}^\dagger$ cancel out of the gradient expression; as shown in the Appendix, to obtain a good Hessian approximation, they are replaced by very large values in the expression for $\mathbf{H}_{o,k}$.

Because $\hat{x}_k$ is not a function of $\mathbf{w}$ (it depends on $\hat{\mathbf{w}}$), the gradient $\nabla_{\mathbf{w}}^T\tilde{\hat{x}}_{k|k}$ is simply the negative of $\nabla_{\mathbf{w}}^T\hat{x}_{k|k}^-$, where

$$\nabla_{\mathbf{w}}\hat{x}_{k|k}^- = \nabla_{\mathbf{w}} f(\hat{\mathbf{x}}_{k-1|k}, \mathbf{w}), \tag{3.179}$$

which is evaluated at $\hat{\mathbf{x}}_{k-1|k}$ and $\hat{\mathbf{w}}$. Following Equation 3.177, the i^{th} element of the gradient vector $\nabla_{\mathbf{w}}p_{k|k}^-$ is constructed from the expression:

$$\frac{\partial p_{k|k}^-}{\partial w^{(i)}} = \mathbf{C}\Big(\frac{\partial \mathbf{A}_{k-1|k}}{\partial w^{(i)}}(\mathbf{P}_{k-1|k})\mathbf{A}_{k-1|k}^T + \mathbf{A}_{k-1|k}(\mathbf{P}_{k-1|k})\frac{\partial \mathbf{A}_{k-1|k}^T}{\partial w^{(i)}}\Big)\mathbf{C}^T, \tag{3.180}$$

and each of the elements of the gradient $\nabla_{\mathbf{w}}p_{k|k}^\dagger$ is

$$\frac{\partial p_{k|k}^\dagger}{\partial w^{(i)}} = \mathbf{C}\Big(\frac{\partial \mathbf{A}_{k-1|k}}{\partial w^{(i)}}(\mathbf{P}_k^\sharp)^T\Big)\mathbf{C}^T. \tag{3.181}$$

EM Variance Estimation

The variances σ_v^2 and/or σ_n^2 can also be estimated as part of the sequential M-step. As seen in Equations 3.174- 3.175 on page 86, either of the variances can be estimated by minimizing an instantaneous cost of the form:

$$J_k^{em}(\sigma^2) = \log(2\pi\sigma^2) + \frac{num_k}{\sigma^2}, \tag{3.182}$$

where num_k is one of the *numerator* terms in Equations 3.174-3.175, and σ^2 represents either σ_n^2 or σ_v^2. Recall that the numerator term is independent of σ^2. As shown in Appendix E, the best approximation to the first and second derivatives of J_k with respect to σ^2 is obtained by defining the terms for the variance filter as in Formula 3.20, and letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$. This is a slight

$$\breve{\mathbf{e}}_k = \frac{1}{2}\begin{bmatrix} (\ell)^{\frac{1}{2}} \\ \sigma^{-1}\sqrt{num_k} \end{bmatrix} \quad \text{and} \quad \breve{\mathbf{H}}_{o,k} = -2\begin{bmatrix} \frac{1}{2}\frac{(\ell)^{-\frac{1}{2}}}{\sigma^2} \\ -\frac{1}{2}\frac{\sqrt{num_k}}{(\sigma_n^2)^{(3/2)}} \end{bmatrix}$$

Formula 3.20: EM cost function observed-error terms for dual EKF variance filter.

adjustment from the conventional definitions used elsewhere in this thesis, in that $\breve{\mathbf{H}}_{o,k}$ is not exactly the negative derivative of $\breve{\mathbf{e}}_k$ in this case. In particular, both quantities have been scaled. See the appendix for details.

3.5.2 Colored Noise Case

When the measurement noise n_k is colored (*i.e.*, temporally correlated), the dual EKF algorithms must be adjusted somewhat. The cost functions for the colored noise case are shown in Sections 2.3.2 and 2.4. The dual EKF algorithms for colored noise parallel those developed on the preceding pages for white noise, with some modification.

Decoupling with Direct Substitution – Colored Noise

The joint cost function for the case of colored measurement noise is given in Equation 2.29 on page 30 as:

$$J_c^j(\mathbf{x}_1^N, \mathbf{n}_1^N, \mathbf{w}) = \sum_{k=1}^{N}\left(\frac{(x_k - \bar{x_k})^2}{\sigma_v^2} + \frac{(n_k - \bar{n_k})^2}{\sigma_{v_n}^2}\right),$$

where $\bar{n_k} = \sum_{i=1}^{M_n} w_n^{(i)} \cdot n_{k-i}$ is the predicted value of the colored noise, and $\sigma_{v_n}^2$ is the variance of the process noise that drives the colored noise model.

As explained in Chapter 2, the cost must be minimized subject to the constraint $\mathbf{y}_1^N = \mathbf{x}_1^N + \mathbf{n}_1^N$. The joint EKF algorithm for sequentially minimizing this cost was developed in Section 3.4.2 on page 67 by concatenating the signal, noise, and weights into a joint state-vector.

The joint cost can be minimized with respect to the signal and noise while fixing the weights at the current estimate $\hat{\mathbf{w}}_k$, and minimized with respect to the weights while fixing the signal and noise arguments at *their* current estimates. As in the white noise case, this can be done

Initialize with:

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

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

$$\hat{\mathbf{w}}_k^- = \hat{\mathbf{w}}_{k-1} \tag{3.183}$$
$$\mathbf{Q}_k^- = \mathbf{Q}_{k-1} + \mathbf{U}_k \tag{3.184}$$

and for the signal filter are:

$$\hat{\boldsymbol{\xi}}_k^- = \mathbf{F}(\hat{\boldsymbol{\xi}}_{k-1}, \hat{\mathbf{w}}_k^-) \tag{3.185}$$
$$\mathbf{P}_{c,k}^- = \mathbf{A}_{c,k-1}\mathbf{P}_{c,k-1}\mathbf{A}_{c,k-1}^T + \mathbf{B}_c\mathbf{V}_c\mathbf{B}_c^T \tag{3.186}$$

The measurement update equations for the signal filter are:

$$\mathbf{K}_k = \mathbf{P}_{c,k}^-\mathbf{C}^T(\mathbf{C}_c\mathbf{P}_{c,k}^-\mathbf{C}_c^T)^{-1} \tag{3.187}$$
$$\hat{\boldsymbol{\xi}}_k = \hat{\boldsymbol{\xi}}_k^- + \mathbf{K}_k(y_k - \mathbf{C}_c\hat{\boldsymbol{\xi}}_k^-) \tag{3.188}$$
$$\mathbf{P}_{c,k} = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_{c,k}^- \tag{3.189}$$

and for the weight filter are:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^-\mathbf{H}_{o,k}^T(\mathbf{H}_{o,k}\mathbf{Q}_k^-\mathbf{H}_{o,k}^T + \sigma_r^2)^{-1} \tag{3.190}$$
$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} \cdot \boldsymbol{\epsilon}_k \tag{3.191}$$
$$\mathbf{Q}_k = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}}\mathbf{H}_{o,k})\mathbf{Q}_k^-. \tag{3.192}$$

Formula 3.21: The dual extended Kalman filter equations for colored measurement noise. The definitions of $\boldsymbol{\epsilon}_k$ and $\mathbf{H}_{o,k}$ will depend on the cost function used for weight estimation. See the text for details.

in batch mode with an errors-in-variables algorithm, as shown in Appendix G. However, on-line applications will require a sequential algorithm such as the dual EKF approach.

Signal and Noise Estimation - Colored Noise

As described in Section 2.3.2, the signal and measurement noise are estimated simultaneously by minimizing $J_c^j(\mathbf{x}_1^k, \mathbf{n}_1^k, \mathbf{w})$, evaluated at the current weight estimate $\hat{\mathbf{w}}$. The batch cost is shown in Equation 2.30 on page 31, restated here for sequential estimation as:

$$J_c^j(\mathbf{x}_1^k, \mathbf{n}_1^k, \hat{\mathbf{w}}) = \sum_{t=1}^{k}\left(\frac{(x_t - \hat{x}_t^-)^2}{\sigma_v^2} + \frac{(n_t - n_t^-)^2}{\sigma_{v_n}^2}\right), \tag{3.193}$$

minimized subject to the constraint $y_t = x_t + n_t$. This constraint can be satisfied by estimating the signal and noise within a *combined* state-space representation, as explained in Section 3.2, and applying the Kalman filter equations given in Formula 3.3 on page 53 and Formula 3.4 on page 54, except with $\hat{\mathbf{w}}_k$ used instead of $\mathbf{w}$. This forms the signal estimation portion of the dual EKF equations given in Formula 3.21 on the previous page.

Weight Estimation – Colored Noise

Minimizing the joint cost with respect to $\mathbf{w}$ produces estimates of the weights. The cost is evaluated using the estimates of the signal and noise (see Equation 2.31 on page 31):

$$J(\hat{\mathbf{x}}_1^k, \hat{\mathbf{n}}_1^k, \mathbf{w}) = \sum_{t=1}^{k} \left(\frac{(\hat{x}_t - \hat{x}_t^-)^2}{\sigma_v^2} + \frac{(\hat{n}_t - \hat{n}_t^-)^2}{\sigma_{v_n}^2} \right), \tag{3.194}$$

where $\hat{x}_t^- = f(\hat{\mathbf{x}}_{t-1}, \mathbf{w})$ and $\hat{n}_t^- = \sum_{i=1}^{M_n} w_n^{(i)} \cdot \hat{n}_{t-i}$. As in the white noise case, the signal estimates $\{\hat{x}_t\}_1^k$ are not necessarily a function of the weights $\mathbf{w}$; the same is true for the noise estimates $\{\hat{n}_t\}_1^k$. In this case, the first term alone will be minimized to generate $\hat{\mathbf{w}}$, as described on page 79.

However, both $\{\hat{x}_k\}_1^N$ and $\{\hat{n}_k\}_1^N$ will typically be generated by a Kalman filter as described above; the KF signal estimates will clearly be a function of $\mathbf{w}$. In addition, however, because of the constraint $\mathbf{y}_1^k = \mathbf{x}_1^k + \mathbf{n}_1^k$, the noise estimates will *also* be a function of $\mathbf{w}$, as any change in $\hat{x}_k$ will result in an equal but opposite change in $\hat{n}_k$. Furthermore, if a KF (or EKF) is used to generate the signal and noise estimates, then $\hat{x}_k$, $\hat{x}_k^-$, $\hat{n}_k$, and $\hat{n}_k^-$ will all be *recursive* functions of the weights.

To estimate the weights with an observed-error weight filter, the instantaneous cost is defined as

$$J_k = \frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2} + \frac{(\hat{n}_k - \hat{n}_k^-)^2}{\sigma_{v_n}^2} \qquad \text{or} \quad J_k = \frac{\tilde{\hat{x}}_k^2}{\sigma_v^2} + \frac{\tilde{\hat{n}}_k^2}{\sigma_{v_n}^2}, \tag{3.195}$$

where $\tilde{\hat{x}}_k \triangleq (\hat{x}_k - \hat{x}_k^-)$ and $\tilde{\hat{n}}_k \triangleq (\hat{n}_k - \hat{n}_k^-)$. The gradient and Hessian of J_k are approximated as described on page 62 by defining a vector form of the observed-error as in Formula 3.22, so that

$$\mathbf{e}_k \triangleq \begin{bmatrix} \sigma_v^{-1} \tilde{\hat{x}}_k \\ \sigma_{v_n}^{-1} \tilde{\hat{n}}_k \end{bmatrix}, \qquad \text{with negative Jacobian} \quad \mathbf{H}_{o,k} = -\begin{bmatrix} \sigma_v^{-1} \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k \\ \sigma_{v_n}^{-1} \nabla_{\mathbf{w}}^T \tilde{\hat{n}}_k \end{bmatrix}$$

Formula 3.22: Colored noise joint cost function. Observed-error terms for dual EKF weight filter.

$\mathbf{e}_k^T \mathbf{e}_k = J_k$, as required. Letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$, the negative gradient is produced by $\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{e}_k = -\nabla_{\mathbf{w}} J_k$, as shown in Appendix E, and a first-order approximation to the instantaneous Hessian $\nabla_{\mathbf{w}}^2 J_k$ is

given by: $\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k}$. Hence, the weight estimate portion of the dual EKF shown in Formula 3.21 represents a modified-Newton update for minimizing the joint cost function.

The derivatives contained in $\mathbf{H}_{o,k}$ above evaluate as:

$$\nabla_{\mathbf{w}} \tilde{\hat{x}}_k = (\nabla_{\mathbf{w}} \hat{x}_k - \nabla_{\mathbf{w}} \hat{x}_k^-), \quad \text{and} \quad \nabla_{\mathbf{w}} \tilde{\hat{n}}_k = (\nabla_{\mathbf{w}} \hat{n}_k - \nabla_{\mathbf{w}} \hat{n}_k^-), \tag{3.196}$$

and are computed using the 1^{st} and $(1+M)^{th}$ rows of $\nabla_{\mathbf{w}} \hat{\boldsymbol{\xi}}_k$ and $\nabla_{\mathbf{w}} \hat{\boldsymbol{\xi}}_k^-$. The derivatives of $\hat{\boldsymbol{\xi}}_k$ and $\hat{\boldsymbol{\xi}}_k^-$ must be computed recursively, following the framework given in Section 3.6.1 on page 102 for the derivatives of $\hat{\mathbf{x}}_k$.

Variance Estimation – Colored Noise

When the variance terms σ_v^2 and $\sigma_{v_n}^2$ are not known, they can be estimated by minimizing the cost function given in Equation 2.32.

$$J_c^j(\sigma^2) = \sum_{t=1}^{k} \Bigg(\log(2\pi\sigma_v^2) + \frac{(\hat{x}_t - \hat{x}_t^-)^2}{\sigma_v^2} + \log(2\pi\sigma_{v_n}^2) + \frac{(\hat{n}_t - \hat{n}_t^-)^2}{\sigma_{v_n}^2} \Bigg).$$

Kalman filter estimates of the signal and noise will produce errors $\tilde{\hat{x}}_k$ and $\tilde{\hat{n}}_k$ which are functions of σ_v^2 and $\sigma_{v_n}^2$ respectively. Also, if $\hat{x}_k$ and $\hat{n}_k$ are constrained to sum to y_k, then $\tilde{\hat{x}}_k$ will be a function of $\sigma_{v_n}^2$ by way of its dependence on $\tilde{\hat{n}}_k$. A modified-Newton algorithm is found for each variance by putting it in an observed-error state-space representation as on page 74. The observed-error is defined as in Formula 3.23 to give the desired estimation algorithm. Similar to the discussion on

$$\breve{\boldsymbol{\epsilon}}_k \triangleq \begin{bmatrix} (\ell_v)^{\frac{1}{2}} \\ \sigma_v^{-1} \tilde{\hat{x}}_k \\ (\ell_{v_n})^{\frac{1}{2}} \\ \sigma_{v_n}^{-1} \tilde{\hat{n}}_k \end{bmatrix}, \quad \text{and} \quad \breve{\mathbf{H}}_{o,k} = \begin{bmatrix} -\frac{1}{2} \frac{(\ell_v)^{-\frac{1}{2}}}{\sigma_v^2} \frac{\partial \sigma_v^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_v} \frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} + \frac{\tilde{\hat{x}}_k}{2(\sigma_v^2)^{(3/2)}} \frac{\partial \sigma_v^2}{\partial \sigma^2} \\ -\frac{1}{2} \frac{(\ell_{v_n})^{-\frac{1}{2}}}{\sigma_{v_n}^2} \frac{\partial \sigma_{v_n}^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_{v_n}} \frac{\partial \tilde{\hat{n}}_k}{\partial \sigma^2} + \frac{\tilde{\hat{n}}_k}{2(\sigma_{v_n}^2)^{(3/2)}} \frac{\partial \sigma_{v_n}^2}{\partial \sigma^2} \end{bmatrix}$$

Formula 3.23: Colored noise joint cost function. Observed-error terms for dual EKF variance filter.

page 76 for the white noise case, the approximation to the second derivative is improved by using the redefinitions:

$$\ell_{v_n,k} = \frac{\sigma_n^2}{3\tilde{\hat{n}}_k^2 - 2\sigma_{v_n}^2} \quad \text{and} \quad \ell_{v,k} = \frac{\sigma_v^2}{3\tilde{\hat{x}}_k^2 - 2\sigma_v^2} \tag{3.197}$$

for all time k. With these adjustments in place, the variance filter equations are the same as for the white noise case, shown in Formula 3.13 on page 78.

The derivatives $\frac{\partial \sigma_{v_n}^2}{\partial \sigma^2}$ and $\frac{\partial \sigma_v^2}{\partial \sigma^2}$ evaluate to either 0 or 1, depending on whether $\sigma^2 = \sigma_{v_n}^2$ or $\sigma^2 = \sigma_v^2$. The other derivatives:

$$\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} = \left(\frac{\partial \hat{x}_k}{\partial \sigma^2} - \frac{\partial \hat{x}_k^-}{\partial \sigma^2}\right), \quad \text{and} \quad \frac{\partial \tilde{\hat{n}}_k}{\partial \sigma^2} = \left(\frac{\partial \hat{n}_k}{\partial \sigma^2} - \frac{\partial \hat{n}_k^-}{\partial \sigma^2}\right), \tag{3.198}$$

are found by using the 1^{st} and $(1+M)^{th}$ elements of $\frac{\partial \hat{\boldsymbol{\xi}}_k}{\partial \sigma^2}$ and $\frac{\partial \hat{\boldsymbol{\xi}}_k^-}{\partial \sigma^2}$, which are themselves computed recursively.

Error Coupling – Colored Noise

As described on page 78 for the white noise case, information about the errors in the current estimates can be used to modify the colored noise cost functions minimized by the dual EKF.

Signal and Colored Noise Estimation

When estimating the signal and noise, the error in the weights is accounted for by modeling the resultant error $\tilde{f}_k$ in the dynamics as a white Gaussian noise process. The cost function is given in Equation 2.34 on page 32 as:

$$J_c^{ec}(\mathbf{x}_1^N, \mathbf{n}_1^N) = \sum_{k=1}^{N} \left(\frac{(x_k - \hat{x}_k^-)^2}{(\sigma_{\tilde{f}}^2 + \sigma_v^2)} + \frac{(n_k - n_k^-)^2}{\sigma_{v_n}^2} + \log(2\pi(\sigma_{\tilde{f},k}^2 + \sigma_v^2)) \right),$$

minimized subject to $y_k = x_k + n_k$. As shown in Equation 3.155 on page 79, $\sigma_{\tilde{f},k}^2 \approx \mathbf{H}_k \mathbf{Q}_k^- \mathbf{H}_k^T$. In sequential estimation, this cost will be minimized with respect to only the current signal and noise values: x_k and n_k. The log term is therefore dropped, because $\sigma_{\tilde{f},k}^2$ and σ_v^2 are both functionally independent of x_k. The remaining two terms of $J_c^{ec}(\mathbf{x}_1^N, \mathbf{n}_1^N)$ constitute the cost minimized by the signal filter portion of the dual EKF given in Formula 3.21 on page 90, except that σ_v^2 is replaced by $(\sigma_{\tilde{f}}^2 + \sigma_v^2)$ in the definition of $\mathbf{V}_v$.

Weight Estimation – Colored Noise

For weight estimation, the errors in the signal and noise estimates are accounted for by using the cost given in Equation 2.41 on page 33:

$$J_c^{ec}(\mathbf{w}) = \sum_{k=1}^{N} \left(\log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} + \log(2\pi g_{n,k}) + \frac{(\hat{n}_k - \hat{n}_k^-)^2}{g_{n,k}} \right),$$

where g_k and $g_{n,k}$ are the variances of $\tilde{\hat{x}}_k$ and $\tilde{\hat{n}}_k$. If the signal and noise estimates are from the colored noise Kalman filter described above, g_k and $g_{n,k}$ are calculated as:

$$\begin{aligned} g_k &= E[(\hat{x}_k - \hat{x}_k^-)^2] && \text{(3.199a)} \\ &= E\left[[\mathbf{C}\ 0](\hat{\boldsymbol{\xi}}_k - \hat{\boldsymbol{\xi}}_k^-)(\hat{\boldsymbol{\xi}}_k - \hat{\boldsymbol{\xi}}_k^-)^T[\mathbf{C}\ 0]^T\right] && \text{(3.199b)} \\ &= E\left[[\mathbf{C}\ 0]\mathbf{K}_k(y_k - \mathbf{C}_c\hat{\boldsymbol{\xi}}_k^-)^2\mathbf{K}_k^T[\mathbf{C}\ 0]^T\right] && \text{(3.199c)} \\ &= E[\mathbf{K}_k^{(1)}(\mathbf{C}_c\boldsymbol{\xi}_k - \mathbf{C}_c\hat{\boldsymbol{\xi}}_k^-)^2\mathbf{K}_k^{(1)}] && \text{(3.199d)} \\ &= \mathbf{K}_k^{(1)}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T)\mathbf{K}_k^{(1)}, && \text{(3.199)} \end{aligned}$$

and similarly

$$\begin{aligned} g_{n,k} &= E[(\hat{n}_k - \hat{n}_k^-)^2] && \text{(3.200a)} \\ &= E\left[[0\ \mathbf{C}_n](\hat{\boldsymbol{\xi}}_k - \hat{\boldsymbol{\xi}}_k^-)(\hat{\boldsymbol{\xi}}_k - \hat{\boldsymbol{\xi}}_k^-)^T[0\ \mathbf{C}_n]^T\right] && \text{(3.200b)} \\ &= E\left[[0\ \mathbf{C}_n]\mathbf{K}_k(y_k - \mathbf{C}_c\hat{\boldsymbol{\xi}}_k^-)^2\mathbf{K}_k^T[0\ \mathbf{C}_n]^T\right] && \text{(3.200c)} \\ &= E[\mathbf{K}_k^{(1+M)}(\mathbf{C}_c\boldsymbol{\xi}_k - \mathbf{C}_c\hat{\boldsymbol{\xi}}_k^-)^2\mathbf{K}_k^{(1+M)}] && \text{(3.200d)} \\ &= \mathbf{K}_k^{(1+M)}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T)\mathbf{K}_k^{(1+M)}, && \text{(3.200)} \end{aligned}$$

where $[\mathbf{C}\ 0]$ and $[0\ \mathbf{C}_n]$ are $(M + M_n)$-dimensional row vectors containing all zeros except for the 1^{st} and $(1+M)^{th}$ elements, respectively. Hence $[\mathbf{C}\ 0]\mathbf{K}_k = \mathbf{K}_k^{(1)}$ and $[0\ \mathbf{C}_n]\mathbf{K}_k = \mathbf{K}_k^{(1+M)}$.

The gradients with respect to the weights are easily calculated from these expressions as:

$$\nabla_{\mathbf{w}} g_k = 2\mathbf{K}_k^{(1)}\nabla_{\mathbf{w}}\mathbf{K}_k^{(1)}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T) + (\mathbf{K}_k^{(1)})^2\nabla_{\mathbf{w}}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T), \quad \text{and} \tag{3.201}$$

$$\nabla_{\mathbf{w}} g_{n,k} = 2\mathbf{K}_k^{(1+M)}\nabla_{\mathbf{w}}\mathbf{K}_k^{(1+M)}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T) + (\mathbf{K}_k^{(1+M)})^2\nabla_{\mathbf{w}}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T). \tag{3.202}$$

The instantaneous cost J_k is the term inside the summation in Equation 2.41. The gradient and Hessian of J_k are approximated by defining the observed-error term as in Formula 3.24. Using the values $\ell_{g,k} = g_k/(3\tilde{\hat{x}}_k^2 - 2g_k)$, and $\ell_{g_n} = g_{n,k}/(3\tilde{\hat{n}}_k^2 - 2g_{n,k})$, the negative gradient of J_k is

$$\boldsymbol{\epsilon}_k = \begin{bmatrix} (\ell_g)^{\frac{1}{2}} \\ g_k^{(-1/2)}\tilde{\hat{x}}_k \\ (\ell_{g_n})^{\frac{1}{2}} \\ g_{n,k}^{(-1/2)}\tilde{\hat{n}}_k \end{bmatrix}, \quad \text{and} \quad \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_g)^{-\frac{1}{2}}}{g_k}\nabla_{\mathbf{w}}^T(g_k) \\ -\frac{1}{g_k^{(1/2)}}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k + \frac{\tilde{\hat{x}}_k}{2g_k^{(3/2)}}\nabla_{\mathbf{w}}^T(g_k) \\ -\frac{1}{2}\frac{(\ell_{g_n})^{-\frac{1}{2}}}{g_{n,k}}\nabla_{\mathbf{w}}^T(g_{n,k}) \\ -\frac{1}{g_{n,k}^{(1/2)}}\nabla_{\mathbf{w}}\tilde{\hat{n}}_k + \frac{\tilde{\hat{n}}_k}{2g_{n,k}^{(3/2)}}\nabla_{\mathbf{w}}^T(g_{n,k}) \end{bmatrix}$$

Formula 3.24: Colored noise error-coupled cost function. Observed-error terms for dual EKF weight filter.

given by $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{e}_k = -\nabla_{\mathbf{w}} J_k$, and the Hessian of J_k is approximated by $\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{H}_{o,k}$, as shown in Appendix E.

Variance Estimation – Colored Noise

Estimation of the variance terms $\sigma_{v_n}^2$ and σ_v^2 can be done in a way that takes the errors in the signal, noise, and weight estimates into account. The cost function is given in Equation 2.45 as:

$$J_c^{ec}(\sigma^2) = \sum_{k=1}^{N}\left(\log(2\pi g_k) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{g_k} + \log(2\pi g_{n,k}) + \frac{(\hat{n}_k - \hat{n}_k^-)^2}{g_{n,k}}\right),$$

which is identical to the cost given for weight estimation except that predictions here are given by $\hat{x}_k^- = f\big(\hat{x}_{k-1}, \ldots \hat{x}_{k-M}, \hat{\mathbf{w}}\big)$. The error-variance terms are $g_k = \mathbf{K}_k^{(1)}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T)\mathbf{K}_k^{(1)}$ and $g_{n,k} = \mathbf{K}_k^{(1+M)}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T)\mathbf{K}_k^{(1+M)}$, as shown in Equation 3.199 and 3.200 on the preceding page. The redefinition of $\hat{x}_k^-$ is reflected in $\mathbf{P}_k^-$ since this is produced by the error-coupled signal filter, which makes use of the statistics of $\hat{\mathbf{w}}$.

The instantaneous cost J_k is the quantity inside the above summation. The variance terms are estimated by defining the observed-error measurement $\breve{\mathbf{e}}_k$ the same as $\mathbf{e}_k$ in Formula 3.24 on the previous page; the negative of the corresponding derivative is given in Formula 3.25, where the

$$\breve{\mathbf{e}}_k = \begin{bmatrix} (\ell_g)^{\frac{1}{2}} \\ g_k^{(-1/2)}\tilde{\tilde{x}}_k \\ (\ell_{g_n})^{\frac{1}{2}} \\ g_{n,k}^{(-1/2)}\tilde{\tilde{n}}_k \end{bmatrix}, \quad \text{and} \quad \breve{\mathbf{H}}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_g)^{-\frac{1}{2}}}{g_k}\frac{\partial g_k}{\partial \sigma^2} \\ -\frac{1}{g_k^{(1/2)}}\frac{\partial \tilde{\tilde{x}}_k}{\partial \sigma^2} + \frac{\tilde{\tilde{x}}_k}{2g_k^{(3/2)}}\frac{\partial g_k}{\partial \sigma^2} \\ -\frac{1}{2}\frac{(\ell_{g_n})^{-\frac{1}{2}}}{g_{n,k}}\frac{\partial g_{n,k}}{\partial \sigma^2} \\ -\frac{1}{g_{n,k}^{(1/2)}}\frac{\partial \tilde{\tilde{n}}_k}{\partial \sigma^2} + \frac{\tilde{\tilde{n}}_k}{2g_{n,k}^{(3/2)}}\frac{\partial g_{n,k}}{\partial \sigma^2} \end{bmatrix}$$

Formula 3.25: Colored noise error-coupled cost function. Observed-error terms for dual EKF variance filter.

derivatives $\frac{\partial g_k}{\partial \sigma^2}$ and $\frac{\partial g_{n,k}}{\partial \sigma^2}$ are:

$$\frac{\partial g_k}{\partial \sigma^2} = 2\mathbf{K}_k^{(1)}\frac{\partial \mathbf{K}_k^{(1)}}{\partial \sigma^2}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T) + (\mathbf{K}_k^{(1)})^2\frac{\partial(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T)}{\partial \sigma^2}, \quad \text{and} \tag{3.203}$$

$$\frac{\partial g_{n,k}}{\partial \sigma^2} = 2\mathbf{K}_k^{(1+M)}\frac{\partial \mathbf{K}_k^{(1+M)}}{\partial \sigma^2}(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T) + (\mathbf{K}_k^{(1+M)})^2\frac{\partial(\mathbf{C}_c\mathbf{P}_k^-\mathbf{C}_c^T)}{\partial \sigma^2}. \tag{3.204}$$

With these quantities in hand, the variances can be estimated as described before, with the algorithm in Formula 3.13 on page 78.

Marginal Estimation – Colored Noise

As described in Section 2.4 on page 34, a marginal estimation approach for colored measurement noise is found by expanding the joint density function $\rho_{\mathbf{x}_1^N \mathbf{n}_1^N \mathbf{w}|\mathbf{y}_1^N}$ into two factors: $\rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{y}_1^N \mathbf{w}}$ and $\rho_{\mathbf{w}|\mathbf{y}_1^N}$. Marginal estimation methods maximize the first factor with respect to the signal, and the second factor with respect to the weights.

Signal and Noise Estimation

The signal and noise estimates that maximize $\rho_{\mathbf{x}_1^N \mathbf{n}_1^N|\mathbf{y}_1^N \mathbf{w}}$ are found by substituting the current weight estimates $\hat{\mathbf{w}}_k^-$ for $\mathbf{w}$ in the colored-noise Kalman filter of Formula 3.3 on page 53, and Formula 3.4. This produces sequential estimates $\hat{\mathbf{x}}_1^N$ and $\hat{\mathbf{n}}_1^N$.

Weight Estimation

Meanwhile, $\rho_{\mathbf{w}|\mathbf{y}_1^N}$ is maximized with respect to the weights by maximizing the log:

$$\log \rho_{\mathbf{Y}|\mathbf{W}} = -\frac{1}{2} \sum_{k=1}^{N} \left(\log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \overline{y_{k|k-1}})^2}{\sigma_{\varepsilon_k}^2} \right), \tag{3.205}$$

where $\varepsilon_k \triangleq (y_k - \overline{y_{k|k-1}})$. This expression is identical to that given for the white noise case except that, for colored measurement noise, the mean is calculated as:

$$\begin{aligned} \overline{y_{k|k-1}} &= E[y_k | \{y_t\}_1^{k-1}, \mathbf{w}] && (3.206a) \\ &= E[x_k + n_k | \{y_t\}_1^{k-1}, \mathbf{w}] \qquad = \hat{x}_k^- + \hat{n}_k^- && (3.206b) \\ &= \mathbf{C}_c \cdot \hat{\boldsymbol{\xi}}_k^- && (3.206) \end{aligned}$$

and the variance is given by:

$$\begin{aligned} \sigma_{\varepsilon_k}^2 &= E[(y_k - \overline{y_{k|k-1}})^2 | \{y_t\}_1^{k-1}, \mathbf{w}] && (3.207a) \\ &= E[(x_k - \hat{x}_k^- + n_k - \hat{n}_k^-)^2 | \{y_t\}_1^{k-1}, \mathbf{w}] && (3.207b) \\ &= \mathbf{C}_c \mathbf{P}_{c,k}^- \mathbf{C}_c^T. && (3.207) \end{aligned}$$

Both terms are computed by the colored-noise Kalman signal filter, and thus are recursive functions of the weights $\mathbf{w}$. Their gradients are

$$\begin{aligned} \nabla_{\mathbf{w}}(\overline{y_{k|k-1}}) &= \mathbf{C}_c \cdot \nabla_{\mathbf{w}} \hat{\boldsymbol{\xi}}_k^- && (3.208) \\ \nabla_{\mathbf{w}}(\sigma_{\varepsilon_k}^2) &= \nabla_{\mathbf{w}}(\mathbf{C}_c \mathbf{P}_{c,k}^- \mathbf{C}_c^T) && (3.209) \end{aligned}$$

and must be computed recursively, as shown in Section 3.6.1.

Variance Estimation

Unknown noise variances can also be estimated by minimizing the negative log in Equation 3.205 on the previous page. From Equations 3.206 and 3.207, the required derivatives are:

$$\frac{\partial \varepsilon_k}{\partial \sigma^2} = -\left(\frac{\partial \hat{x}_k^-}{\partial \sigma^2} + \frac{\partial \hat{n}_k^-}{\partial \sigma^2}\right) \qquad = -\mathbf{C}_c \frac{\partial \hat{\boldsymbol{\xi}}_k^-}{\partial \sigma^2} \tag{3.210}$$

$$\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} = \frac{\partial \mathbf{P}_{c,k}^{-(1,1)}}{\partial \sigma^2} + 2\frac{\partial \mathbf{P}_{c,k}^{-(1,1+M)}}{\partial \sigma^2} + \frac{\partial \mathbf{P}_{c,k}^{-(1+M,1+M)}}{\partial \sigma^2} \quad = \mathbf{C}_c \frac{\partial \mathbf{P}_{c,k}}{\partial \sigma^2} \mathbf{C}_c^T. \tag{3.211}$$

Section 3.6.1 shows the recursive computation of $\frac{\partial \hat{\boldsymbol{\xi}}_k^-}{\partial \sigma^2}$ and $\frac{\partial \mathbf{P}_{c,k}}{\partial \sigma^2}$.

Prediction Error Weight Estimation - Colored Noise

If $\sigma_{\varepsilon_k}^2$ is assumed to be independent of $\mathbf{w}$, then the log term can be dropped from the cost function, leaving the squared prediction error cost:

$$J_c^{pe}(\mathbf{w}) = \sum_{k=1}^{N} \varepsilon_k^2, \tag{3.212}$$

where $\varepsilon_k \triangleq (y_k - (\hat{x}_k^- + \hat{n}_k^-))$. This cost corresponds to the recursive prediction-error form of the colored-noise dual EKF developed in [88].

The observed-error form of weight filter is found by defining the instantaneous cost as $J_k = (y_k - \hat{x}_k^- - \hat{n}_k^-)^2 = \varepsilon_k^2$, and letting the observed error terms be defined as in Formula 3.26, so that

$$\mathbf{e}_k \triangleq \varepsilon_k \quad \text{and} \qquad \mathbf{H}_{o,k} = -\nabla_{\mathbf{w}} \varepsilon_k \quad = \nabla_{\mathbf{w}} \overline{y_{k|k-1}} = (\nabla_{\mathbf{w}} \hat{x}_k^- + \nabla_{\mathbf{w}} \hat{n}_k^-)$$

Formula 3.26: Colored noise prediction-error cost function. Observed-error terms for dual EKF weight filter.

the negative gradient of J_k is given by $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{e}_k = -2(\nabla_{\mathbf{w}}\varepsilon_k)\varepsilon_k$, and the Hessian is approximated to first-order by $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{H}_{o,k}^T = 2(\nabla_{\mathbf{w}}\varepsilon_k)(\nabla_{\mathbf{w}}\varepsilon_k)^T$.

Alternatively, the prediction error cost can be defined using $\varepsilon_k \triangleq (y_k - (\hat{x}_k^- + \hat{n}_k))$, which replaces the noise *prediction* with its *estimate*. Although this is a further departure from the original probabilistic approach, it has a simple heuristic justification. Namely, using $y_k - \hat{n}_k$ as the prediction target is the next best thing to predicting $x_k = y_k - n_k$, and will produce lower variance weight estimates than using y_k as the target. Furthermore, the weights should be adjusted to make $\hat{x}_k^-$ a better prediction, without regard to $\hat{n}_k^-$ or $\hat{n}_k$.

This last comment suggests that only the signal prediction $\hat{x}_k^-$ should be considered in the derivatives. This "alternative" approach, shown in Formula 3.27, shows superior performance to

that provided by using $\varepsilon_k \triangleq (y_k - (\hat{x}_k^- + \hat{n}_k^-))$, and is therefore used in the next chapter to represent the colored-noise prediction error method.

$$\boldsymbol{\epsilon}_k \triangleq ([y_k - \hat{n}_k] - \hat{x}_k^-) \quad \text{and} \quad \mathbf{H}_{o,k} = -\nabla_{\mathbf{w}} \hat{x}_k^-$$

Formula 3.27: Alternative colored noise prediction-error cost function. Observed-error terms for dual EKF weight filter.

Prediction Error Variance Estimation - Colored Noise

Either of the noise variances, σ_v^2 and $\sigma_{v_n}^2$, can be estimated using the above "alternative" prediction error cost by defining $\breve{\boldsymbol{\epsilon}}_k \triangleq ([y_k - \hat{n}_k] - \hat{x}_k^-)$ as before, and $\breve{\mathbf{H}}_{o,k} = -\frac{\partial \hat{x}_k^-}{\partial \sigma^2}$. This differs from the white noise case only in the inclusion of $\hat{n}_k$ in $\breve{\boldsymbol{\epsilon}}_k$.

Maximum Likelihood Weight Estimation - Colored Noise

Taking the dependence of $\sigma_{\varepsilon_k}^2$ on $\mathbf{w}$ into account requires the minimization of the full maximum-likelihood cost function given in Equation 2.52 on page 36, restated here as:

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

The appropriate weight filter is found by defining the observed-error and its negative derivative just as they appear in Formula 3.17 on page 84. The only difference is in the definition $\varepsilon_k \triangleq (y_k - (\hat{x}_k^- + \hat{n}_k^-))$, and its variance $\sigma_{\varepsilon_k}^2$.

Maximum Likelihood Variance Estimation - Colored Noise

Similarly, the variance terms σ_v^2 and $\sigma_{v_n}^2$ can be estimated by minimizing the same cost. Here, $\breve{\boldsymbol{\epsilon}}_k$ and $\breve{\mathbf{H}}_{o,k}$ are defined the same as in Formula 3.18 on page 85. The desired variance is estimated according to the equations in Formula 3.13 on page 78.

Expectation Maximization – Colored Noise

Just as in the white noise case, an alternative way of maximizing the marginal likelihood is supplied by the EM algorithm. As given in Equation 2.58 on page 37, the colored-noise EM cost is:

$$J_c^{em} = E_{\mathbf{x}_1^N \mathbf{n}_1^N | \mathbf{y}_1^N \mathbf{w}} \left[\sum_{k=1}^{N} \left(\log(2\pi\sigma_v^2) + \frac{(x_k - x_k^-)^2}{\sigma_v^2} + \log(2\pi\sigma_{v_n}^2) + \frac{(n_k - n_k^-)^2}{\sigma_{v_n}^2} \right) \Bigg| \{y_k\}_1^N, \hat{\mathbf{w}} \right],$$

As shown in Appendix F, the expectation evaluates to:

$$J_c^{em} = N \log(4\pi^2 \sigma_v^2 \sigma_{v_n}^2) + \sum_{k=1}^{N} \left(\frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 + p_{k|N} - 2p_{k|N}^\dagger + p_{k|N}^-}{\sigma_v^2} + \frac{(\hat{n}_{k|N} - \hat{n}_{k|N}^-)^2 + p_{n,k|N} - 2p_{n,k|N}^\dagger + p_{n,k|N}^-}{\sigma_{v_n}^2} \right) \tag{3.213}$$

where $\hat{n}_{k|N}$, $p_{n,k|N}$ are defined as the conditional mean and variance of n_k given $\mathbf{w}$ and the data $\{y_k\}_1^N$. The terms $\hat{n}_{k|N}^-$ and $p_{n,k|N}^-$ are the conditional mean and variance, respectively, of $n_k^- = \mathbf{w}_n^T \cdot \mathbf{n}_{k-1}$. The additional term $p_{n,k|N}^\dagger$ represents the cross-variance of n_k and n_k^-, conditioned on all the data. The corresponding terms for the signal x_k are defined on page 86, following Equation 3.173.

An iterative-batch EM algorithm for the case of colored measurement noise is suggested (without equations) by Gannot et al. in [21]. A sequential EM algorithm is produced by computing the expectations in the E-step with a colored-noise Kalman filter. This is equivalent to replacing the off-line means and covariances in Equation 3.213 with their on-line equivalents. As in the white-noise case, these on-line statistics are found by augmenting the combined state-vector with one additional lagged value for both the signal and noise. Specifically:

$$\boldsymbol{\xi}_k^+ = \begin{bmatrix} \mathbf{x}_k \\ x_{k-M} \\ \mathbf{n}_k \\ n_{k-M_n} \end{bmatrix} = \begin{bmatrix} x_k \\ \mathbf{x}_{k-1} \\ n_k \\ \mathbf{n}_{k-1} \end{bmatrix}, \tag{3.214}$$

so that the estimate $\hat{\boldsymbol{\xi}}_k^+$ produced by a Kalman filter will contain $\hat{\mathbf{x}}_{k-1|k}$ in elements 2 through $1+M$, and $\hat{\mathbf{n}}_{k-1|k}$ in its last M_n elements.

Furthermore, the covariance $\mathbf{P}_{c,k}^+$ of $\boldsymbol{\xi}_k^+$ produced by the KF allows for approximate calculation of the variances p_k, $p_{k|k}^-$, $p_{k|k}^\dagger$, $p_{n,k}$, $p_{n,k|k}^-$ and $p_{n,k|k}^\dagger$. Denote the first $(1+M) \times (1+M)$ block diagonal of $\mathbf{P}_{c,k}^+$ as $\mathbf{P}_{x,k}^+$. Then let $\mathbf{P}_{x,k-1|k}$ be the lower right block of $\mathbf{P}_{x,k}^+$, and $\mathbf{P}_{x,k}^\sharp$ be the

upper right block of $\mathbf{P}^{+}_{x,k}$. Defining $\mathbf{P}^{-}_{x,k|k} \triangleq \mathbf{A}_{k-1|k}(\mathbf{P}_{x,k-1|k})\mathbf{A}^{T}_{k-1|k}$ and $\mathbf{P}^{\dagger}_{x,k|k} \triangleq \mathbf{P}^{\sharp}_{x,k}\mathbf{A}^{T}_{k-1|k}$ gives:

$$p_{k|k} = (\mathbf{P}^{+}_{x,k})^{(1,1)}, \quad p^{-}_{k|k} = (\mathbf{P}^{-}_{x,k|k})^{(1,1)}, \quad \text{and} \quad p^{\dagger}_{k|k} = (\mathbf{P}^{\dagger}_{x,k|k})^{(1,1)}, \tag{3.215}$$

as shown in Appendix F. Similar calculations can be made for the noise statistics using the last $(M_n+1) \times (M_n+1)$ block diagonal of $\mathbf{P}^{+}_{c,k}$, denoted $\mathbf{P}^{+}_{n,k}$. Specifically:

$$p_{n,k|k} = (\mathbf{P}^{+}_{n,k})^{(1,1)}, \quad p^{-}_{n,k|k} = (\mathbf{P}^{-}_{n,k|k})^{(1,1)}, \quad \text{and} \quad p^{\dagger}_{n,k|k} = (\mathbf{P}^{\dagger}_{n,k|k})^{(1,1)}, \tag{3.216}$$

where letting $\mathbf{P}_{n,k-1|k}$ be the lower right block of $\mathbf{P}^{+}_{n,k}$, and $\mathbf{P}^{\sharp}_{n,k}$ be the upper right block of $\mathbf{P}^{+}_{n,k}$ produces the required quantities: $\mathbf{P}^{-}_{n,k|k} \triangleq \mathbf{A}_n(\mathbf{P}_{n,k-1|k})\mathbf{A}^{T}_{n}$ and $\mathbf{P}^{\dagger}_{n,k|k} \triangleq \mathbf{P}^{\sharp}_{n,k}\mathbf{A}^{T}_{n}$.

EM via the Dual EKF – Colored Noise

As discussed in Appendix F, the only terms in the cost of Equation 3.213 that depend on the weights $\mathbf{w}$ are the predictions, $\hat{x}^{-}_{k}$, and their associated variances, $p^{-}_{k|k}$ and $p^{\dagger}_{k|k}$. The other terms are functions either of the previous estimates, $\hat{\mathbf{w}}_{k-1}$, or the noise coefficients $\mathbf{w}_n$. Therefore, the observed-error weight filter can be used to produce a generalized M-step by dropping the irrelevant terms, and defining the instantaneous error as:

$$J^{em}_{c,k}(\mathbf{w}) = \frac{(\hat{x}_k - \hat{x}^{-}_{k|k})^2 - 2p^{\dagger}_{k|k} + p^{-}_{k|k}}{\sigma^2_v} \tag{3.217}$$

The terms of $J^{em}_{c,k}$ are computed sequentially with a colored-noise Kalman signal filter as just described. The sequential M-step is computed by a Kalman weight filter. The appropriate observed-error vector and its negative Jacobian matrix are given in Formula 3.28. Letting $\sigma^2_r = \frac{1}{2}\mathbf{I}$, the

$$\boldsymbol{\epsilon}_k = \begin{bmatrix} \sigma_v^{-1}\tilde{\hat{x}}_k \\ \sqrt{-2}\sigma_v^{-1}(p^{\dagger}_{k|k})^{\frac{1}{2}} \\ \sigma_v^{-1}(p^{-}_{k|k})^{\frac{1}{2}} \end{bmatrix} \quad \text{and} \quad \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{\sigma_v}\nabla^{T}_{\mathbf{w}}\tilde{\hat{x}}_k \\ -\frac{\sqrt{-2}(p^{\dagger}_{k|k})^{-\frac{1}{2}}}{2\sigma_v}\nabla^{T}_{\mathbf{w}}p^{\dagger}_{k|k} \\ -\frac{(p^{-}_{k|k})^{-\frac{1}{2}}}{2\sigma_v}\nabla^{T}_{\mathbf{w}}p^{-}_{k|k} \end{bmatrix}$$

Formula 3.28: Colored noise EM cost function. Observed-error terms for dual EKF weight filter.

negative gradient and Hessian of $J^{em}_{c,k}$ are approximated by $\mathbf{H}_{o,k}\sigma_r^{-2}\boldsymbol{\epsilon}_k$ and $\mathbf{H}_{o,k}\sigma_r^{-2}\mathbf{H}^{T}_{o,k}$, as shown in Appendix E.

The gradient $\nabla^{T}_{\mathbf{w}}\tilde{\hat{x}}_{k|k}$ is simply $-\nabla^{T}_{\mathbf{w}}\hat{x}^{-}_{k|k}$, where

$$\nabla_{\mathbf{w}}\hat{x}^{-}_{k|k} = \nabla_{\mathbf{w}} f(\hat{\mathbf{x}}_{k-1|k}, \mathbf{w}), \tag{3.218}$$

and the gradient $\nabla_{\mathbf{w}} p_{k|k}^{-}$ is constructed element-wise from the expression:

$$\mathbf{C}\left(\frac{\partial \mathbf{A}_{k-1|k}}{\partial w^{(i)}}(\mathbf{P}_{x,k-1|k})\mathbf{A}_{k-1|k}^{T} + \mathbf{A}_{k-1|k}(\mathbf{P}_{x,k-1|k})\frac{\partial \mathbf{A}_{x,k-1}^{T}}{\partial w^{(i)}}\right)\mathbf{C}^{T}. \tag{3.219}$$

Finally, the gradient $\nabla_{\mathbf{w}} p_{k|k}^{\dagger}$ is constructed from terms:

$$\mathbf{C}\left(\mathbf{P}_{x,k}^{\sharp}\frac{\partial \mathbf{A}_{k-1|k}^{T}}{\partial w^{(i)}}\right)\mathbf{C}^{T}. \tag{3.220}$$

EM Variance Estimation – Colored Noise

The colored noise EM cost can also be minimized with respect to σ_v^2 and/or $\sigma_{v_n}^2$ to produce estimates of the noise variances. Just as in the white noise case, the instantaneous cost for either variance takes the form:

$$J_{c,k}^{em}(\sigma^2) = \log(2\pi\sigma^2) + \frac{num_k}{\sigma^2}, \tag{3.221}$$

where num_k is one of the relevant numerator term in Equation 3.213, and σ^2 represents either σ_n^2 or σ_v^2. As before, each numerator term is independent of σ^2; the variance filter can be found using the same definitions of $\check{\mathbf{e}}_k$ and $\check{\mathbf{H}}_{o,k}$ as in Formula 3.20 on page 89, but with new definitions given to num_k.

3.5.3 Dual EKF Summary

This has been by far the longest section of the chapter, but rightfully so. The dual EKF incorporates algorithms for signal estimation, weight estimation, and estimation of the unknown noise variances. Signal estimates are obtained using a standard Kalman filter or EKF, or by using an "error-coupled" variation. It was shown how weight estimation and variance estimation can be performed using any of the cost functions derived in Chapter 2 by changing only the definition of a few terms in the algorithm, namely: $\mathbf{e}_k, \mathbf{H}_{o,k}, \check{\mathbf{e}}_k$, and $\check{\mathbf{H}}_{o,k}$.

3.6 Other Issues

In the preceding sections, the joint EKF and dual EKF algorithms were derived for minimizing the cost functions in Chapter 2. However, a few practical considerations warrant further discussion:

1. Most of the dual EKF cost functions require computing the derivatives of a recursive Kalman filter. These computations, sometimes called *sensitivity filtering*, are described in this section.

2. The initial values for the signal and weight state estimates and for the variance estimates are discussed.

3. Application of the dual EKF (or joint EKF) in off-line settings is considered.

4. A criterion for stopping iterative training (in the off-line setting) is described.

5. It is beneficial in some contexts to selectively emphasize and de-emphasize the data used during estimation. Both the "forgetting factor" described earlier, and data-windowing for nonstationary signals are discussed in this context.

6. The computational complexity of the joint EKF and various forms of the dual EKF are discussed and compared.

3.6.1 Computing Derivatives

With Respect to the Weights

In the weight-estimation portion of the dual EKF, computing the negative derivative $\mathbf{H}_{o,k}$ of the observed-error vector $\boldsymbol{\epsilon}_k$ generally requires taking the Jacobian of various quantities in the signal filter. Because the signal filter is a recursive structure, the gradients of quantities such as the state estimate $\hat{\mathbf{x}}_k$, gain $\mathbf{K}_k$, and error covariance $\mathbf{P}_k$ must all be computed recurrently. Taking the derivative of the signal filter equations in Formula 3.10 on page 75 results in the following system of recursive equations:

$$\frac{\partial \hat{\mathbf{x}}_{k+1}^-}{\partial \hat{\mathbf{w}}} = \frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{x}}_k} \frac{\partial \hat{\mathbf{x}}_k}{\partial \hat{\mathbf{w}}} + \frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{w}}_k}, \tag{3.222}$$

$$\frac{\partial \hat{\mathbf{x}}_k}{\partial \hat{\mathbf{w}}} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \frac{\partial \hat{\mathbf{x}}_k^-}{\partial \hat{\mathbf{w}}} + \frac{\partial \mathbf{K}_k}{\partial \hat{\mathbf{w}}} (y_k - \mathbf{C}\hat{\mathbf{x}}_k^-), \tag{3.223}$$

where $\frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{x}}_k}$ and $\frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{w}}_k}$ are evaluated at $\hat{\mathbf{w}}_k$ and contain static linearizations of the neural network. The derivatives in these equations are all derivatives of vectors with respect to vectors, or Jacobian matrices. The last term in Equation 3.223 may be dropped if we assume that the Kalman gain $\mathbf{K}_k$ is independent of $\mathbf{w}$. Although this greatly simplifies the algorithm, accurate computation of the recursive derivatives requires calculating $\frac{\partial \mathbf{K}_k}{\partial \hat{\mathbf{w}}}$ as follows. Denoting the derivative of $\mathbf{K}_k$ with respect to the i^{th} element of $\hat{\mathbf{w}}$ by $\frac{\partial \mathbf{K}_k}{\partial \hat{w}^{(i)}}$ (the i^{th} column of $\frac{\partial \mathbf{K}_k}{\partial \hat{\mathbf{w}}}$) gives:

$$\frac{\partial \mathbf{K}_k}{\partial \hat{w}^{(i)}} = \frac{(\mathbf{I} - \mathbf{K}_k \mathbf{C})}{\mathbf{C}\mathbf{P}_k^- \mathbf{C}^T + \sigma_n^2} \cdot \frac{\partial \mathbf{P}_k^-}{\partial \hat{w}^{(i)}} \mathbf{C}^T, \tag{3.224}$$

where

$$\frac{\partial \mathbf{P}_k^-}{\partial \hat{w}^{(i)}} = \frac{\partial \mathbf{A}_{k-1}}{\partial \hat{w}^{(i)}} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \frac{\partial \mathbf{P}_{k-1}}{\partial \hat{w}^{(i)}} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \mathbf{P}_{k-1} \frac{\partial \mathbf{A}_{k-1}^T}{\partial \hat{w}^{(i)}} \tag{3.225}$$

$$\frac{\partial \mathbf{P}_{k-1}}{\partial \hat{w}^{(i)}} = -\frac{\partial \mathbf{K}_{k-1}}{\partial \hat{w}^{(i)}} \mathbf{C}\mathbf{P}_{k-1}^- + (\mathbf{I} - \mathbf{K}_{k-1}\mathbf{C}) \frac{\partial \mathbf{P}_{k-1}^-}{\partial \hat{w}^{(i)}} \tag{3.226}$$

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,

$$\frac{\partial \mathbf{A}_{k-1}}{\partial \hat{w}^{(i)}} = \frac{\partial^2 \mathbf{F}}{\partial \hat{\mathbf{x}}_{k-1} \partial \hat{w}^{(i)}} + \frac{\partial^2 \mathbf{F}}{(\partial \hat{\mathbf{x}}_{k-1})^2} \frac{\partial \hat{\mathbf{x}}_{k-1}}{\partial \hat{w}^{(i)}}, \tag{3.227}$$

where the first term is the static derivative of $\mathbf{A}_{k-1} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}_{k-1}}$ with $\hat{\mathbf{x}}_{k-1}$ fixed, and the second term includes the recurrent derivative of $\hat{\mathbf{x}}_{k-1}$. The term $\frac{\partial^2 \mathbf{F}}{(\partial \hat{\mathbf{x}}_{k-1})^2}$ actually represents a three dimensional tensor (rather than a matrix). However, because $\mathbf{A}_{k-1}$ takes the special structure shown in Equation 3.24 on page 50, its derivative with respect to $\mathbf{x}$ contains mostly zeros, and is in fact entirely zero for linear models.

The largest computational expense is incurred by the calculation of $\frac{\partial \mathbf{K}_k}{\partial \hat{\mathbf{w}}}$, which requires that $\frac{\partial \mathbf{P}_k^-}{\partial \hat{w}^{(i)}}$ be computed for all $i \in \{1 \ldots \dim(\hat{\mathbf{w}})\}$. Whether this expense is worth the improvement in performance is clearly a design issue, and is investigated in Chapter 4; the recursive derivatives appear to be more critical when the signal is highly nonlinear, or is corrupted by a high level of noise. Various simplifications to the recursive derivatives are possible:

1. Ignore the dependence of $\mathbf{P}_k^-$ and $\mathbf{K}_k$ on $\mathbf{w}$. This would result in the largest savings, and would effectively drop the second term in Equation 3.223 on the preceding page.

2. Ignore the dependence of $\hat{\mathbf{x}}_k$ on $\mathbf{w}$. This drops Equation 3.223 on the previous page altogether, and leaves only the second term in Equation 3.222 on the preceding page. This results in a purely static linearization of the model, and is the simplification made in [61, 87] and investigated in Chapter 4 of this thesis.

With Respect to the Variances

In the variance-estimation filter, the derivative $\breve{\mathbf{H}}_{o,k}$ of the observed-error vector $\breve{\boldsymbol{\epsilon}}_k$ requires taking recursive derivatives similar to those just described for the weight filter. Taking the derivative of the Kalman filter equations with respect to either variance term (represented by σ^2) results in the following system of recursive equations:

$$\frac{\partial \hat{\mathbf{x}}_{k+1}^-}{\partial \sigma^2} = \frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{x}}_k} \frac{\partial \hat{\mathbf{x}}_k}{\partial \sigma^2}, \tag{3.228}$$

$$\frac{\partial \hat{\mathbf{x}}_k}{\partial \sigma^2} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \frac{\partial \hat{\mathbf{x}}_k^-}{\partial \sigma^2} + \frac{\partial \mathbf{K}_k}{\partial \sigma^2} (y_k - \mathbf{C}\hat{\mathbf{x}}_k^-), \tag{3.229}$$

where $\frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{x}}_k}$ is evaluated at $\hat{\mathbf{w}}_k$, and represents a static linearization of the neural network. Note that $\frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{w}}_k} \cdot \frac{\partial \hat{\mathbf{w}}}{\partial \sigma^2}$ does not appear in Equation 3.228, under the assumption that $\frac{\partial \hat{\mathbf{w}}}{\partial \sigma^2} = 0$. The derivatives in the above equations are all derivatives of vectors with respect to a scalar. The last

term in Equation 3.229 may be dropped if we assume that the Kalman gain $\mathbf{K}$ is independent of σ^2. However, for accurate computation of the recursive derivatives, $\frac{\partial \mathbf{K}_k}{\partial \sigma^2}$ may be calculated as follows:

$$\frac{\partial \mathbf{K}_k}{\partial \sigma^2} = \frac{(\mathbf{I} - \mathbf{K}_k \mathbf{C})}{\mathbf{C}\mathbf{P}_k^- \mathbf{C}^T + \sigma_n^2} \cdot \frac{\partial \mathbf{P}_k^-}{\partial \sigma^2} \mathbf{C}^T, \tag{3.230}$$

where

$$\frac{\partial \mathbf{P}_k^-}{\partial \sigma^2} = \frac{\partial \mathbf{A}_{k-1}}{\partial \sigma^2} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \frac{\partial \mathbf{P}_{k-1}}{\partial \sigma^2} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \mathbf{P}_{k-1} \frac{\partial \mathbf{A}_{k-1}}{\partial \sigma^2} \tag{3.231}$$

$$\frac{\partial \mathbf{P}_{k-1}}{\partial \sigma^2} = -\frac{\partial \mathbf{K}_{k-1}}{\partial \sigma^2} \mathbf{C}\mathbf{P}_{k-1}^- + (\mathbf{I} - \mathbf{K}_{k-1}\mathbf{C}) \frac{\partial \mathbf{P}_{k-1}^-}{\partial \sigma^2} \tag{3.232}$$

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

$$\frac{\partial \mathbf{A}_{k-1}}{\partial \sigma^2} = \frac{\partial \mathbf{A}_{k-1}}{\partial \hat{\mathbf{x}}_{k-1}} \frac{\partial \hat{\mathbf{x}}_{k-1}}{\partial \sigma^2} \tag{3.233}$$

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

3.6.2 Initialization

The dual EKF and joint EKF algorithms require initial values $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{w}}_0$ for the signal and weight estimates, and initial values $\hat{\sigma}_{v,0}^2$ and $\hat{\sigma}_{n,0}^2$ for the variances, if they are to be estimated as well.

The additive noise is assumed to have zero mean, and the noisy time-series is generally normalized prior to processing; hence, the signal can also typically be assumed to be zero mean. In the absence of any other information, this assumption is represented by letting $\hat{\mathbf{x}}_0 = 0$.

Reasonable initial values for the weights can be found be training a predictor on some noisy data, as described in Section 1.3.1. Of course, this will result in biased estimates, but a few iterations of training in this way should provide a $\hat{\mathbf{w}}_0$ that is in the right general area of weight space. Initializing the weights in this way can be very beneficial to the training process.

Finding reasonable initial values for the variances is somewhat more complicated. Although in some cases short segments of the noise and/or clean signal might be available for this purpose, such data is not always available. Therefore, some of the heuristic approaches in the literature may be suitable.

Measurement Noise Variance

As mentioned in Section 2.3.2 on page 29, this thesis assumes that the autocorrelation structure of the noise is known within a scalar multiple. Either the noise is white, with a possibly unknown variance σ_n^2, or it is colored, with known AR coefficients $\mathbf{w}_n$ and a possibly unknown process noise variance $\sigma_{v_n}^2$.

Seeing that the noise model $\mathbf{w}_n$ is assumed known in the case of colored noise, it is likely that a reasonable initial estimate of $\sigma^2_{v_n}$ will also be available. For example, if $\mathbf{w}_n$ is known *a priori*, then something might also be known about the range of values taken by $\sigma^2_{v_n}$; alternatively, if $\mathbf{w}_n$ is estimated from some available noise data, then an initial estimate of $\sigma^2_{v_n}$ can be obtained simultaneously.

The white noise assumption corresponds to the specific case when the autocorrelation is an impulse function (*i.e.*, the spectrum is flat); the arguments given in the previous paragraph can therefore also be made in this case. However, because of the analytic simplicity of the white noise case, some heuristic methods for finding an initial estimate of σ_n^2 can also be considered.

An approach developed by Wan and Nelson [87] for nonstationary white noise sources, estimates σ_n^2 as follows. First, consider the noncausal linear estimator of the signal x_k as a function of $2M+1$ noisy data points:

$$\hat{x}_k = \sum_{i=-M}^{M} w^{(i)} y_{k-i} = \mathbf{w}^T \mathbf{y}_{k-M}^{k+M}. \tag{3.234}$$

Note that the optimal weights can be expressed as

$$\begin{aligned} \hat{\mathbf{w}}^* &= \hat{\mathbf{R}}_{yy}^{-1} \hat{\mathbf{r}}_{yx} \\ &= \hat{\mathbf{R}}_{yy}^{-1} (\hat{\mathbf{r}}_{yy} - \sigma_n^2 \mathbf{e}_0), \end{aligned} \tag{3.235}$$

where $\hat{\mathbf{R}}_{yy}$ is the sample autocorrelation of $\mathbf{y}_{k-M}^{k+M}$, $\hat{\mathbf{r}}_{yy}$ is the cross-correlation between $\mathbf{y}_{k-M}^{k+M}$ and y_k, and $\mathbf{e}_0 = [0 \cdots 0\ 1\ 0 \cdots 0]$. The value of σ_n^2 that gives the minimum variance estimate of x_k (minimizes $var(\hat{\mathbf{w}}^{*T} \mathbf{y}_k)$) is:

$$\hat{\sigma}_n^2 = \frac{1}{(\hat{\mathbf{R}}_{yy}^{-1})^{(0,0)}}. \tag{3.236}$$

Appendix H shows that this expression provides an upper bound on σ_n^2. Starting at this upper bound, $\hat{\sigma}_n^2$ is iteratively decreased until $\hat{w}^{(0)} > \hat{w}^{(i)}\ \forall i \neq 0$; this forces the current observation to have the greatest influence on the estimator output, relative to other observations.

Process Noise Variance

To estimate σ_v^2 (assuming an all-pole model for the signal), Lim and Oppenheim [44] used an expression for the inverse Fourier transform of the signal power (which is a function of σ_v^2). An alternative approach is developed in [87] by noting that the process noise variance σ_v^2 can be estimated as the minimum mean squared error of a linear AR predictor on the clean data x_k. Specifically,

$$\sigma_v^2 = \sigma_x^2 - \mathbf{p}_{xx}^T \mathbf{R}_{xx}^{-1} \mathbf{p}_{xx}, \tag{3.237}$$

where $\mathbf{p}_{xx}$ is the cross-correlation between the lagged input vector $\mathbf{x}_{k-1}$ and the current x_k, and $\mathbf{R}_{xx}$ is the autocorrelation of the inputs[8]. Because only the noisy signal y_k with prediction residual $\sigma_x^2 + \sigma_n^2 - \mathbf{p}_{yy}^T \mathbf{R}_{yy}^{-1} \mathbf{p}_{yy}$ is available, σ_v^2 is estimated using:

$$\hat{\sigma}_x^2 = \sigma_y^2 - \hat{\sigma}_n^2, \qquad \hat{\mathbf{p}}_{xx} = \mathbf{p}_{yy} - \hat{\mathbf{p}}_{nn}, \qquad \hat{\mathbf{R}}_{xx} = \mathbf{R}_{yy} - \hat{\mathbf{R}}_{nn} \tag{3.238}$$

in place of the true values in Equation 3.237, giving:

$$\hat{\sigma}_v^2 = (\hat{\sigma}_y^2 - \hat{\sigma}_n^2) - \hat{\mathbf{p}}_{xx}^T \hat{\mathbf{R}}_{xx}^{-1} \hat{\mathbf{p}}_{xx}. \tag{3.239}$$

Note that when n_k is white, the terms in (3.238) simplify because $\hat{\mathbf{p}}_{nn} = 0$ and $\hat{\mathbf{R}}_{nn} = \hat{\sigma}_n^2 I$, where the additive noise variance is estimated as above. Of course, it is possible to obtain negative values for $\hat{\sigma}_v^2$ using the above approach. The estimates should therefore be thresholded from below at some small positive value (*e.g.*, 10^{-4}).

These "ad-hoc" methods alone do not always provide the accuracy required for effective dual estimation. However, they are sufficient for initializing the on-line variance estimation methods described in this thesis.

3.6.3 Iterative Applications

In many contexts, available processing power allows enough time between the arrival of each measurement y_k for the repeated filtering of the previously collected data, in order to better model its dynamics. This might be done with a fixed number N_{win} of data points; at each time step, k, the algorithm is run repeatedly over the window of data: $\{y_t\}_{k-N_{win}}^{k}$. The model is assumed stationary over the window, so the estimates of the parameters ($\hat{\mathbf{w}}_k$, $\hat{\sigma}_{v,k}^2$, and $\hat{\sigma}_{n,k}^2$) and their error covariances ($\mathbf{Q}_k$, $q_{v,k}$ and $q_{n,k}$) at the end of each pass (or *epoch*) are used to initialize their values for the next epoch. The signal state, $\hat{\mathbf{x}}_k$, can of course *not* be used in this way to initialize $\hat{\mathbf{x}}_{k-N_{win}}$ in the next epoch.

By completing several passes over a fixed window, multiple copies of the data in the window are effectively concatenated to create a longer sequence. This better accommodates the convergence time of the algorithm, and reduces the variance in estimation errors; however, the posterior distribution $\rho_{\mathbf{x}_1^k \mathbf{w} | \mathbf{y}_1^k}$ is consequently biased more heavily towards the observed data in the window, and away from whatever the prior distribution is. The procedure has two desirable effects: first, the algorithm converges farther with less data; second, in nonstationary environments the tracking performance increases due to the additional emphasis placed on the most recent data. However,

[8] This is exact, assuming the signal is generated by a linear autoregression.

this can have negative consequences as well, because the chosen cost function may continue to decrease with each epoch, while the performance on future data is being unknowingly degraded, due to the bias on the empirical distribution. This is referred to as *over-training* on the data.

The tendency of the algorithm to over-train will be influenced by three factors: the length of the window, N_{win}; the number of epochs; and the cost function minimized during training. The more data and fewer epochs used, the less likely over-training will be to occur, because the bias in the posterior distribution will be less severe. However, N_{win} must also be chosen in accordance with real-time processing constraints, as well as the time-constants of any nonstationarity in the data. The window length will be at least 1, and constrained from above by either a function of the processor speed, or the number of data points k already collected. The maximum allowable choice of N_{win} is quantified in Formula 3.29, wherein N_{cpu} denotes the maximum number of data points

$$N_{win} = \min\big(1, k, (N_{cpu} - 1), N_{ns}\big) \qquad epochs = \left\lfloor \frac{N_{cpu} - 1}{N_{win}} \right\rfloor \tag{3.240}$$

Formula 3.29: The length of the iteration window, and the number of training epochs.

that can be processed between observations; this number depends on the sampling rate of the data, and the processing speed of the hardware implementation. Furthermore, if any nonstationarity is present in the signal, N_{ns} defines the maximum number of data points over which the signal dynamics are approximately stationary; N_{ns} is highly subjective, and depends on the flexibility of the model structure as well as what is meant be "approximately" stationary. For stationary data, $N_{ns} = \infty$.

The window length is always less than N_{cpu} by at least one, because time must be allowed for processing the newly arrived data point before passing over the old data. Hence, when a large amount of data has been collected (or the processor is slow relative to the sampling rate), then $N_{win} = N_{cpu} - 1$, and only 1 epoch is used. Of course, when there is time for only one data point to be processed, then $N_{win} = 1$, and the algorithm is run in its "on-line" mode. Conversely, when the processor speed allows all of the data to be used, then $N_{win} = k$, and the number of epochs is limited by the processor.

Nonstationary Data

Things are somewhat more complicated when the signal is nonstationary. As mentioned above, the window length is limited from above by N_{ns}. However, it is quite possible that $N_{win} < N_{ns}$ because of processor limitations. A single training epoch will typically not be enough to track

the dynamics of the signal. Instead, a sliding window approach can be used, as described in the next section. Sometimes, even when a sliding window is used, real-time processing is still not a possibility. In this case, the algorithm must be iterated over the window off-line, until it converges on a solution.

Early-Stopping

In cases where the algorithm is iterated many times over a segment of data – such as in the use of short windows on nonstationary signals – the issue arises of when to stop the iteration. If one is concerned with how well the model will generalize its performance with new data, then it is important to avoid over-training. The process of halting iteration before over-training occurs is called *early-stopping* [30], and usually involves evaluating the performance of the algorithm on a set of data not used for training (called the *validation set*). The mean squared error (MSE) on the validation set can be used to determine when over-training begins to occur, and from which epoch the final estimate should be chosen; it is important that this data be different from the *test set*, which is only used after training is complete. The use of a validation set for tasks such as model selection, determining hyperparameter values, and early-stopping, is referred to in the statistics and machine learning communities as *cross-validation* [30].

Note that over-training is a problem even in dual estimation applications such as speech-enhancement, wherein generalization to future data is not a concern. Because the true signal is not available, dual estimation is essentially an unsupervised learning problem; it is possible that the estimation error between $\hat{x}_k$ and the true signal will begin to increase after several epochs, even as the cost function continues to decrease.

Using a validation set in time-series applications is not always straight-forward. For example, if the signal is highly nonstationary then it must be windowed, and the training window will be quite short to begin with. Hence, reserving a block of data from the end of this window is likely to hurt performance both because the amount of training data is reduced, and because the model is no longer trained on the data that is most relevant to the next portion of the signal.

One approach is to first train on a representative window while using a validation set, to determine a reasonable number of training epochs. The algorithm is subsequently run without a validation set, but is stopped after this predetermined number of epochs. This method of early stopping is used successfully in the speech enhancement experiments described in Chapter 5.

Another alternative is to randomly sample the validation set from within the training window; this reduces the amount of training data, but avoids the difficulty of removing data only from the end of the training set. For the dual EKF, withholding randomly chosen measurements from

the training set is very much like dealing with missing observations; something at which Kalman filters are quite adept. Essentially, during time-steps when the data y_k is missing or withheld, no measurement update occurs in either the signal filter or weight filter. Instead, the predictions ($\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{w}}_k^-$) and their covariances are substituted for the estimates ($\hat{\mathbf{x}}_k$ and $\hat{\mathbf{w}}_k$) and their covariances. Similar substitutions are made for the recurrent derivative computations. For the purposes of cross-validation, the prediction error ($y_k - \hat{x}_k^-$) is computed for the withheld points, and the validation MSE is tracked from one epoch to the next. The parameter and signal estimates for the epoch with the lowest validation MSE are saved as the final solution.

Trajectory Learning

Choosing validation points from within the training set has an additional effect on the dual estimation process that can improve performance in some situations. Withholding an observation causes the signal prediction $\hat{\mathbf{x}}_k^-$ to be used as the input to the model at the next time step (to generate output $\hat{\mathbf{x}}_{k+1}^-$); thus, the model is effectively being trained as an iterated, or ***multistep predictor***. When the next observation arrives, the recurrent derivatives allow adjustment of the model to reduce the error in the iterated prediction; this puts additional constraints on the model, and has been shown to improve the predictive power of neural network time-series models. The approach has been referred to in the literature as *trajectory learning* [31], and is also related to the *compromise method* [94].

The trajectory learning technique can be applied independently of cross-validation, although in the dual Kalman filter the same mechanism (of handling missing observations) is used for either trajectory learning or cross-validation. The only difference is that for early-stopping, the same hold-out set must be used across all epochs, whereas for trajectory learning a different set of points can be withheld during each pass through the data.

3.6.4 Data Weighting

In some contexts, the dual estimation process can benefit from rescaling data in various ways. In this section, the importance of the forgetting factor, data windowing, and data normalization are described.

Forgetting Factor

As described on page 71, the sequence of signal-state estimates $\{\hat{\mathbf{x}}_k\}_{k=1}^{\infty}$ is generated using the sequence of costs $\{J(\mathbf{x}_1^k, \hat{\mathbf{w}}_k)\}_{k=1}^{\infty}$. Clearly, the signal estimates $\hat{\mathbf{x}}_k$ will improve as the weight estimates $\hat{\mathbf{w}}_k$ used to generate them improve. The sequence of weight estimates $\{\hat{\mathbf{w}}_k\}_{k=1}^{\infty}$, meanwhile,

is based on the sequence of signal estimates.

Hence, because the signal estimates at early times are less accurate than the later ones, it stands to reason that their influence on the weight estimation process should be scaled down. As described in Section 3.3, the amount of past data used to estimate the weights can be controlled through the use of a forgetting factor $\lambda < 1$, or equivalently, by appropriate use of a process noise term $\mathbf{u}_k$ in the state-space representation for the weights.

Specifically, defining the process noise covariance $\mathbf{U}_k$ as in Equation 3.63 effectively places an exponentially decaying window (see Figure 3.3 on page 59) on the data used for weight estimation, so that $\hat{\mathbf{w}}_k$ depends more heavily on recent signal estimates than older ones. The appropriate time constant of this exponential window depends on the complexity of the model: enough data should be left inside the window to accurately estimate the weights.

The forgetting factor introduces a certain amount of flexibility into the parameter estimation filters, making them more responsive to new data, and improved signal estimates. Note, however, that although the forgetting factor is implemented in the Kalman weight filter through a process noise covariance, $\mathbf{U}_k$, this does not imply that the underlying system is actually time-varying.

For off-line applications involving a finite amount of data, the time-constant might be chosen to be somewhat larger than N, to ensure that all of the available data is used. In these applications the algorithm is usually run over the data repeatedly, so that $\lambda < 1$ causes the earlier iterations to be "forgotten."

The effect of λ on dual Kalman filter convergence is shown experimentally in Chapter 4. As mentioned previously, other schemes for defining $\mathbf{U}_k$ - such as letting it be a constant diagonal matrix, or annealing it over time - can also be considered, but are not investigated in this thesis.

Windowing

For nonstationary signals, such as speech, the optimal weight vector $\mathbf{w}$ is time-varying. As described in the previous section, the change in the dynamics generally cannot be assumed to be slower than the tracking time-constant of the learning algorithm, so special measures are required. When the computational demands are too great to process a window $\{y_t\}_{k-N_s}^{k}$ at every time-step k, an alternative is to divide the data into approximately stationary, overlapping windows.

These data-windows are filtered separately to produce signal estimates $\{\hat{x}_k\}_1^N$, and then are recombined to produce the complete enhanced signal. Because the windows are typically short, the dual estimation algorithm (either joint EKF or dual EKF) should be run repeatedly over the window until convergence is achieved. Hence, windowing usually entails an off-line mode of processing.

If the windows are not overlapped, then discontinuities, or "edge-effects" will be evident at the window boundaries. Typically, the windows should be overlapped, and ***shaped*** after processing so that they can be recombined by simple addition. This can be accomplished, for example, with a normalized Hamming window of the form:

$$c_k = \frac{1}{gain}\left(0.54 - 0.46 \cdot \cos\frac{2\pi k + \pi}{N_{win}}\right), \tag{3.241}$$

where N_{win} is the length of each window, and *gain* is the sum of all overlapping window values at a particular data point. Division by this gain term scales the windows so that they sum to 1, as shown in Figure 3.7. Note that the noisy data are *not* shaped before filtering; this would disrupt

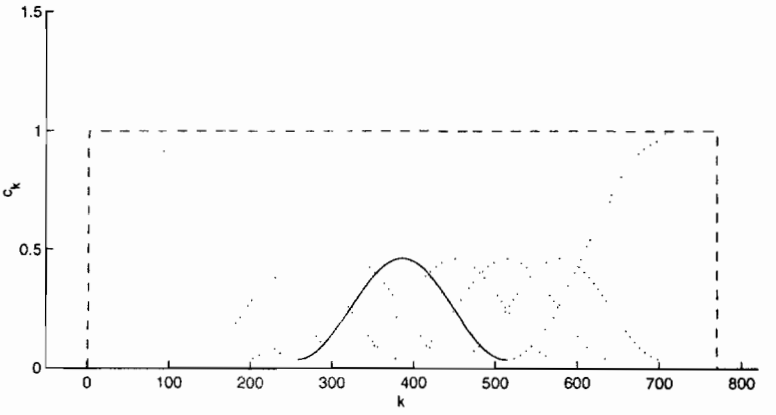


Figure 3.7: Normalized Hamming windows can be used to scale the filtered data in each window so that the windows can be added together without affecting the overall gain. Special shapes are used for the first and last windows to avoid attenuation near the endpoints. Here, the window length is 256, with a shift of 64 points between windows.

the dynamics of the underlying signal. Rather, the Hamming window is applied to the filtered signal $\hat{\mathbf{x}}_1^{N_{win}}$.

For the Hamming window, the contribution of each window to the overall signal estimate is focused at the center of the window. Therefore, the estimation of the data near the center of each window is more critical than the estimation of the data towards the periphery. To reflect this preference in the model estimation procedure, the weights can be estimated in such a way as to emphasize the data according to the shaping window. In the dual EKF, this requires adjusting the weight filter equations slightly; the Kalman gains in Formulae 3.10 and 3.21 are replaced by:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^- \mathbf{H}_{o,k}^T (\mathbf{H}_{o,k}\mathbf{Q}_k^-\mathbf{H}_{o,k}^T + c_k^{-1}\sigma_r^2)^{-1}. \tag{3.242}$$

An analogous change is also made to the variance estimation filter of Formula 3.13 on page 78 by replacing σ_r^{-2} with $c_k\sigma_r^{-2}$. In the linear model case, the above change effectively implements a recursive *weighted least squares* algorithm [6], where the weighting matrix has the window coefficients

c_k along its diagonal, and zeros elsewhere.

Normalization

A related issue is that of pre-scaling the data so that it falls in a reasonable range for numerical accuracy on a finite-precision computer. Most machines do poorly at representing either extremely large, or extremely small numbers, so it is important that the signal and weights take on values that give reasonable precision.

Furthermore, normalizing the data facilitates the use of default initialization values across different data sets. For example, initial covariances $\mathbf{P}_0$ and $\mathbf{Q}_0$ must be set for both the signal and weight estimates, and these matrices should ultimately depend on the expected scale of the data. Hence, normalizing the data reduces this data-dependence.

Normalization requires both subtracting the mean from the data, and scaling by a normalization factor, which can be either the standard deviation or the maximum absolute value of the noisy time-series. For on-line applications, the data can be normalized as it arrives using a preset estimate of the mean and normalization factor. In off-line contexts, these values can be determined from the entire data set.

A possible complication arises when either of the variances σ_v^2, σ_n^2 (or $\sigma_{v_n}^2$) is known *a priori*. In this case, its value must be scaled by the square of the normalizing factor. While this scaling is correct for the measurement noise statistics, it is only correct for the process noise variance σ_v^2 in the case of a linear model $f(\cdot)$; otherwise, it represents an approximation.

3.6.5 Computational Expense

In the following, the number of floating point operations required at each time step by the dual EKF and joint EKF are roughly calculated to show how the algorithms compare from the perspective of computational expense. Although the calculations are made for the white noise case, the expense for colored noise can be approximately found by substituting $(M + M_n)$ for M.

Dual EKF

As suggested in Section 3.6.1, computation of $\mathbf{H}_{o,k}$ - which involves the recursive derivatives of the signal estimates and covariances with respect to the weights - accounts for a large part of the computational cost of the dual EKF algorithm. Table 3.2 lists the approximate number of floating point operations for each part of the algorithm. This is often referred to as the *order* of the computational expense, and is denoted by $O(\cdot)$. The sparse structure of matrices (such as $\mathbf{A}_k$

Table 3.2: The order of computational expense for various equations in the dual EKF.

	Equation	Term	Order of Expense	Explanation
Formula 3.10	3.129-3.133	$\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-, \mathbf{K}_k, \hat{\mathbf{x}}_k, \mathbf{P}_k$	$6M^2 + 3M + 2M_{\mathbf{w}}$	signal filter
	3.127-3.128	$\hat{\mathbf{w}}_k^-, \mathbf{Q}_k^-$	$M_{\mathbf{w}}^2$	weight filter
	3.134	$\mathbf{K}_k^{\mathbf{w}}$	$4M_{\mathbf{w}}^2 M_o + 4M_o^2 M_{\mathbf{w}} + M_o^3$	"
	3.135	$\hat{\mathbf{w}}_k$	$M_{\mathbf{w}} + 2M_o M_{\mathbf{w}}$	"
	3.136	$\mathbf{Q}_k$	$2M_{\mathbf{w}}^2 M_o + M_{\mathbf{w}}^2$	"
Recurrent Derivatives	3.222	$\frac{\partial \hat{\mathbf{x}}_{k+1}^-}{\partial \hat{\mathbf{w}}}$	$4M_{\mathbf{w}}$ $M_{\mathbf{w}}$ $(2M^2 + M)M_{\mathbf{w}}$	$O(\frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{x}}_k})$ $O(\frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{w}}_k})$ O(matrix mult.)
	3.223	$\frac{\partial \hat{\mathbf{x}}_k}{\partial \hat{\mathbf{w}}}$	$3MM_{\mathbf{w}} + M^2$	O(matrix mult.)
	3.224	$\frac{\partial \mathbf{K}_k}{\partial \hat{w}^{(i)}}\ \forall i$	$2M^2 M_{\mathbf{w}} + MM_{\mathbf{w}}$	O(matrix mult.)
	3.225	$\frac{\partial \mathbf{P}_k^-}{\partial \hat{w}^{(i)}}\ \forall i$	$8M^2 M_{\mathbf{w}} + 4M^2$	O(matrix mult.)
	3.226	$\frac{\partial \mathbf{P}_{k-1}}{\partial \hat{w}^{(i)}}\ \forall i$	$M^2 M_{\mathbf{w}}$	O(matrix mult.)
	3.227	$\frac{\partial \mathbf{A}_{k-1}}{\partial \hat{w}^{(i)}}\ \forall i$	$2MM_{\mathbf{w}} + M_{\mathbf{w}}$ $2MM_{\mathbf{w}}$ $2M^2 M_{\mathbf{w}} + MM_{\mathbf{w}}$	$O(\frac{\partial^2 \mathbf{F}}{\partial \hat{\mathbf{x}}_{k-1} \partial \hat{w}^{(i)}})$ $O(\frac{\partial^2 \mathbf{F}}{(\partial \hat{\mathbf{x}}_{k-1})^2})$ O(mult. & add)

and $\mathbf{C}_k$) is taken into account as much as possible, so the numbers in the table represent a fairly efficient implementation.

Combining the costs of Equations 3.222-3.227, computation of the recursive derivatives required by the dual EKF is $O(15M^2 M_{\mathbf{w}} + 10MM_{\mathbf{w}} + 6M_{\mathbf{w}} + 5M^2)$, while use of the static derivatives alone is only $O(5M_{\mathbf{w}})$. Meanwhile, the Kalman signal filter of Formula 3.10 on page 75 requires $O(6M^2 + 3M + 2M_{\mathbf{w}})$ computations, and the weight filter is $O(M_o^3 + 4M_{\mathbf{w}} M_o^2 + M_{\mathbf{w}} + 2M_{\mathbf{w}} M_o + 6M_{\mathbf{w}}^2 M_o + 2M_{\mathbf{w}}^2)$, where M_o is the dimension of the observation vector, $\boldsymbol{\epsilon}_k$.

Joint EKF

The joint EKF does not use the recursive derivatives required by the dual EKF, so its expense is limited to that of an EKF used to filter the joint state-space equations. Including all derivative computations, this filter requires $O(6M_{\mathbf{z}}^2 + 3M_{\mathbf{z}} + 2M_{\mathbf{w}}^2 + 7M_{\mathbf{w}})$ computations, where $M_{\mathbf{z}}$ is the dimension of the joint state. In the white noise case $M_{\mathbf{z}} = M + M_{\mathbf{w}}$, so the joint EKF is $O(8M_{\mathbf{w}}^2 + 12MM_{\mathbf{w}} + 10M_{\mathbf{w}} + 6M^2 + 3M)$

Table 3.3: Coefficients for the order of computational expense of one time-step for the joint EKF and dual EKF, when written as polynomials in the signal state dimension M, and the number of weights, $M_{\mathbf{w}}$. Static derivative forms of the dual EKF are indicated by a prime (').

Algorithm	$M_{\mathbf{w}}^2$	$M^2 M_{\mathbf{w}}$	$M\ M_{\mathbf{w}}$	$M_{\mathbf{w}}$	M^2	M	1
joint EKF	8	0	12	10	6	3	1
dual EKF $J^{pe}(\mathbf{w})$	8	15	10	15	9	5	1
dual EKF $J^{ml}(\mathbf{w})$	14	15	10	29	9	5	8
dual EKF $J^{j}(\mathbf{w})$	14	15	10	29	9	5	8
dual EKF $J^{em}(\mathbf{w})$	20	0	0	46	4	5	27
dual EKF $J^{ec}(\mathbf{w})$	26	15	10	81	9	5	64
dual EKF' $J^{pe}(\mathbf{w})$	8	0	0	10	4	5	1
dual EKF' $J^{ml}(\mathbf{w})$	14	0	0	24	4	5	8
dual EKF' $J^{j}(\mathbf{w})$	14	0	0	24	4	5	8
dual EKF' $J^{ec}(\mathbf{w})$	26	0	0	76	4	5	64

Comparison

To facilitate comparison of the two algorithms, Table 3.3 shows the coefficients of terms involving M and $M_{\mathbf{w}}$ for the joint EKF, as well as the five different cost functions of the dual EKF. Each cost has a different number of observations, M_o, which gives them different overall costs.

The order of expense for $M = 10$, at various values of $M_{\mathbf{w}}$ is shown in Figure 3.8. Clearly, the joint EKF conveys a significant computational advantage, due to its lack of recursive derivative computation. The dual EKF with the EM cost does not require recurrent derivatives, and so its expense is less than that of other dual EKF algorithms. If static derivatives are used in the other dual EKF costs, their expense is reduced considerably, and the cost of the prediction-error algorithm is less than that of the joint EKF.

However, it should be noted that the above costs involve only floating point multiplies and adds, and assume that the algorithms are coded with an eye for efficiency. For the MATLAB code used to generate the experimental results in the next chapter, the joint EKF and dual EKF are quite comparable in terms of overall execution time.

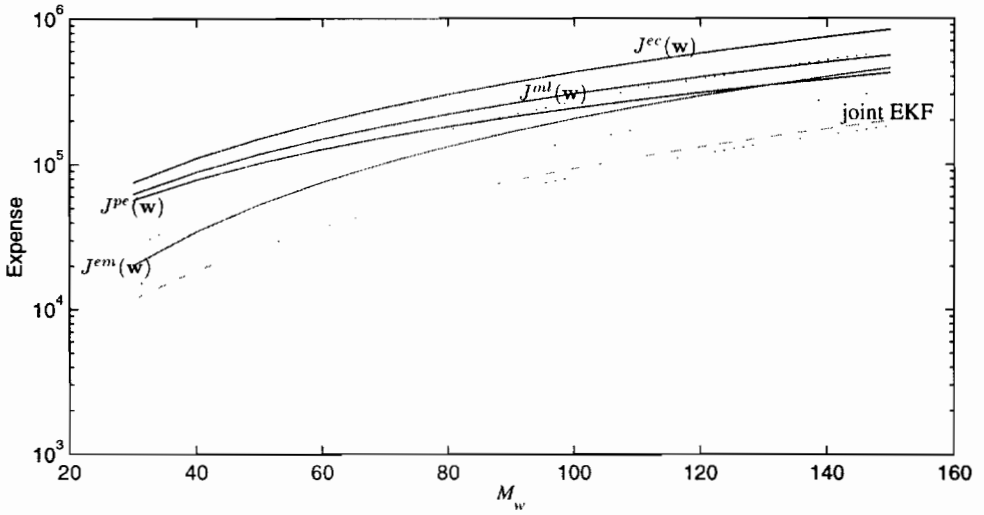


Figure 3.8: Floating point operations required by the joint EKF and dual EKF algorithms, as a function of the dimension of $\mathbf{w}$, with M fixed at 10. The solid curves represent dual EKF algorithms with recurrent derivatives. Note that the expense of $J^j(\mathbf{w})$ is the same as $J^{ml}(\mathbf{w})$. The joint EKF expense is shown with a dashed line. Nonrecurrent approximations are represented by dotted lines without labels: $J^{pe}(\mathbf{w})$ (lower), $J^{ml}(\mathbf{w})$ and $J^j(\mathbf{w})$ (middle), and $J^{ec}(\mathbf{w})$ (upper).

Chapter 4

Comparative Experiments

4.1 Overview

The previous chapters of this thesis describe several different dual estimation algorithms and cost functions. The current chapter has several goals: to determine appropriate settings for various algorithmic parameters; to compare the utility of the different cost functions within the dual EKF framework; and to compare the performance of the dual EKF algorithm with that of the joint EKF and other algorithms. These goals are approached through a series of controlled experiments, in which the clean signal, true model structure, and true model parameters are all known beforehand. This information is necessary for computing objective performance criteria, such as the mean squared error (MSE)

For example, consider the time-series data depicted in the top part of Figure 4.1. The solid curve was generated by a neural network function $f(\cdot)$, driven by white Gaussian process noise with variance $\sigma_v^2 = .36$. Colored noise was then generated by a known linear autoregressive model and added to the signal to produce the noisy measurements shown by the small dots. For clarity, only the last 200 points of the 20,000 point time-series are shown. The bottom part of the figure shows the same signal, but with the dual EKF estimates superimposed as small dots. If the clean signal was not known, we could not see that the dual EKF estimates are closer to the signal than the noisy measurements, nor compute the normalized MSE before (.5016) and after (.1263) processing. Performance measures such as MSE play a crucial role in comparing the performance of the various cost functions, and deciding how to initialize the algorithm or choose a forgetting factor.

The comparative experiments in this chapter are aimed at resolving the question of which cost function or algorithm to use when confronted with a particular noisy signal. In addition to deciding on a cost function, other design issues must also be determined. In particular, the initial values of the covariance matrices for the signal, weight, and variance filters must be chosen, as must a value

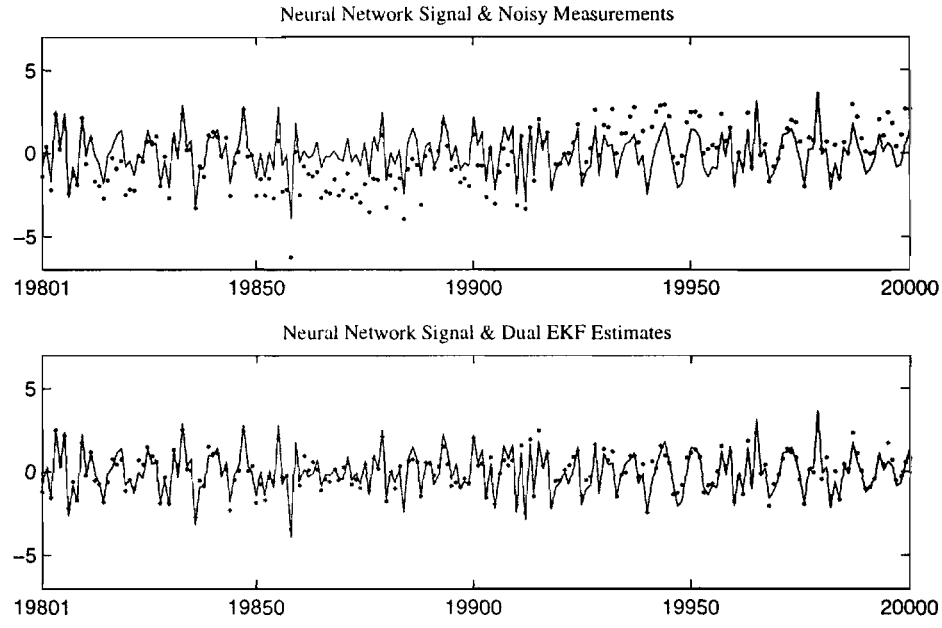


Figure 4.1: Estimation of a nonlinear time-series by the dual EKF. The true signal is shown by the solid curve. Noisy inputs to the dual EKF are shown by the dots in the top plot, and the signal estimates are shown with dots in the bottom plot.

of the forgetting factor, λ. Because the best set of choices may depend on the particular type of signal and noise (*e.g.*, linear or nonlinear, white or colored noise), and the given signal-to-noise ratio (SNR), an effort is made to isolate these factors.

The situation is further complicated by the amount of *a priori* knowledge assumed to be available. That is, the results might depend on whether or not the noise variances are known, and to what degree of certainty the required complexity of the model structure is known. A lack of knowledge about the true noise statistics or model structure represents a potential source of additional error, to which some cost functions are likely to exhibit better robustness than others.

Each of the above design parameters or variables represents a dimension in what is clearly a very large search space. Unfortunately, searching this space exhaustively for the best set of design choices is prohibitively expensive. However, some design choices can be expected to be less tightly coupled with other choices. For example, the effect of the initial covariance matrices $\mathbf{P}_0$ and $\mathbf{Q}_0$ on performance should be somewhat independent of the cost function being used. By optimizing these values first and holding them constant, the dimensionality of the search space is reduced.

Hence, a sequence of experiments is performed. In the first experiment, reasonable choices for $\mathbf{P}_0$ and $\mathbf{Q}_0$ are found, assuming the noise variances σ_v^2 and σ_n^2 are known. A few different

cost functions are used to confirm the assumption of independence. Second, in an experiment using known weights, $\mathbf{w}$, the different variance estimation cost functions are explored, along with values of the initial error variances, $q_{v,0}$ and $q_{n,0}$. A few different noise types are employed, and the experiment is repeated for the various cases wherein one or both of the variances are unknown. The third experiment looks at the effect of the forgetting factor, λ, in the presence of both stationary and nonstationary noise. Fourth, the relative performance of the various dual EKF weight estimation costs are determined using the values of $\mathbf{P}_0$, $\mathbf{Q}_0$, $q_{v,0}$, and $q_{n,0}$, and λ determined in the earlier experiments. The use of static derivatives instead of recursive ones is evaluated in the fifth experiment, and in the sixth experiment, the dual EKF and joint EKF algorithms are compared. Some final experiments are used to evaluate the robustness of the algorithms to incorrect choices of model structure, and the use of the algorithms in iterative estimation settings.

4.2 Experimental Framework

Before presenting the results, however, the experimental framework must be described. The following pages provide an outline of the performance criteria, method of analysis, and various data sets used in the experiments.

4.2.1 Performance Criteria

Comparing different estimation methods requires a means of evaluating the performance of the dual EKF and other algorithms. This choice of a performance criterion ultimately corresponds to a particular definition of the loss function $L(\cdot)$, which may be a function of the signal estimation error, and errors in the parameter estimates (recall the discussion of Bayes Risk in Section 2.2.2 on page 20).

A particularly useful class of functions are the sum of squared errors (SSE), where the sum might be weighted differently over different components of the weight and signal errors, and over different times, k. Although choosing an appropriate loss-function is typically a problem-specific task, some degree of generality is afforded by the fact that a broad class of loss functions all correspond to the same Bayes estimates, so long as $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$ is unimodal and symmetric. In fact, under these conditions, the Bayes estimate for the SSE loss function is also the desired MAP solution described in Chapter 2.

For certain applications (*e.g.*, speech enhancement), a SSE loss function is not the most appropriate choice. In the case of speech, this is because the signal's phase information is not a strong cue in human speech perception, and because the human auditory system is differently sensitive to

different frequencies of sound. These and other factors are ignored by a simple SSE loss function. On the other hand, a satisfactory objective measure of speech quality has yet to be developed. Although, the speech enhancement results reported later in this section are presented in terms of weighted spectral slope, segmental SNR, and several other perceptual metrics, none of these measures is an accurate indicator of human perception of speech quality. Because the emphasis of this thesis is on the theoretical relationship between various cost functions in the MAP context, and not on a specific signal type or application, the SSE loss function is a reasonable choice for evaluating the performance of most signals.

Normalized Mean Squared Errors

To compute the SSE, the squared errors in the signal and weights are summed over a finite period of time. This value can then be divided by the length of time (number of data points) to produce the time-averaged, or mean, squared error (MSE). Furthermore, the MSE can be normalized by the variance of the clean signal (or true weights) to produce numbers close to the range $[0, 1]$. The formula for the normalized mean squared error (NMSE) is:

$$\text{NMSE} = \Big(\sum_{k=k_1}^{k_2} true_k^2\Big)^{-1} \sum_{k=k_1}^{k_2} (true_k - estimate_k)^2, \tag{4.1}$$

where the number of data points $(k_2 - k_1 + 1)$ cancels out of the expression. NMSE values are typically less than 1, which would correspond to an estimate of all zeros for $k \in [k_1, k_2]$. However, values larger than one are possible for exceedingly poor results.

To keep the analysis as general as possible, the estimation error and parameter errors are considered separately. The estimation error at time k is simply $\tilde{x}_k = (x_k - \hat{x}_k)$, where $\hat{x}_k$ is the estimate produced by the algorithm in question. The variance-parameter errors are $\tilde{\sigma^2}_{v,k} = (\sigma_v^2 - \hat{\sigma}_{v,k}^2)$ and $\tilde{\sigma^2}_{n,k} = (\sigma_n^2 - \hat{\sigma}_{n,k}^2)$. When the model is linear, one can also consider the weight-parameter error $\tilde{\mathbf{w}}_k = \sum_i (w^{(i)} - \hat{w}_k^{(i)})$; however, the weight error is not a meaningful quantity for neural network models because of the non-uniqueness of solutions. The prediction error can also be considered, defined as $\tilde{x}_k^- = (x_k - \hat{x}_k^-)$, and can be viewed as a function of the estimation error and parameter errors[1].

If the algorithms are evaluated on a time-series of length N, summing the instantaneous squared errors over $k \in [1, N]$ produces a number that represents the overall quality of the estimation procedure. This can be done separately for estimation, prediction, and parameter errors.

[1] An alternative definition of prediction error, $(y_k - \hat{x}_k^-)$, differs from the above definition by inclusion of the measurement noise n_k.

Computing the overall NMSE facilitates the comparison of algorithms by providing a scalar measure of quality.

However, we are typically interested not only in the overall sum of these squared errors, but also in their values as functions in time. Some information about the time-dependence of the errors can be obtained by summing over shorter segments of the result. For example, summing over the first 100 time steps gives a picture of an algorithm's performance at small times, whereas summing over the last 1000 time steps can show the performance of the algorithm near convergence.

Error Traces

Ultimately, however, a time-trajectory of squared errors conveys the most information about the convergence properties of the algorithm. For example, the squared error in the estimate $\hat{\sigma}^2_{n,k}$ of the measurement noise variance will ideally appear as a monotonically decreasing curve when plotted against time. Information about the convergence rate and asymptotic value can be readily discerned from a plot of the ensemble statistics of these curves.

However, a similar plot of the signal estimation error will not be so easy to evaluate; the squared error will generally appear quite noisy and will vary greatly with the instantaneous dynamics of the underlying signal. Even when the model parameters, $\mathbf{w}$, σ_v^2, and σ_n^2 are known exactly, a plot of the estimation error from an extended Kalman signal appears highly noisy. The situation is improved somewhat by plotting the short-term MSEs, computed every 50 points over a 500 point window (as shown in the middle plot of Figure 4.2). These smoothed plots, referred to in this thesis as *MSE profiles*, are easier to interpret than the raw MSEs, but the convergence properties are still not readily discernible. To aid in the interpretation of estimation error plots, it can be useful to compare the MSE profile of the dual estimation algorithm (unknown model) with the profile of the ideal signal estimation result (known model). The difference in these MSE profiles more closely resembles a monotonically decreasing curve, as in the bottom plot of Figure 4.2.

4.2.2 Statistical Analysis

When comparing two methods, one must determine whether the perceived difference in their performance is statistically significant. This can be done by repeating the experiment several times, and looking at the ensemble statistics of the loss function taken over different realizations of the data. If the difference between the means of two methods is much greater than the variation seen for an individual method, then the difference is significant.

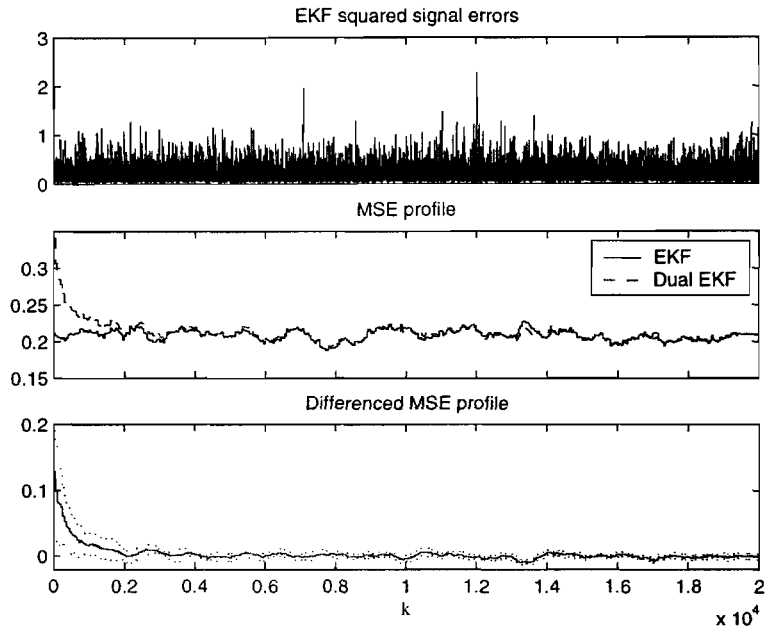


Figure 4.2: The ensemble average (10 repetitions) of the squared signal error $(x_k - \hat{x}_k)^2$ is too noisy to interpret (top plot). Computing the MSEs over a sliding 500-point window produces an "MSE profile" (middle plot). The convergence behavior of a dual estimation algorithm can be viewed by subtracting its profile from that of the EKF, to produced a "differenced MSE profile" (bottom). The ensemble mean and standard deviation are shown by solid and dotted lines, respectively.

Sampling Distribution

For a particular algorithmic treatment, the value of the loss function $L(\cdot)$ generally depends on the underlying signal $\{x_k\}_1^N$, weights $\mathbf{w}$, and data $\{y_k\}_1^N$. As described on page 20, these quantities can be viewed as random samples drawn according to $\rho_{\mathbf{x}_1^N \mathbf{w} \mathbf{y}_1^N}$. The loss $L(\cdot)$ is therefore a random variable whose probability distribution depends on this joint density function. A particular algorithm can be evaluated by estimating both the mean of $L(\cdot)$ (*i.e.*, the Bayes risk):

$$E_{\mathbf{Y}}\Big[E_{\mathbf{X}\mathbf{W}|\mathbf{Y}}\big[L(\{x_k - \hat{x}_k\}_1^N, \mathbf{w} - \hat{\mathbf{w}})\big|\{y_k\}_1^N\big]\Big], \tag{4.2}$$

and the variance of $L(\cdot)$ empirically, using samples drawn from $\rho_{\mathbf{x}_1^N \mathbf{w} \mathbf{y}_1^N}$.

For example, $\mathbf{w}$ might be sampled according to a multivariate Gaussian distribution around a particular mean. Next, $\mathbf{x}_1^N$ can be drawn from $\rho_{\mathbf{x}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{x}_0}$, by generating the initial condition $\mathbf{x}_0$ and process noise as Gaussian random variables. Finally, the noisy data $\mathbf{y}_1^N$ can be produced by sampling from, for example, $\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N \mathbf{w}} = \mathcal{N}(\mathbf{x}_1^N, \sigma_n^2 \mathbf{I})$, in the white noise case. By repeating this sampling procedure, the statistics of $L(\cdot)$ for a particular algorithm will emerge.

However, varying the weights in this manner can produce widely varying signal dynamics, which would dominate the resultant variation in the loss function. The variance of $L(\cdot)$ for a

particular algorithm would then be quite large, thereby obscuring the difference between the Bayes risk of two algorithms. Ultimately, statistical techniques can be applied to mitigate the problem, but a large number of samples are required for this, resulting in great computational expense. Furthermore, the average error trajectories would be much less meaningful.

A better use of computation time can be made by taking a different approach. Consider the idea that the various algorithms are likely to behave in the same way *relative to one another* at different points in the weight space. In other words, for a given parameterization of the underlying system, $f(\cdot, \mathbf{w})$, the *ranking* of the algorithms will not depend on the true underlying weights, $\mathbf{w}$. This means that the conclusions made about the relative performance of algorithms for one system can be generalized to other systems of similar complexity. This assumption can be tested, but it is clearly a highly desirable situation; if it is not true, there is little value in running the comparative experiments in the first place.

Under the above assumption, the evaluations can be performed using a fixed weight vector $\mathbf{w}$. In other words, $\rho_{\mathbf{w}}$ is taken to be singular, producing a fixed value. The experiment can be performed for a few different values of $\mathbf{w}$ to validate the assumption, but this is not as troublesome as sampling across the space of all possible weight values.

With the weights fixed, sampling across the signal space (varying $\{x_k\}_1^N$) amounts to varying the initial conditions $\mathbf{x}_0$, and the particular realization of the process noise $\{v_k\}_1^N$. However, if the signal is *ergodic*, the statistics of the signal space computed across *time* will tend toward the ensemble statistics at large values of time k. In other words, taking the ensemble average of the error across different realizations of the signal $\mathbf{x}_1^N$ should be equivalent to computing the time-average.

Thus, a more efficient use of computational resources is to sample only from $\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N\mathbf{w}}$ (drawing different realizations of the measurement noise $\mathbf{n}_1^N$) while keeping $\mathbf{w}$ and $\mathbf{x}_1^N$ fixed. This can be viewed as estimating the outer expectation in Equation 4.2 alone, while computing the inner expectation with a singular density function $\rho_{\mathbf{x}_1^N\mathbf{w}}$, located at the known values $\{x_k\}_1^N$ and $\mathbf{w}$.

T Test

With $\rho_{\mathbf{x}_1^N\mathbf{w}\mathbf{y}_1^N}$ in place, and assuming the loss function $L(\cdot)$ is chosen, the task remains to evaluate the various algorithmic design choices in a set A. If the loss under a particular algorithmic treatment, $a \in A$, is denoted by the random variable $L_a(\mathbf{x}_1^N, \mathbf{w}, \mathbf{y}_1^N)$, then the ultimate goal is to determine if $E_{\mathbf{Y}}[L_a] < E_{\mathbf{Y}}[L_b]$ for all pairs $(a \neq b) \in A$.

This can be accomplished with a *paired sample t test*, as explained in Appendix I. This test looks at the difference between two treatments, and produces a statistic, called the p-value,

that reflects how likely the difference is to be zero-mean. When the p-value is close to zero, the difference between the treatments is significant. If the p-value is large, then the data do not support a difference between the treatments.

Boxplots

One weakness of the t test comes from its assumption of Gaussianity. Suppose that treatment b is significantly worse on average than treatment a, and moreover, is prone to occasional divergence. The problem occurs when b diverges on the r^{th} repetition, causing a difference, $d_{a,b}^{[r]} \triangleq (L_a - L_b)$, much larger than the average difference, $D_{a,b} = \frac{1}{R}\sum_r d_{a,b}^{[r]}$, across R repetitions; this inflates the sample variance of the differences, s_D^2. The $t_{a,b}$ statistic makes no distinction between values of $d_{a,b}^{[r]}$ larger than the mean and values smaller than the mean, so even though no differences close to zero are observed, $t_{a,b}$ is decreased, and the p-value becomes large. This shortcoming can be compensated for by viewing a boxplot (*e.g.*, see Figure 4.10 on page 133). This plot shows the median, and the upper and lower quartiles of the data with horizontal lines. Vertical "whiskers" show the range of data within a length of 1.5 times the interquartile range, both above and below the inner quartiles. Outliers are points outside of the whiskers, and are plotted separately with a "+" symbol. When all the data are in-range (no outliers), this is indicated by a small dot at the bottom of the lower whisker.

Typically, then, the experimental results in this section are interpreted by making boxplots of the various treatments to show the range of results obtained. Next, the algorithmic treatment a with the smallest sample average of L_a is found, and the p-values for the differences between a and each of the other treatments are computed by a paired sample *t test* to determine whether these differences are significant.

4.2.3 Signals

Several different clean time-series are used in the comparative experiments[2]. The first two are generated from a known autoregressive function $f(\cdot)$ according to:

$$x_k = f(x_{k-1}, \ldots x_{k-M}, \mathbf{w}) + v_k \qquad \forall k \in \{1 \ldots N\}, \tag{4.3}$$

where v_k is a zero-mean white Gaussian process. Initial conditions are obtained by starting at random values of x, and running the recursion until the transients disappeared. These transients are then omitted by removing the first several hundred points of the signal. As described below,

[2] Additional, application-specific time-series will be described in later sections.

the first signal uses a linear function for $f(\cdot)$, and the second two use neural networks. The fourth and fifth signals are generated by known chaotic maps.

Linear AR-10 Signal

One of the signals used in the experiments is generated by a linear function $x_k = \mathbf{w}^T\mathbf{x}_{k-1}$ with 10 taps ($M = 10$). This results in a linear IIR (all-pole) model driven by white process noise, commonly referred to as an autoregressive (AR) system [46]. The weight vector and process noise variance used to generate the data are:

$$\mathbf{w} = \begin{bmatrix} .9 & .3 & -.4 & .2 & -.1 & .1 & -.3 & .2 & .01 & -.05 \end{bmatrix}^T, \quad \text{and} \quad \sigma_v^2 = .09. \tag{4.4}$$

A portion of signal is shown in Figure 4.3. The main utility of this time-series is that it satisfies all of the assumptions of Gaussianity made in the theoretical development of this thesis. The linear AR data therefore allow testing of the algorithms under "ideal" conditions.

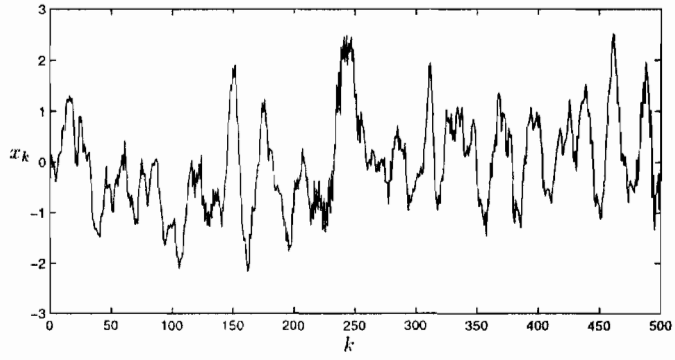
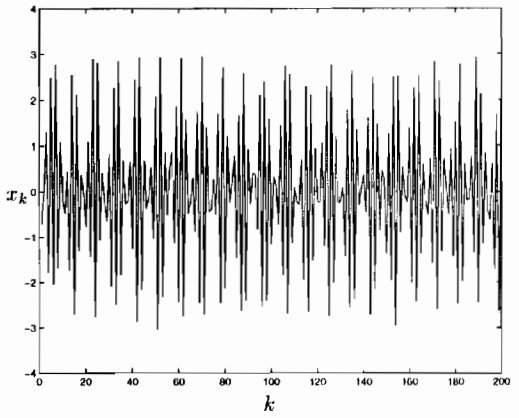


Figure 4.3: Data generated by 10th order linear AR model.

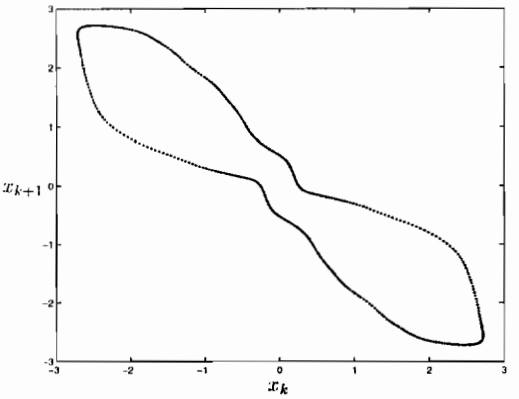
Nonlinear Neural Network Signals

To generate a nonlinear time-series, a feedforward neural network with 10 inputs, 5 hidden units, and 1 output is used as the nonlinear autoregressive function $f(\cdot)$ in Equation 4.3. Two different networks (*i.e.*, with different weights $\mathbf{w}$) are used to generate time-series with different dynamical properties; this is useful for testing the hypothesis that the relative performance of different costs should be similar at different points in the weight space.

The first network is driven with white Gaussian process noise with variance $\sigma_v^2 = 0.04$. The resulting signal is shown in Figure 4.4(a), along with a phase diagram of the undriven dynamics (no process noise) in Figure 4.4(b). As shown, a limit cycle is produced: a fairly simple form of nonlinear dynamics.

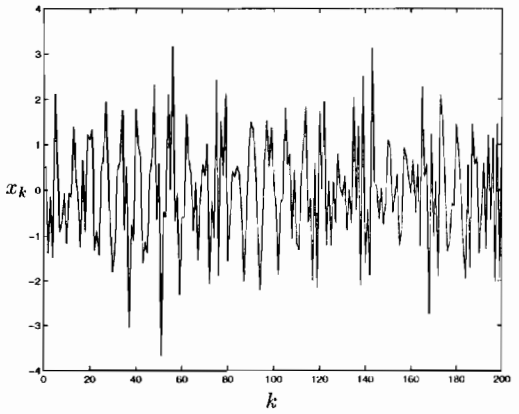


(a) Portion of first neural network signal

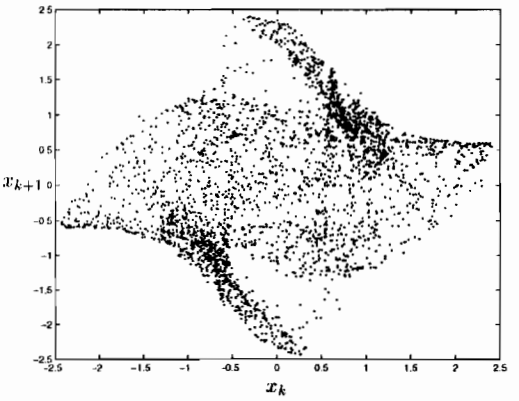


(b) Phase diagram of undriven dynamics

Figure 4.4: Data generated by limit cycle neural network.



(a) Portion of second neural network signal



(b) Phase diagram of undriven dynamics

Figure 4.5: Data generated by chaotic neural network.

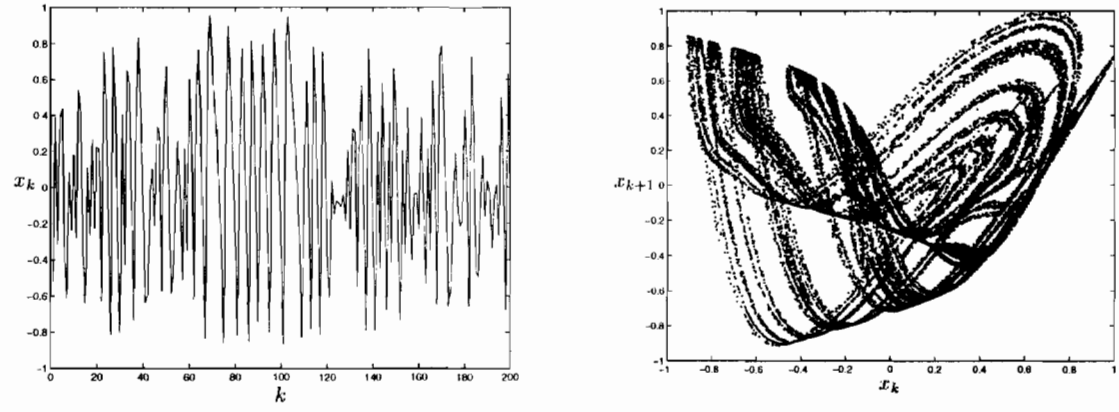
The second network is driven with white Gaussian process noise with variance $\sigma_v^2 = 0.36$. The resulting signal is shown in Figure 4.5(a), along with a phase diagram of the undriven dynamics (no process noise) in Figure 4.5(b). The dynamics of this network are considerably more complex than the first, and appear chaotic in nature.

Ikeda Chaotic Series

The Ikeda chaotic map [27] is defined by the discrete-time, complex valued function:

$$z_{k+1} = a + R \cdot z_k \cdot e^{j\left(\phi - \frac{p}{1+|z_k|^2}\right)} \quad \forall k \in [1, \infty), \tag{4.5}$$

where z_0 is chosen randomly, $a = 1$, $R = 0.9$, $\phi = 0.4$, $p = 6$. A one-dimensional time-series was produced by taking the real part of the data; *i.e.*, $x_k = \Re(z_k)$. Furthermore, the data are normalized to have zero mean and fall approximately in the range $[-1, 1]$. The time-series is shown in part, along with its phase diagram, in Figure 4.6.



(a) Portion of signal

(b) Phase diagram of signal

Figure 4.6: Chaotic Ikeda data.

The linear and neural network models have the advantage that the exact model structure is known in advance, thereby removing any effects which might arise from using a model which is either too flexible or too rigid during the dual estimation process. In contrast, if a neural network is used to model the noisy Ikeda data, the architecture is not known *a priori*, nor are the optimal set of weights $\mathbf{w}$ and variance σ_v^2.

As these parameters are not required for dual estimation, this does not pose any problem for the use of the algorithms. On the other hand, it makes establishing a benchmark result for the known-model case difficult on the Ikeda data. One solution is to train a neural network as a predictor on the clean data, and to interpret this as the true model. However, the chaotic properties of the Ikeda data set make this a difficult modeling problem in its own right, and even when a reasonably accurate model is found, it proves ineffective when used for signal estimation in the Kalman filtering context. The reason for this is probably related to the severity of the

nonlinearities in the system, which disrupt the Gaussianity of the statistics, so that this "true" model is no longer optimal for estimating noisy data.

Mackey-Glass Series

A continuous-time chaotic map, first described by Mackey and Glass ([49], 1977) for modeling the dynamics of white blood cell production in the human body, is given by the differential equation:

$$\frac{dx(t)}{dt} = \frac{.2x(t-\tau)}{1+x^{10}(t-\tau)} - .1x(t). \tag{4.6}$$

Here, τ is delay parameter which results in either fixed point, limit cycle, or chaotic behavior. The system has been used frequently in the literature for testing nonlinear predictive models (*e.g.*, in [40], [85], [31]). The experiments later in this chapter use a delay of $\tau = 30$, which produces a strange attractor with a fractal dimension near 3.5. Following the convention in the literature, the continuous time signal is sampled every 6 seconds to produce the discrete-time series shown in Figure 4.7.

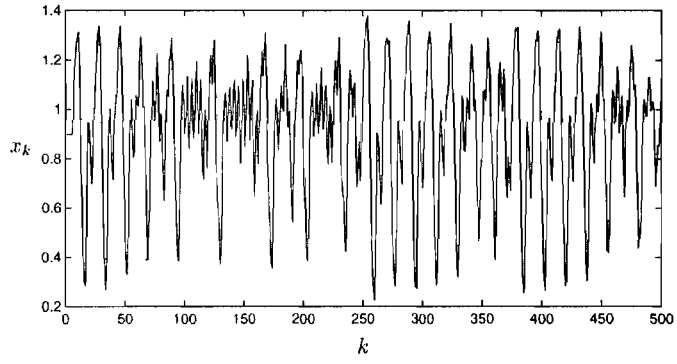


Figure 4.7: Data generated by Mackey-Glass equation with $\tau = 30$ and a sampling period of 6.

Normalization

Because the above data sets are generated so as to fall in a reasonable numerical range, normalization is not a critical issue. Nevertheless, to provide consistency, and to allow the use of a common set of initial error covariances $\mathbf{P}_0$, $\mathbf{Q}_0$, $q_{v,0}$, and $q_{n,0}$, the noisy time-series are normalized to be zero-mean, and to fall in the range $[-1, 1]$ so that $\max|y_k| \approx 1$. Of course, the unnormalized time-series is used whenever the true model is employed for signal or variance estimation: the data must match the model in this case.

4.2.4 Measurement Noise

Several different classes of noise are investigated, ranging from stationary white Gaussian noise, to nonstationary colored noise, to pink noise. As explained in Section 4.2.2, several realizations of the noise are required for each noise type. These data are scaled appropriately before adding them to the clean signals, to produce time-series at the desired SNR.

White Stationary Noise

White noise refers to a signal whose value at time k is statistically independent of its value at time $k-1$. Typically, the pseudorandom numbers generated by a computer can be considered to form a white noise sequence in this sense. As an alternative, the Signal Processing Information Base (SPIB) at Rice University [69] makes available 235 seconds of white noise, which was sampled from an analog noise generator with 16 bit precision at a rate of 19.98 kHz. The original data are in integer format.

To produce the noise samples $\{n_k\}_1^N$ used in these experiments, the SPIB data set was normalized to fall in the $[-1, 1]$ range, and was divided into nonoverlapping segments to produce the required repetitions.

White Nonstationary Noise

Nonstationary white noise data are produced by modulating each of the above stationary noise realizations with a sine wave. The D.C. offset of the sine wave is 1, and the amplitude is 0.2. The period is about 15,000 points, thereby producing a white noise signal with a slowly chang-

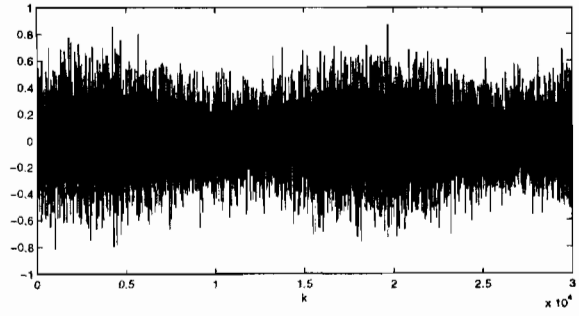


Figure 4.8: White nonstationary noise.

ing variance $\sigma_{n,k}^2$ (see Figure 4.8). Strictly speaking, the resulting noise is not truly white, but the time-scale of the amplitude modulation is slow enough that the noise remains approximately uncorrelated (white) on smaller time-scales.

Autoregressive Colored Noise

To generate noise with temporal correlations between data points, a linear 5^{th}-order AR model is used. To ensure that reasonable noise-like dynamics are obtained, the model is itself trained to predict sampled analog pink noise. The noise model parameters are:

$$\mathbf{w}_n = \begin{bmatrix} 0.6297 & 0.0515 & 0.1061 & -0.0024 & 0.0893 \end{bmatrix}^T, \quad \text{and} \quad \sigma^2_{v_n} = .09. \tag{4.7}$$

Generating a noise signal from this AR-5 model ensures that everything about the noise is known.

Note that colored noise of a given power is "less random" than white noise at the same power, because the colored noise has a certain component that is predictable, or deterministic. Hence, the signal estimation NMSEs for a signal corrupted by colored noise are typically less than they are for white noise at the same SNR.

Autoregressive Nonstationary Noise

The above noise model can also be used with a time-varying process noise variance, $\sigma^2_{v_n,k}$, to produce a nonstationary colored noise series. Note that the nonstationarity in this case is highly restricted, as the parameters $\mathbf{w}_n$ remain constant. The standard deviation of the process noise is modulated as:

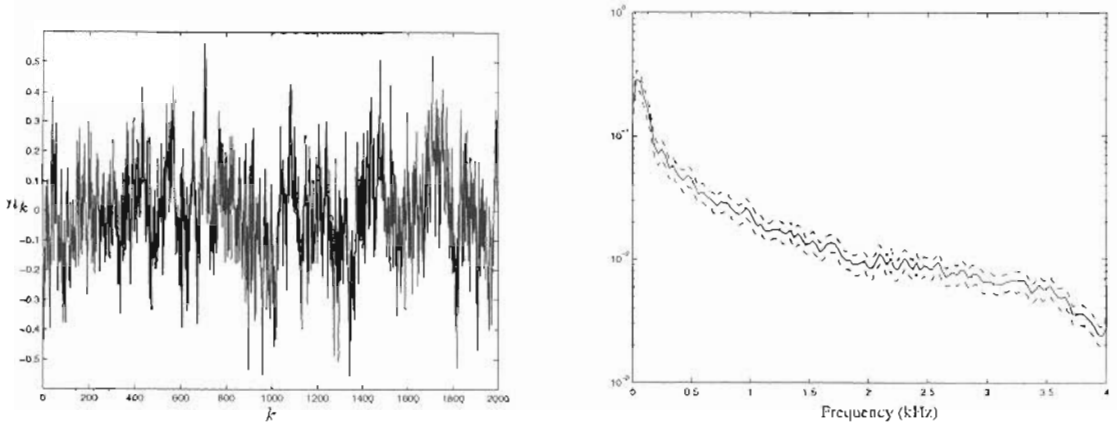
$$\sigma_{v_n,k} = 0.3 gain_k \qquad \text{where} \quad gain_k = 1 + 0.2 \sin\left(\frac{k}{2000}\right) \tag{4.8}$$

producing a nonstationarity with a period of about 12,500 points.

Pink Noise

While white noise has a flat frequency spectrum, showing equal energy at all frequencies, the power spectral density of *pink* noise decreases as the inverse of the frequency. Alternatively, it can be described as having an equal amount of energy in each 1/3 octave band. The noise is called "pink" because, if the spectrum were interpreted in the electromagnetic domain as visible light frequencies, the signal would appear as pink light due to its emphasis of longer (red) wavelengths.

The SPIB [69] resource contains pink noise sampled with 16 bit precision at a rate of 19.98 kHz from an analog noise generator. This integer-valued noise is downsampled to 8 kHz and normalized to the range $[-1, 1]$, then segmented into 30,000 point sections. The power spectral density of one such segment is shown in Figure 4.9(b). Unlike the stationary AR-5 noise described above, the true model order of this pink noise is uncertain (as is the weight vector $\mathbf{w}$ and variance $\sigma^2_{v_n}$). As with the Ikeda series, the purpose of this data set is to test the algorithms in conditions wherein the correct model structure is not known exactly.



(a) Portion of noise

(b) Power spectral density

Figure 4.9: Pink noise

4.3 Synopsis of Results

Numerous experimental results are reported in the subsequent sections of this Chapter, in the form of boxplot figures, summarizing tables, and discussion. Because the results are fairly detailed, this section provides an overview of the major findings in each experiment.

1. *Initial Error-Covariances:* Appropriate values for the initial covariances, $\mathbf{P}_0$ and $\mathbf{Q}_0$, are investigated in the context of both linear and neural network signals, using normalized data. The main conclusion from these results is that too large a value of the initial weight covariance, $\mathbf{Q}_0$, can sometimes prevent the dual EKF from converging. The results – reported in terms of overall signal NMSE – are generally not very sensitive to the signal covariance, $\mathbf{P}_0$. For most cost functions and data sets, $\mathbf{P}_0 = \mathbf{I}$ and $\mathbf{Q}_0 = .1\mathbf{I}$ produce good results; however, a later experiment shows that the $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ costs are prone to stability problems, and require a smaller value of $\mathbf{Q}_0$ $(.01\mathbf{I})$ in many circumstances.

2. *Variance Estimation:* The cost functions for estimating the noise variances are compared (beginning on page 134) using an EKF with known weights, $\mathbf{w}$. When estimating σ_v^2, the joint cost gives the best long-term result on white noise, although it shows slightly slower convergence speed than the maximum-likelihood cost. On colored noise data, the joint cost appears prone to under-estimation of the variance, leaving $J^{ml}(\sigma_v^2)$ as the best choice in this case. For estimating the measurement noise variance (σ_n^2 or $\sigma_{v_n}^2$) with σ_v^2 known, the maximum-likelihood cost was consistently better than any other cost.

The results are less satisfying when both variances are unknown, with the $J^{j}(\sigma_v^2)$ replaced as the top choice on white noise by $J^{ec}(\sigma_v^2)$ and $J^{pe}(\sigma_v^2)$, and the maximum-likelihood measurement noise variance cost replaced by $J^{eml}(\sigma_n^2)$ and $J^{pe}(\sigma_{v_n}^2)$ in some cases. Contributing to the variation in results might be the inaccuracies of the EKF signal estimator itself. Fortunately, the results are much more consistent when the weights are estimated as well – with a dual Kalman filter – in the fourth experiment.

3. *Forgetting Factor:* The results reported in this section pertain to the scalar λ, used in the weight and variance filters to determine how quickly old data are "forgotten" by an exponential window. For stationary data, a value around $\lambda_{\mathbf{w}} = 0.9999$ produces good results, although this number will probably depend on the complexity of the model. Variance estimation involves only a single unknown parameter for each variance filter; less data should therefore be required. A value of $\lambda_{\sigma^2} = 0.9993$ appears the best for either of the variance filters. Meanwhile, the best choices of forgetting factor on nonstationary data – such as sinusoidally modulated white noise – is dominated by the rate of nonstationarity itself. The value of λ should be as large as possible while still allowing tracking of the noise variance.

4. *Dual Kalman Weight Costs:* The various dual EKF weight costs are compared in Section 4.7 on page 148. With both variances known, the $J^{pe}(\mathbf{w})$ and $J^{ml}(\mathbf{w})$ costs excel on white noise, while $J^{j}(\mathbf{w})$ and $J^{ml}(\mathbf{w})$ do the best with colored noise. However, the maximum-likelihood cost is prone to unstable behavior (an ill-conditioned Hessian); it and the error-coupled joint cost both require a smaller value of $Q_0 = .01$ to prevent this, and can go unstable even so.

   When the process noise variance, σ_v^2, is estimated along with the weights, the rankings of weight estimation costs are much the same. Meanwhile, the variance estimation cost $J^{ml}(\sigma_v^2)$ is the best across all data sets and SNRs. Estimation of both variances (measurement noise and process noise) shows the maximum-likelihood cost to be the most effective, generally, for estimating σ_n^2 (or $\sigma_{v_n}^2$), as well.

   Overall, the dual EKF algorithm works very well, and shows good robustness to uncertainty in the noise variances. Comparing the dual EKF results (both with known and unknown variances) with those of an EKF shows that the dual EKF can actually compensate for the inaccuracies of the EKF in some circumstances, and produce better results than when the weights and noise variances are known (see Figure 4.46 on page 179)!

5. *Static Derivatives in the Dual EKF:* The next experiment explores the effect of using static derivatives in place of the dual EKF's recursive derivatives of $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ with respect to $\mathbf{w}$.

As expected, the performance is degraded by this approximation; although the difference is less significant on white noise data, recursive derivatives are play an important role for data in colored noise (see Figure 4.48 on page 182).

6. *Joint EKF Performance:* In Section 4.9 on page 183, the best dual EKF cost functions are compared with the joint EKF algorithm. There is little difference in performance when both variances are known, except that the dual EKF performs significantly better on the Ikeda data, for which the model structure is unknown. When σ_v^2 is estimated, the joint EKF does significantly better on linear data in white noise, and when tracking nonstationary noise at higher SNRs. However, letting σ_n^2 (or $\sigma_{v_n}^2$) be unknown as well makes the dual EKF the better performer on white noise. The dual EKF is better on the Ikeda data in all cases. In contrast to reports elsewhere in the literature [45, 47, 61] the joint EKF exhibited no stability or convergence problems during these experiments.

7. *Model Mismatch Effects:* The effect of uncertainty in the model structure is investigated formally in this experiment, and shows that the dual EKF is considerably more robust than the joint EKF to a model structure that is either underparameterized, or overparameterized, with respect to the underlying signal. This form of robustness is important for most applications where the model structure is not known *a priori*.

8. *Over-Training:* The experimental results in Section 4.11 on page 192 demonstrate the susceptibility of dual estimation algorithms to over-training whenever a finite data set is used in an iterative fashion. This underscores the importance of an early-stopping technique to maintain good generalization in the test set. The results are otherwise consistent with those for the on-line (infinite data) case.

The above experiments are described in more detail in the following sections.

4.4 Experiment 1: Initial Error-Covariances

The dual EKF algorithm requires initial values for the signal-state covariance, $\mathbf{P}_k$, and the weight covariance, $\mathbf{Q}_k$, at time $k = 0$. If the signal has been normalized to be approximately unit variance, then $\mathbf{P}_0 = \mathbf{I}$ is a reasonable choice. A reasonable value for $\mathbf{Q}_0$ is less clear, and involves several factors discussed in Section 3.3.2 on page 62. In addition to finding good initial values, the sensitivity of the dual EKF to these values should be determined.

To obtain this information, the dual EKF is run using three of the five cost functions derived in Chapter 2: the prediction error cost $J^{pe}(\mathbf{w})$, the joint cost $J^{j}(\mathbf{w})$, and the EM cost $J^{em}(\mathbf{w})$.

Both the process noise variance σ_v^2 and measurement noise statistics (σ_n^2 in the white noise case, $\sigma_{v_n}^2$ and $\mathbf{w}_n$ in the colored noise case) are known. The forgetting factor, λ, is 1.

Two different noisy data sets are used.

- the AR-10 series with white stationary noise added at 0 dB.
- the chaotic neural network series (NN) with autoregressive stationary noise added at 3 dB.

Each of the two series $\{x_k\}_1^N$ contains $N = 10,000$ points, and is corrupted with 10 different realizations of the corresponding noise series, for a total of 20 different noisy series $\{y_k\}_1^N$.

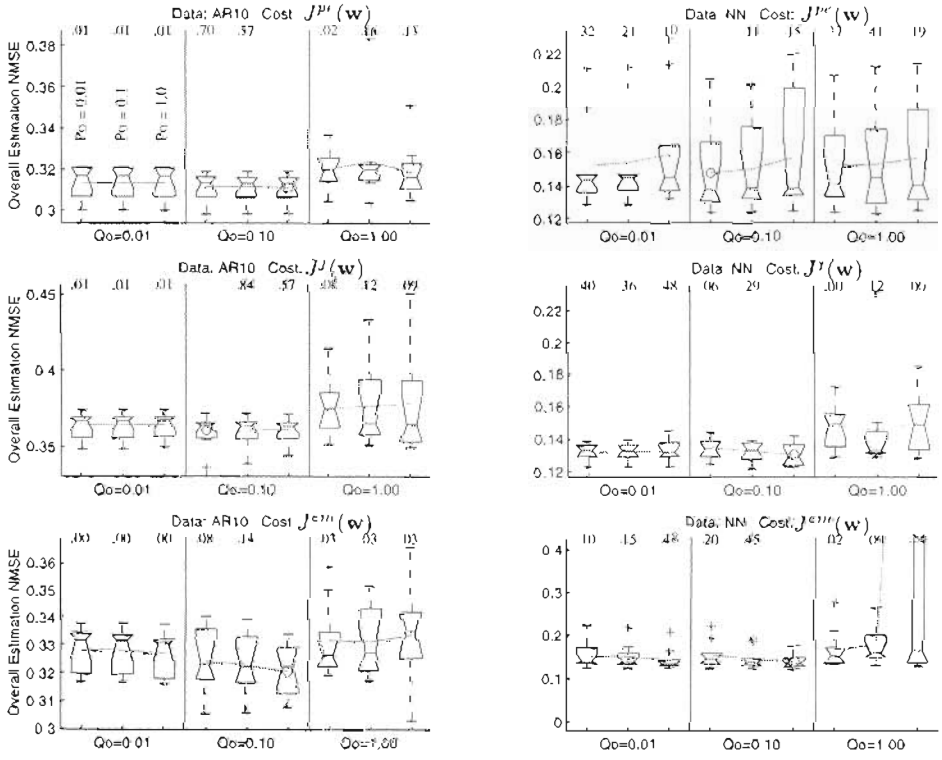


Figure 4.10: Overall estimation NMSE statistics. For each of the six different test conditions, boxplots are shown for P_0 increasing as $\{.01, .1, 1\}$ from left to right within each panel, and Q_0 increasing between panels as indicated. Horizontal lines indicate the median, and upper and lower quartile values, and range of the overall NMSE. The mean values within each Q_0 are linked by thin lines, and the choice of P_0 and Q_0 with the lowest mean NMSE is indicated by a superimposed circle. The p-value for the difference between this choice and each of the others is shown along the top of each plot.

The covariances $\mathbf{P}_0$ and $\mathbf{Q}_0$ are tested at values $P_0 \cdot \mathbf{I}$ and $Q_0 \cdot \mathbf{I}$, respectively, where P_0 and Q_0 are scalars chosen from $\{ .01, .1, 1 \}$. Thus, there are a total of 9 different test configurations, tested at 6 different algorithm-data combinations, and with 10 repetitions each.

The most informative performance criterion in this case is the overall estimation error. The

prediction error is strongly correlated with the estimation error, but exhibits lower resolving power because of the inclusion of process noise. The short-term (first 100 point) errors are consistently biased towards small values of both $\mathbf{P}_0$ and $\mathbf{Q}_0$, which produce lower-variance estimates (at the expense of high bias) before the algorithms have seen enough data to provide reliable results. On the other hand, the final 1000-point NMSE has little dependence on initial parameter values. The error trajectories are fairly consistent across algorithms, and are informative mostly in showing that convergence occurs before the end of the data.

The results are summarized graphically in Figure 4.10. The clearest conclusion is that $\mathbf{Q}_0$ should *not* be as large as $\mathbf{I}$. In all cases, the minimum mean NMSE appears for $\mathbf{Q}_0 = .1\mathbf{I}$, although the difference between $\mathbf{Q}_0 = .1\mathbf{I}$ and $\mathbf{Q}_0 = .01\mathbf{I}$ is significant only on the linear data.

Choosing $\mathbf{P}_0$ is even more difficult. Setting $\mathbf{P}_0 = \mathbf{I}$ can be justified on the grounds that it has the lowest mean NMSE in most cases, and never differs from the optimal choice with p-value lower than 15%. However, using $\mathbf{P}_0 = .1\mathbf{I}$ is also reasonable because it never differs with p-value lower than 14%, although this choice never attains the lowest mean. While $\mathbf{P}_0 = .001\mathbf{I}$ gives the minimum in two cases, the advantage is completely insignificant in one of these (AR-10 data, joint cost). Furthermore, it is significantly worse than $\mathbf{P}_0 = \mathbf{I}$ in two other cases, with a p-value of less than 10%.

In any case, the choices $\mathbf{Q}_0 = .1\mathbf{I}$ and $\mathbf{P}_0 = \mathbf{I}$ seem to convey a slight advantage, and are used in the remainder of the experiments presented, unless indicated otherwise. Actually, the lack of sensitivity indicated by the paired-sample t tests is somewhat encouraging, because the initial values of the covariances should, ideally, not have a strong effect on the results. The most important condition is that $\mathbf{Q}_0$ be small enough to prevent instability.

4.5 Experiment 2: Variance Estimation

If the noise variances σ_v^2 and σ_n^2 (or $\sigma_{v_n}^2$) are unknown, they must be estimated using, for example, the Kalman variance estimation algorithm given in Formula 3.13. In this case, the error variances of the filters must be initialized, just as is required by the signal and weight filters. However, because the state is one-dimensional in this case, these initial error variances, $q_{v,0}$ and $q_{n,0}$, are scalar-valued. In addition to choosing $q_{v,0}$ and $q_{n,0}$, variance estimation requires choosing a cost function. To simplify the search, the weights $\mathbf{w}$ are assumed known, so that only the variances and signal need to be estimated.

An example result for neural network signal in 3dB colored noise is shown in Figure 4.11, in which the estimates $\hat{\sigma}_v^2$ and $\hat{\sigma}_{v_n}^2$ are plotted as functions of time. The true values, σ_v^2 and $\sigma_{v_n}^2$ are

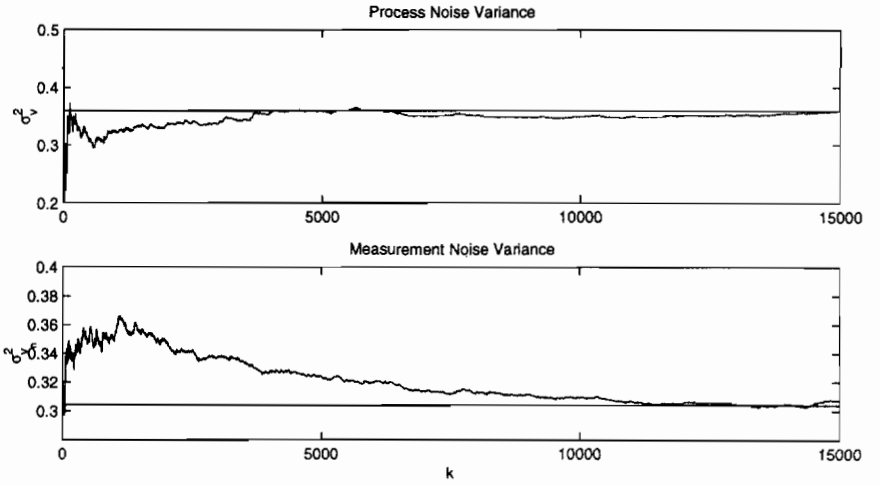


Figure 4.11: Example variance estimation trajectories, using the maximum-likelihood cost function and known weights. True values of the variances are shown by the horizontal lines.

shown by horizontal lines, for comparison. In this example, the maximum-likelihood cost is used to estimate both variances, and initial error variances are $q_{v,0} = .1$ and $q_{n,0} = .01$. In the following experiments, a variety of cost functions and initial covariance values are tested.

The cost functions and initial variances are evaluated with 15,000 points of chaotic neural network data, corrupted by either white stationary (WS) noise or autoregressive (AR-5) stationary (AS) noise. Both noise types are added to the clean signal at 3 different SNRs: 0 dB, 3 dB, and 7 dB; each of the six noisy signal combinations is replicated 10 times. Boxplots and paired-sample t tests are used to select both q_0 values and the best cost function for variance estimation.

All five of the cost functions are tested: prediction error, maximum-likelihood, joint, joint error-coupled, and EM. The initial variances are tested at $q_0 \in \{.001, .01, .1\}$. Other parameter values are $\mathbf{P} = \mathbf{I}$ and $\lambda = 1$. Matters are complicated somewhat by the fact that sometimes only one of the noise variances might need to be estimated, and sometimes both might. All three possible situations are considered, each tested with 10 repetitions of the 6 different noisy time-series.

4.5.1 Estimating the Process Noise Variance

First, consider the case of known measurement noise statistics. If not known *a priori*, these statistics can sometimes be estimated from portions of the data wherein no signal is present. Because the current experiment also uses a known signal model $f(\cdot)$, only the process noise variance σ_v^2 and signal are estimated concurrently. The signal is estimated with a standard EKF (Formulae 3.1-3.2), while σ_v^2 is estimated with the alternative variance filter shown in Formulae 3.12- 3.13 on page 78.

The focus of the experiment is on variance estimation, so the variance MSE is a reasonable

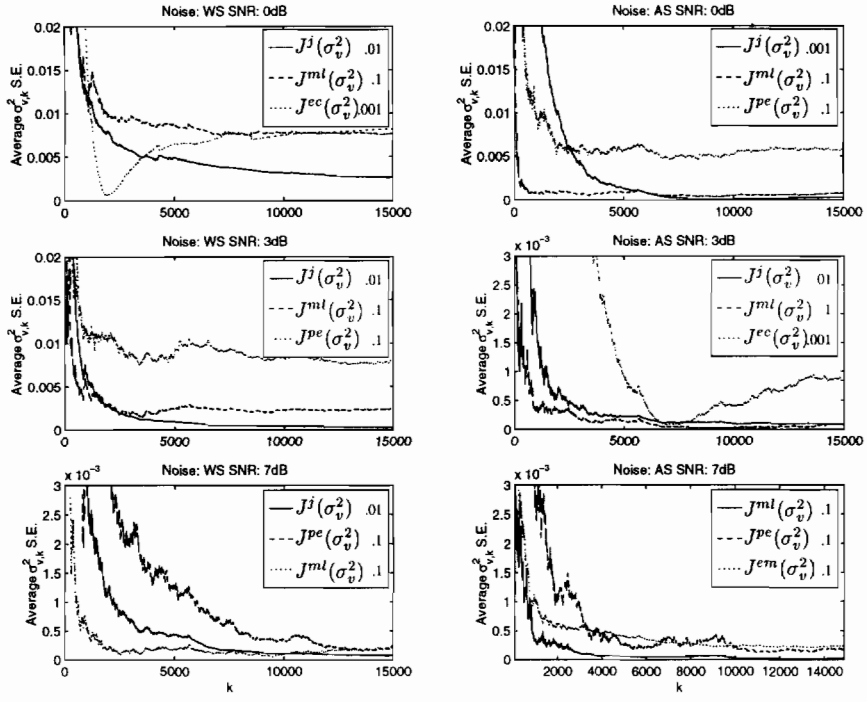


Figure 4.12: The average squared-error trajectories of the estimates $\hat{\sigma}^2_{v,k}$ are plotted for the three best combinations of $q_{v,0}$ and cost function.

criterion for comparing treatments. Although the signal estimation error is also informative, it tends to be highly correlated with the variance estimation error, and so does not provide any new information.

The algorithms are ranked by time-averaging the squared error of the variance estimates over the final 1000 data points, which provides an evaluation of each algorithm near convergence. For the best three treatments ranked in this way, the squared-error trajectories of the variance estimates, averaged over the ensemble of 10 repetitions, are displayed in Figure 4.12. Separate plots are shown for the three levels each of white and AR-5 noise. The significance of the rankings are indicated by the boxplots in Figure 4.13.

One disadvantage of the squared-error trajectories is that information about the actual values of the estimates is obscured. For example, on the 0dB WS data, the $J^{ec}(\sigma_v^2)$ method shows a minimum in the average squared-error trajectory at around $k = 2000$, followed by an increase in the error. A similar effect is seen in the 3dB AS trajectories near $k = 7000$.

The fact that this is caused by the under-estimation of σ_v^2 on average by $J^{ec}(\sigma_v^2)$ is evident from the average time trajectories of the variance in Figure 4.14. Although not shown in the figures, σ_v^2 is under-estimated by $J^{ec}(\sigma_v^2)$ with $q_{v,0} = .001$ in all cases except for 7dB AS noise, on which it converges at too large a value. For $q_{v,0} = .1$, the algorithm can be unstable, generating

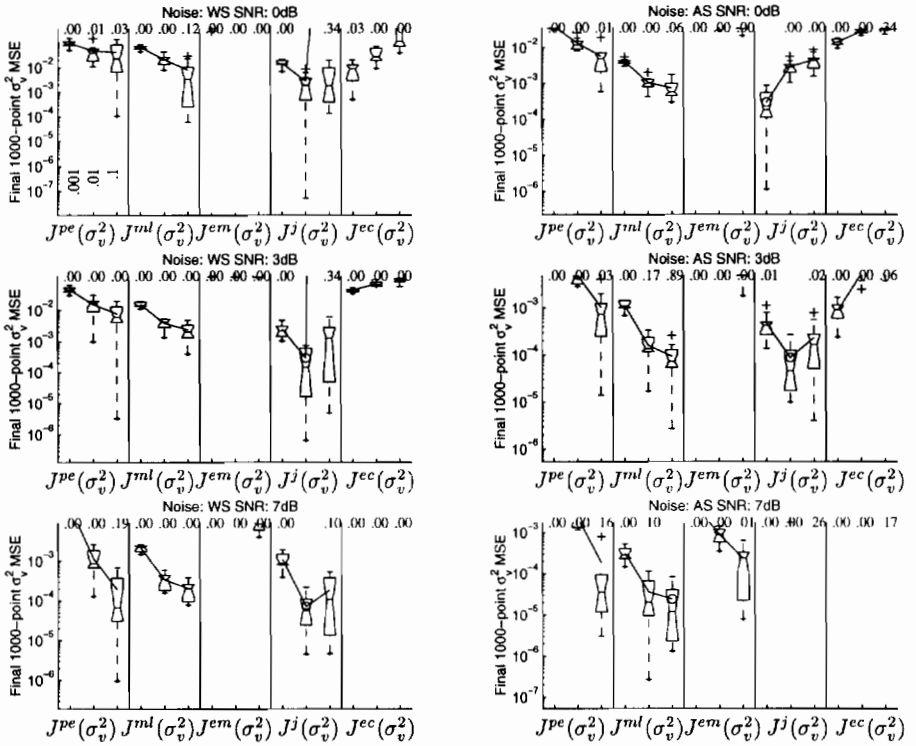


Figure 4.13: Variance MSEs computed over the final 1000 points. For each of the six different test conditions, boxplots are shown for $q_{v,0}$ increasing as $\{.001,\ .01,\ .1\}$ from left to right within each panel; each panel represents a different cost function, as indicated. See the caption of Figure 4.10 for additional information.

either very large values, or values close to zero ($\ell \approx -\infty$).

The joint cost $J^j(\sigma_v^2)$ with $q_{v,0} = .01$ is arguably the best choice for white noise. However, on the AS noise, the performance of $J^j(\sigma_v^2)$ is less consistent: at 0dB it under-estimates the variance, while at 7dB it over-estimates the variance (not shown). Slower convergence speed appears to be a general drawback of the $J^j(\sigma_v^2)$ approach; even on white noise, where its final performance is superior, it exhibits slower convergence than the maximum-likelihood or prediction-error cost functions. In fact, for the AS 0dB case, the apparent advantage of the $J^j(\sigma_v^2)$ $q_{v,0} = .001$ treatment is largely a spurious effect of its slow convergence for that value of $q_{v,0}$ (the algorithm has not converged yet). Although not shown in the plots, the $J^j(\sigma_v^2)$ method converges much faster at $q_{v,0} = .01$ and $q_{v,0} = .1$, and for those values under-estimates the variance by a larger amount than the other algorithms over-estimate it.

The maximum-likelihood cost $J^{ml}(\sigma_v^2)$ exhibits fast convergence to good solutions. The method is ranked in the top three on all six data sets, and is arguably the best choice for colored noise. As mentioned above, the second place ranking of $J^{ml}(\sigma_v^2)$ on 0dB AS noise is probably due largely

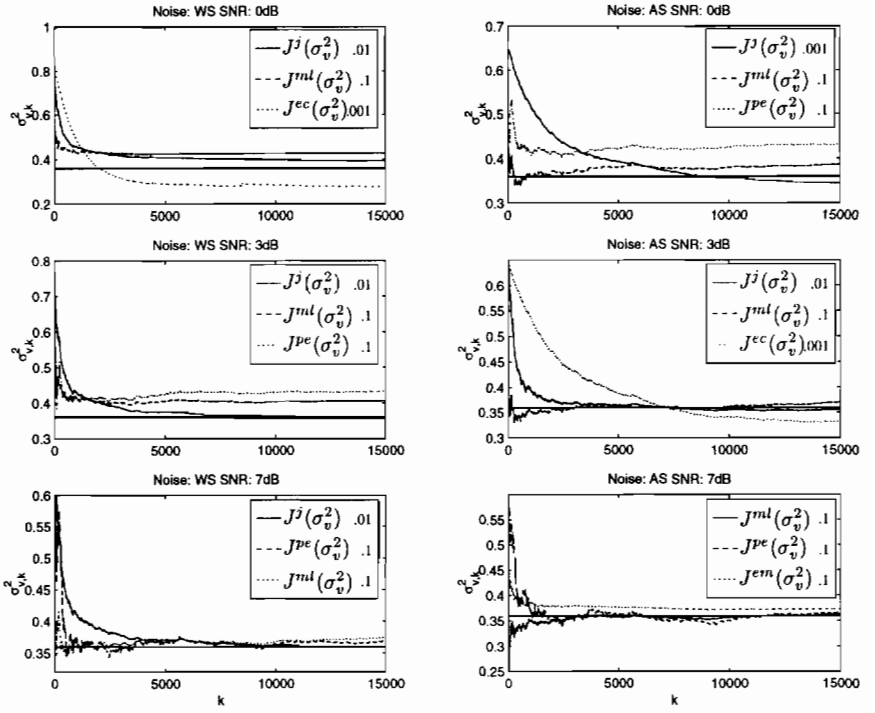


Figure 4.14: The average trajectories of the variance estimates $\hat{\sigma}^2_{v,k}$ are plotted for the three best combinations of $q_{v,0}$ and cost function.

to the fact that $J^j(\sigma_v^2)$ has not yet converged at $N = 15,000$. Furthermore on 3dB AS noise, the $J^{ml}(\sigma_v^2)$ and $J^j(\sigma_v^2)$ methods are not significantly different, showing a p-value of 89%. At 7dB, the $J^{ml}(\sigma_v^2)$ $q_0 = .1$ method conveys a significant advantage. The main deficits of the approach are its high volatility at early times, and its slight tendency towards over-estimation at later times. However, as discussed in Appendix D, an inflated estimate of σ_v^2 can ameliorate the negative effects of the EKF approximation by accounting for the inaccurate mean propagation.

The EM cost over-estimates σ_v^2 in all cases, but by decreasing amounts for higher SNRs. On 7dB AS noise, the method actually places in the top three. However, it is clear that the cost function is not amenable to on-line estimation of σ_v^2. This could be due to approximations made in the E-step, or because the cost function surface is difficult to navigate for some reason.

The prediction error cost, $J^{pe}(\sigma_v^2)$, does moderately well on the higher SNR cases, but never outperforms the $J^{ml}(\sigma_v^2)$ method. The MSE trajectories generally show higher volatility, slower convergence, and higher final MSE than their $J^{ml}(\sigma_v^2)$ counterparts. On WS noise at 7dB, the method comes in second place behind the $J^j(\sigma_v^2)$ cost, but with only a slightly lower average MSE, and significantly higher variance than the $J^{ml}(\sigma_v^2)$ cost. This high variance is primarily responsible for the p-value of 19% in this case.

In conclusion, the joint cost is the best choice for estimating σ_v^2 in white noise, while the maximum-likelihood cost is somewhat preferable in the presence of colored noise. Unlike the weight and signal estimation filters, the variance estimation filter is clearly sensitive (as shown in Figure 4.13) to the value of the initial error variance, $q_{v,0}$. As discussed previously, this value affects how fast the algorithm can converge, and too large a value can cause instability. Moreover, the best choice of $q_{v,0}$ depends on the cost function; for $J^{ml}(\sigma_v^2)$, the best value is $q_{v,0} = .1$, while $q_{v,0} = .01$ is better for $J^{j}(\sigma_v^2)$. Values of $q_{v,0} = 1$ were found to cause too much instability, and so were not evaluated formally. The conclusions of this experiment are summarized in Table 4.1.

Table 4.1: Best choices of cost function and initial variance $q_{v,0}$ when estimating σ_v^2 with an EKF using known weights $\mathbf{w}$ and measurement noise statistics.

	White Stat.		AR-5 Stat.	
SNR	Cost	$q_{v,0}$	Cost	$q_{v,0}$
0 dB	$J^{j}(\sigma_v^2)$	.01	$J^{ml}(\sigma_v^2)$	.1
3 dB	$J^{j}(\sigma_v^2)$	.01	$J^{ml}(\sigma_v^2)$	.1
7 dB	$J^{j}(\sigma_v^2)$	.01	$J^{ml}(\sigma_v^2)$	.1

4.5.2 Estimating the Measurement Noise Variance

In some applications, a great deal is known in advance about the statistics of the signal, but little is known about its SNR. That is, while the process noise variance σ_v^2 is known, the variance σ_n^2 of the measurement noise must be estimated. When the noise is colored, only its power (or equivalently, its process noise variance $\sigma_{v_n}^2$) is assumed unknown. This situation might arise if the structure (*i.e.*, the spectral shape) of the noise has been estimated beforehand, and is expected to remain stationary, but the *level* of the noise is uncertain.

For the colored noise experiments, the exact model $\mathbf{w}_n$ of the AR-5 noise is used; this ensures that the true value of $\sigma_{v_n}^2$ can be taken as the optimal value against which to compare $\hat{\sigma}_{v_n,k}^2$. For each repetition of the data, initial estimates $\hat{\sigma}_{v_n,0}^2$ are obtained from a 500-point segment of the noise. In the white noise case, $\hat{\sigma}_{n,0}^2$ is estimated *via* the ad hoc procedure described in Section 3.6.2.

The final 1000-point MSE of the variance estimate is used as the ranking criterion. The error trajectories for the three best treatments, averaged over the ensemble of 10 repetitions, are displayed in Figure 4.15. The significance of the rankings are indicated by the boxplots in Figure 4.16.

The maximum-likelihood method is the best choice on all data sets, and conveys a significant advantage in nearly all cases. The only exception is on the 7dB AS noise, on which the $J^{pe}(\sigma_{v_n}^2)$ and EM treatments are not significantly worse, but this sort of equalization among treatments is

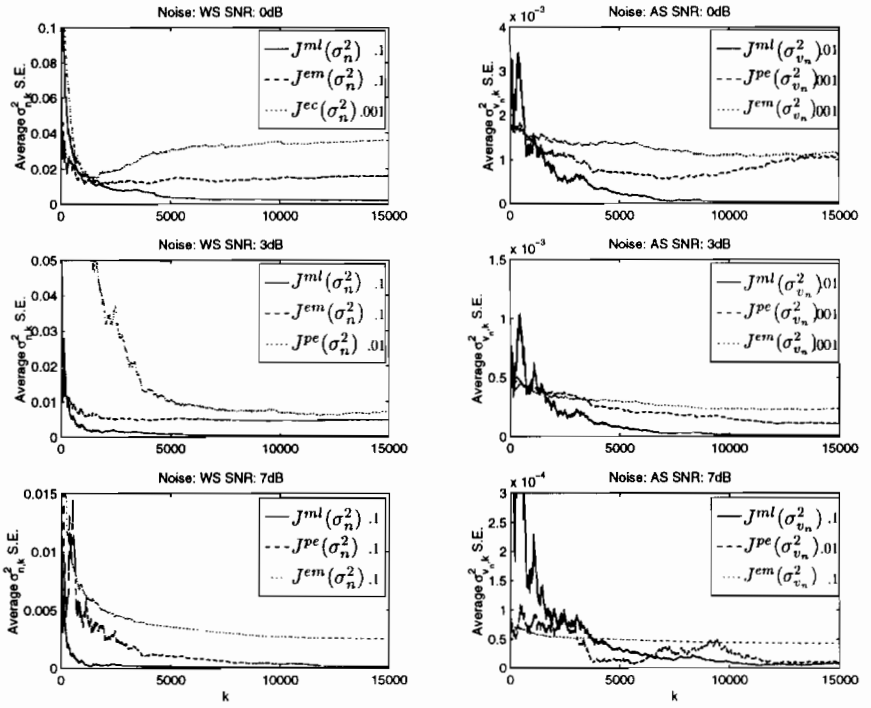


Figure 4.15: The average squared-error trajectories of the estimates $\hat{\sigma}^2_{n,k}$ are plotted for the three best combinations of $q_{n,0}$ and cost function. Averages are computed over 10 repetitions of the data and initial conditions.

to be expected at high SNRs. While the $J^{em}(\sigma_n^2)$ and $J^{pe}(\sigma_n^2)$ costs are nearly always in the top three, they typically converge to significantly higher MSEs than does the $J^{ml}(\sigma_n^2)$ cost.

For the maximum-likelihood method, the choice of $q_{n,0} = .1$ is generally the best on white noise, while a slight but insignificant advantage is conveyed by $q_{n,0} = .01$ on colored noise. Although $q_{n,0} = .1$ actually has a slightly lower final MSE for 7dB AS noise, this small advantage is outweighed by its higher volatility at earlier times.

Table 4.2: Best choices of cost function and initial variance $q_{n,0}$ when estimating σ_n^2 (or $\sigma_{v_n}^2$) with an EKF using known weights w and process noise statistics.

	White Stat.		AR-5 Stat.	
SNR	Cost	$q_{n,0}$	Cost	$q_{n,0}$
0 dB	$J^{ml}(\sigma_n^2)$	.1	$J^{ml}(\sigma_{v_n}^2)$	.01
3 dB	$J^{ml}(\sigma_n^2)$	.1	$J^{ml}(\sigma_{v_n}^2)$	.01
7 dB	$J^{ml}(\sigma_n^2)$	.1	$J^{ml}(\sigma_{v_n}^2)$	.01

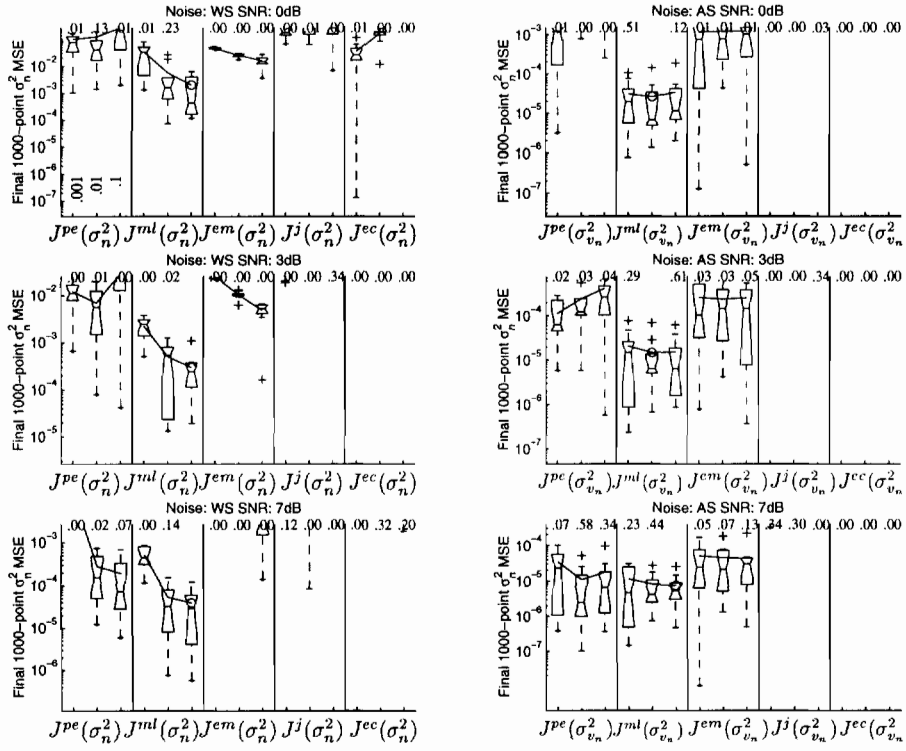


Figure 4.16: MSEs of $\hat{\sigma}^2_{n,k}$ on white noise (left) and $\hat{\sigma}^2_{v_n,k}$ on colored noise (right), computed over the final 1000 points. For each of the six different test conditions, boxplots are shown for $q_{n,0}$ increasing as $\{.001, .01, .1\}$ from left to right within each panel; each panel represents a different cost function, as indicated. See the caption of Figure 4.10 for additional information.

4.5.3 Estimating Both Noise Variances

When neither the process noise variance σ_v^2, nor the measurement noise variance σ_n^2 are known, they must be estimated simultaneously from the data. Because there is a strong interaction between the estimation of the two variances, it is not necessarily true that combining the best individual estimation methods for σ_v^2 and σ_n^2 will produce the best result in the present case.

The problem is that some treatments for estimating one of the parameters might be sensitive to errors in the other parameter, and will produce poor results if that parameter is not known exactly. However, rather than exhaustively search the entire space of cost functions and values of q_0, it is reasonable to consider combinations of only the best 3 treatments for each of the variances. This means a search space of nine possibilities, where the possibilities will be different for each of the data sets, because the previously chosen best three treatments depend on the noise type and SNR.

Just as when estimating the measurement noise variance alone, the exact model $\mathbf{w}_n$ of the AR-5 noise is used, and initial estimates $\hat{\sigma}^2_{v_n,0}$ are obtained from a 500-point segment of the noise.

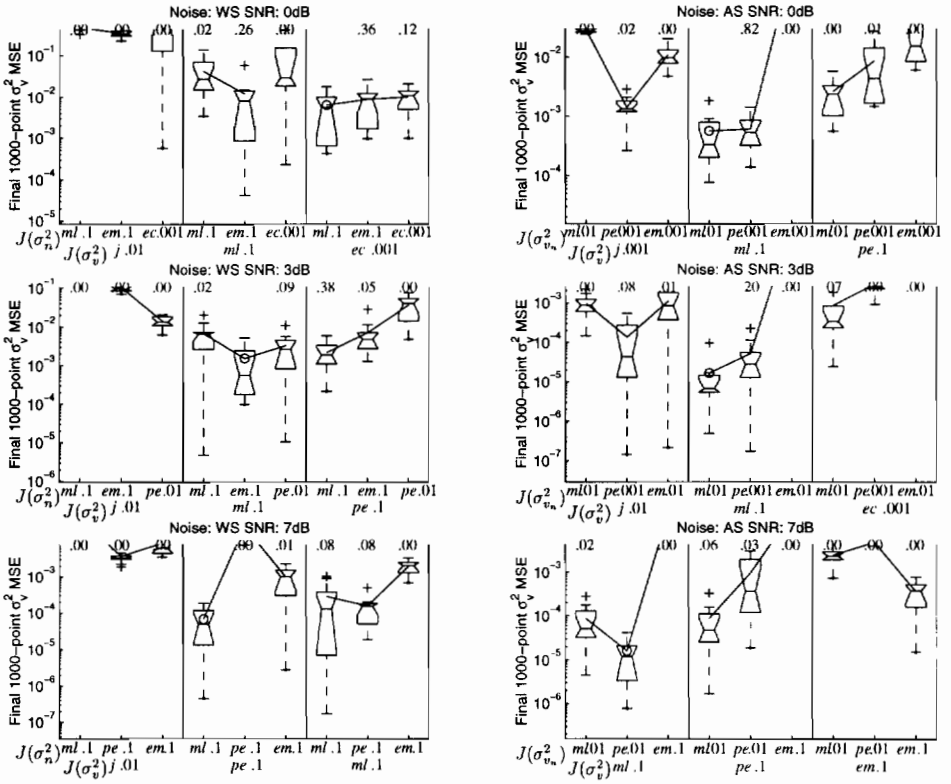


Figure 4.17: Boxplots of the $\hat{\sigma}^2_{v,k}$ MSEs computed over the final 1000 points are shown for the combinations of the top 3 cost-q_0 treatments for estimating σ^2_v and σ^2_n. The σ^2_n estimation methods vary within each panel each panel, and σ^2_v costs vary across panels, as indicated.

In the white noise case, $\hat{\sigma}^2_{n,0}$ is estimated *via* the ad hoc procedure described in Section 3.6.2 on page 104. Using these initial estimates of the measurement noise variance, the process noise variance, σ^2_v, is also initialized by the ad hoc procedure. However, when the measurement noise is colored, the estimates of the autocorrelation $\mathbf{R}_{nn}$ and cross-correlation $\mathbf{p}_{nn}$ are generally too noisy to produce reliable initial estimates of σ^2_v, which causes $\hat{\sigma}^2_{v,0}$ to be truncated near zero in some instances. Therefore, the white noise versions of these quantities are used in all cases, with $\mathbf{R}_{nn} = \hat{\sigma}^2_{n,k}\mathbf{I}$ and $\mathbf{p}_{nn} = 0$.

The previous pages argued that it is sensible to rank the σ^2_v estimation methods using the final 1000-point MSE in σ^2_v, and to rank the σ^2_n estimation methods using the final 1000-point MSE in σ^2_n. However, it is less clear how to rank algorithms for estimating both σ^2_v and σ^2_n. While signal estimation MSE is a possible criterion, it does not produce the required amount of resolving power for comparing methods. A reasonable solution is to choose the σ^2_v and σ^2_n methods separately using their respective error criteria as before, but then discard choices which result in a bad interaction. Although this approach may sound somewhat vague, it is clarified in the discussion of the results,

below.

For these experiments, the data length is increased to $N = 20,000$ points, to allow additional time for convergence. The 1000-point MSEs of the σ_v^2 estimates are presented in Figure 4.17. Each plot is divided into three panels that correspond to the best three σ_v^2 estimation treatments, ordered in decreasing performance from left to right, as determined when σ_n^2 was known. Within each panel, the σ_n^2 estimation methods are shown in order of decreasing performance from left to right, as determined when σ_v^2 is known.

On all of the data sets, the first panel shows the worst performance, indicating that the ranking determined with known σ_n^2 no longer holds. On the 0dB WS noise, the $J^{ec}(\sigma_v^2)$ cost (third panel) appears to be the best, although it is not significantly better than using $J^{ml}(\sigma_v^2)$ with $J^{em}(\sigma_n^2)$. On the 3dB WS noise, the middle panel contains the best treatment, but there is no significant difference between $J^{ml}(\sigma_v^2)$ with $J^{em}(\sigma_n^2)$ in the middle panel and $J^{pe}(\sigma_v^2)$ with $J^{ml}(\sigma_n^2)$ in the third panel. The boxplots and displayed p-values are important for reconciling this performance criterion with that based on error in $\hat{\sigma}_n^2$.

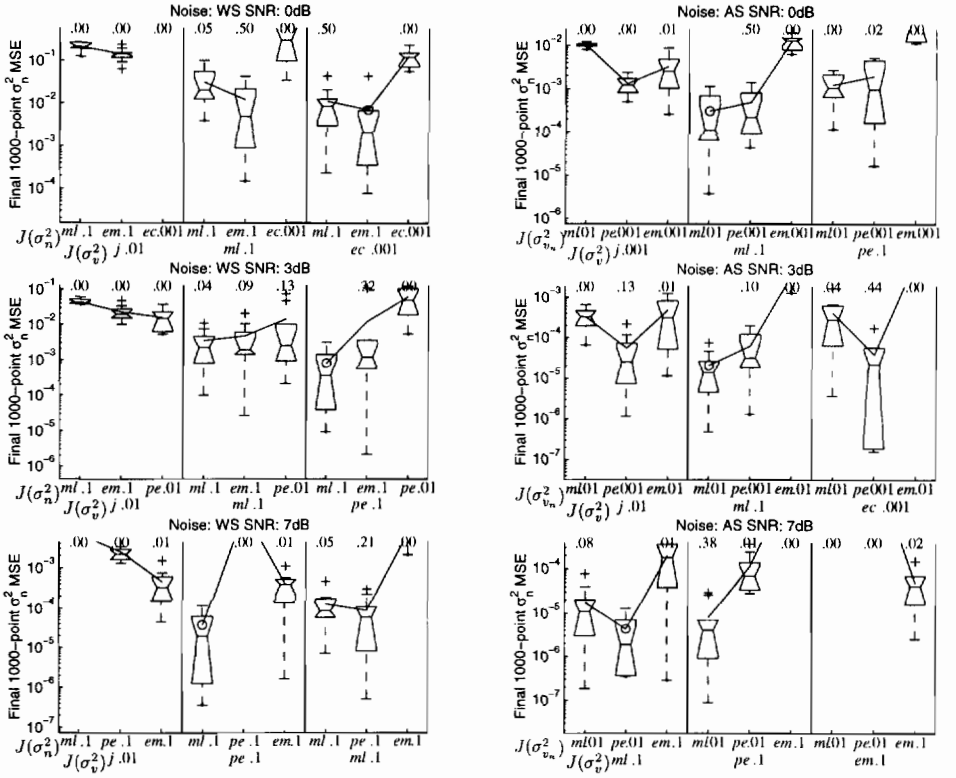


Figure 4.18: Boxplots of the $\hat{\sigma}_{n,k}^2$ MSEs computed over the final 1000 points are shown for the combinations of the top 3 cost-q_0 treatments for estimating σ_v^2 and σ_n^2. The σ_n^2 estimation methods vary within each panel each panel, and σ_v^2 costs vary across panels, as indicated.

Figure 4.18 shows box plots for the 1000-point MSEs of the σ_n^2 (or $\sigma_{v_n}^2$) estimates. The plots are organized the same way as in Figure 4.17; the σ_n^2 methods are listed in order of decreasing performance within each panel. A significant deviation from the original ranking is again evident. For example, for 0dB WS noise, the $J^{em}(\sigma_n^2)$ $q_0 = .1$ treatment performs better than the top-ranked $J^{ml}(\sigma_n^2)$ treatment in all three panels.

To reconcile this result with that in Figure 4.17, notice that the $J^{ec}(\sigma_v^2)$-$J^{em}(\sigma_n^2)$ treatment is not significantly worse than the first choice $J^{ec}(\sigma_v^2)$-$J^{ml}(\sigma_n^2)$ treatment in that figure. It is therefore reasonable to select $J^{ec}(\sigma_v^2)$-$J^{em}(\sigma_n^2)$ as the best treatment, although $J^{ml}(\sigma_v^2)$-$J^{em}(\sigma_n^2)$ and $J^{ec}(\sigma_v^2)$-$J^{ml}(\sigma_n^2)$ are also a good choices.

On the 3dB SNR white noise, $J^{ml}(\sigma_n^2)$ with $q_0 = .1$ retains its top ranking (although not in the first panel). Here, too, the results for the two criterion must be reconciled. This time, the $J^{pe}(\sigma_v^2)$-$J^{ml}(\sigma_n^2)$ is indicated in Figure 4.18, while $J^{ml}(\sigma_v^2)$-$J^{em}(\sigma_n^2)$ is indicated in Figure 4.17. Again the p-values can be used to justify choosing $J^{pe}(\sigma_v^2)$-$J^{ml}(\sigma_n^2)$.

Fortunately, on the remainder of the plots, the optimal choices according to σ_v^2 error and σ_n^2 error coincide. The best treatments for each of the 6 noise cases are shown in Table 4.3.

Table 4.3: Best choices of cost functions and initial variances q_0 when estimating both σ_v^2 and σ_n^2 with EKFs, using known weights $\mathbf{w}$.

	White Stat.			AR-5 Stat.		
SNR	Costs	$q_{v,0}$	$q_{n,0}$	Costs	$q_{v,0}$	$q_{v_n,0}$
0 dB	$J^{ec}(\sigma_v^2)$ $J^{em}(\sigma_n^2)$	.001	.1	$J^{ml}(\sigma_v^2)$ $J^{ml}(\sigma_{v_n}^2)$	.1	.01
3 dB	$J^{pe}(\sigma_v^2)$ $J^{ml}(\sigma_n^2)$	.1	.1	$J^{ml}(\sigma_v^2)$ $J^{ml}(\sigma_{v_n}^2)$	.1	.01
7 dB	$J^{pe}(\sigma_v^2)$ $J^{ml}(\sigma_n^2)$	.1	.1	$J^{ml}(\sigma_v^2)$ $J^{pe}(\sigma_{v_n}^2)$	.1	.01

4.6 Experiment 3: Forgetting Factor

Section 3.3.2 on page 57 introduced a scalar term, λ, which is used during weight and variance estimation to control the length of an exponentially decaying window over the data. The term is called a "forgetting factor" because, for values less than 1, it prevents data in the more distant past from being used to estimate the parameters: they are "forgotten."

In the dual estimation context, a forgetting factor is useful even when the underlying system and noise terms are entirely stationary. The justification is mostly heuristic: the signal estimates $\hat{x}_k$ at small times k are inaccurate, so they should not influence the parameter estimation as much as recent, more accurate signal estimates.

When $\lambda = 1$, the filter will eventually converge to its final solution as k approaches infinity.

However, this convergence will generally be slower than necessary because of the inertia of the initial signal estimates. Conversely, while using $\lambda < 1$ can increase the converge rate, the advantage disappears at large values of k, and may eventually turn into a disadvantage because of a higher variance in the parameter estimates.

4.6.1 Stationary AR-5 Noise

The results described below are obtained by running the dual EKF algorithm on the neural network signal, corrupted by stationary AR-5 noise at 3dB SNR. Initial error covariances are $\mathbf{P}_0 = \mathbf{I}$ and $\mathbf{Q}_0 = .1\mathbf{I}$. The joint cost $J^j(\mathbf{w})$ is used for weight estimation, and the maximum-likelihood cost J^{ml} is used to estimate both of the variances, using $q_{v,0} = .1$ and $q_{v_n,0} = .01$. Figure 4.19 contains boxplots for the signal estimation MSE, computed over all the data, as well as over the first 100 and last 1000 points.

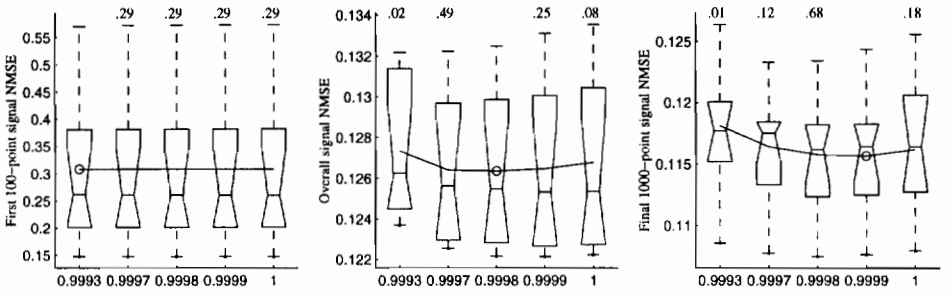


Figure 4.19: Boxplots of the $\hat{x}_k$ NMSEs computed over 3 different ranges of data are shown for 5 values of λ.

At very small times, few data are available, and there is no significant difference between different values of λ. As more data become available, evidence emerges of the tradeoff between too much and too little flexibility. In terms of the overall MSE, the value of $\lambda = 0.9993$ is apparently too small, whereas using $\lambda = 1$ is also disadvantageous. At larger times, the disadvantage of $\lambda = 1$ is reduced, because enough data has been observed to dilute the effects of the early estimates. Values of 0.9998 and 0.9999 appear to give the best results, as shown by the MSEs computed over the last 1000 points in the third panel of the figure.

The situation is clarified somewhat by Figure 4.20, which makes use of the time-averaged MSE profiles described on page 120. The left plot shows the average difference between the MSE profile for the known-model EKF result, and the MSE profile for each of three different forgetting factors. This plot readily shows the convergence behavior as a function of time, but discerning the difference between values of λ is difficult. In the right plot the difference is instead computed between each of the MSE profiles, and the MSE profile using $\lambda = 1$. The problem with using

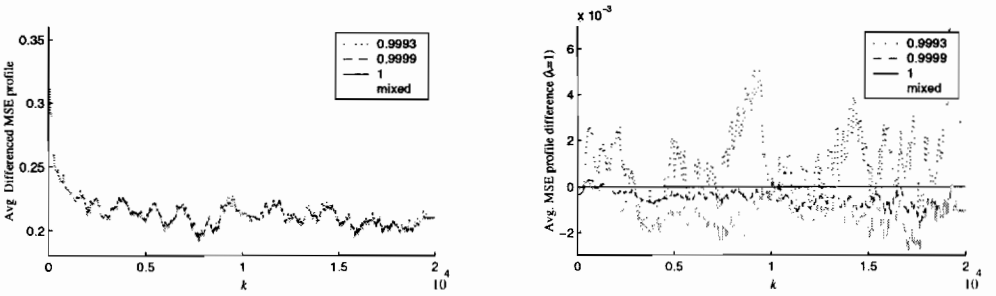


Figure 4.20: MSE profiles of signal estimation error. At 50-point intervals MSEs are computed over the next 500 points to create the profiles. The left graph shows the difference between each profile and the EKF profile. The right graph shows the difference between the $\lambda = 1$ profile and each of the others.

too small a value of λ is clearly shown by the highly volatile plot for $\lambda = 0.9993$. The advantage of using $\lambda = 0.9999$ instead of 1 is also evident; the line labelled "mixed" will be explained in subsequent paragraphs.

Figure 4.21 shows the effect of λ on estimation of the noise variances. Because these are scalar values, much less data are required to produce reliable estimates, as reflected in the preference for $\lambda = .9993$ shown in the boxplots of the final 1000-point MSEs of both $\hat{\sigma}^2_{v,k}$ and $\hat{\sigma}^2_{v_n,k}$. The effect of λ on the trajectories of these estimates is seen in the bottom two plots of the ensemble-averages

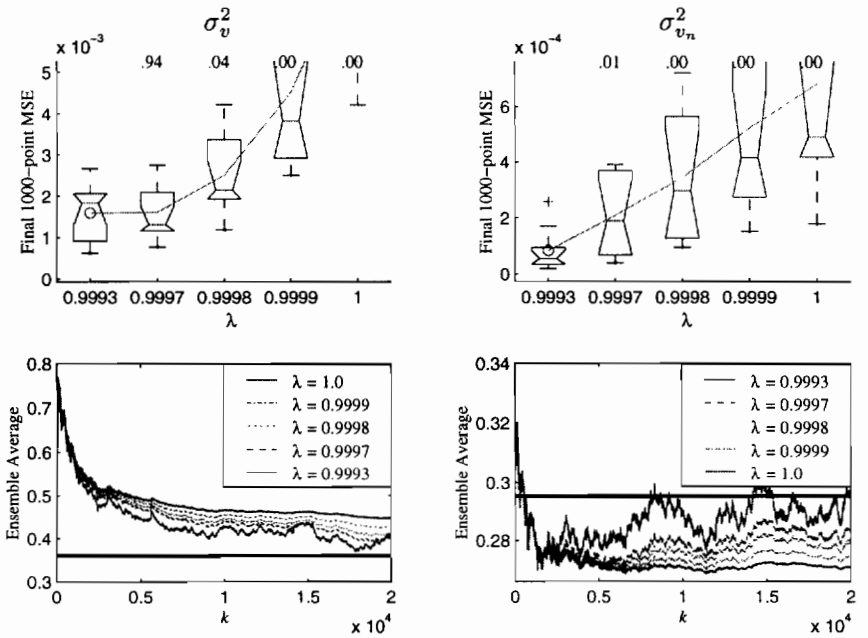


Figure 4.21: Boxplots show the final 1000-point MSEs for $\hat{\sigma}^2_{v,k}$ (left) and $\hat{\sigma}^2_{v_n,k}$ (right). Also shown are the ensemble-averaged trajectories of the variance estimates; heavy horizontal lines indicate the true variance values.

$\hat{\sigma}^2_{v,k}$ and $\hat{\sigma}^2_{v_n,k}$. The true value of each variance is indicated for reference by a horizontal line.

The clear difference in the optimal values for λ used in weight estimation and variance estimation, respectively, justifies using a separate value for each. To test this approach, $\lambda_{\mathbf{w}} = 0.9999$ is used in the weight filter, and $\lambda_{\sigma^2} = 0.9993$ is used in each of the variance filters. The resulting MSE profile is shown in Figure 4.20 and labelled as "mixed"; the overall MSE is significantly improved over the pure $\lambda = .9999$ treatment.

4.6.2 Nonstationary White Noise

Dual estimation can also be applied to nonstationary data. By corrupting the neural network signal with the sinusoidally modulated white noise described on page 128, the relative effects of lag misadjustment and noise misadjustment can be observed. The dual EKF was used with costs $J^{pe}(\mathbf{w})$, $J^{pe}(\sigma_v^2)$, and $J^{ml}(\sigma_n^2)$, and initial error covariances $\mathbf{Q}_0 = .1\mathbf{I}$, $q_{v,0} = .1$, and $q_{n,0} = .1$, to produce the results in Figure 4.22.

The signal is stationary, but the noise has a continuously changing variance. Hence, the forgetting factor $\lambda_{\mathbf{w}}$ is fixed at 0.9999 for estimation of the weights, while λ_{σ^2} is varied across five values. The bottom right plot (showing average trajectories of $\hat{\sigma}^2_n$) shows that a value of $\lambda_{\sigma^2} = 1$ produces a damping effect, with $\hat{\sigma}^2_{n,k}$ tending towards a flat line at the DC level of the

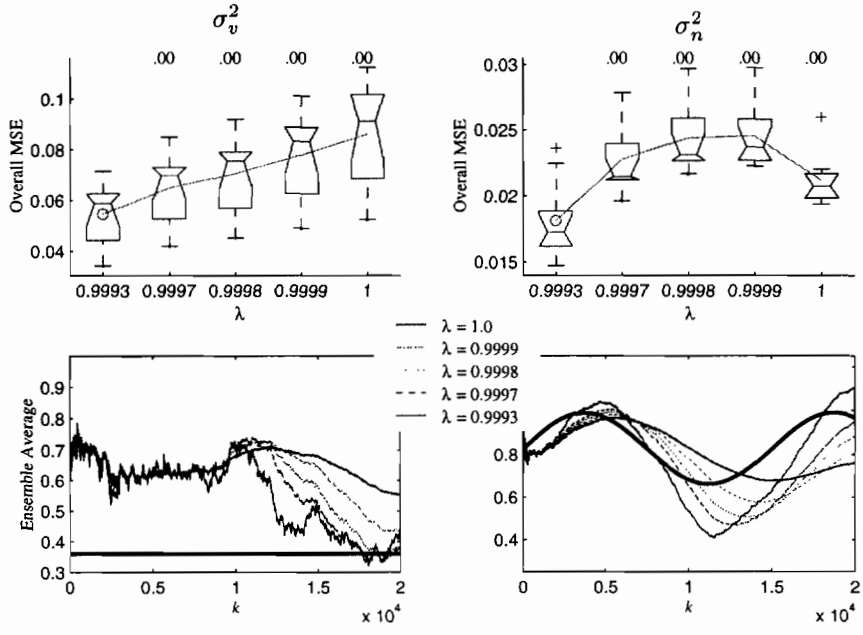


Figure 4.22: Boxplots show the overall MSEs for $\hat{\sigma}^2_{v,n}$ (left) and $\hat{\sigma}^2_{v_n,k}$ (right). At bottom are the ensemble-averaged trajectories of the variance estimates, shown along with the true variances for reference (heavy lines). For weight estimation, $\lambda = 0.9999$ in all cases.

sinusoid, as $k \to \infty$. Values less than 1 show a better ability to track σ_n^2, with smaller values of λ_{σ^2} exhibiting less phase lag. Unfortunately, these same λ_{σ^2} values suffer from higher noise misadjustment because they are based on fewer data points.

The choice of λ_{σ^2} for use on nonstationary noise is clearly data-dependent, as it depends largely on the rate of nonstationarity. This experiment merely highlights the tradeoffs that must be considered between lag misadjustment and noise misadjustment. It also indicates that the sinusoidally varying noise level used here changes too rapidly for the tracking speed of the algorithm.

As a rule of thumb for choosing λ, the time-constant $\tau = -1/\log(\lambda)$ of the forgetting window should be approximately the same as N_{ns}, the length of an approximately stationary section of the data (see Section 3.6.3). However, if N_{ns} is too short, then the algorithm will not be able to track the changing system, and some sort of iterative, windowed processing must be employed.

4.7 Experiment 4: Dual Kalman Weight Costs

With reasonable values for the initial covariances and forgetting factor in place, the cost functions for weight estimation can be compared within the dual Kalman framework. A forgetting factor of $\lambda_{\mathbf{w}} = .9999$ is used for weight estimation, and initial error covariances of $\mathbf{P}_0 = \mathbf{I}$, and $\mathbf{Q}_0 = .1\mathbf{I}$ are used in most cases (a smaller value of Q_0 is sometimes required for convergence). Results are first generated with the noise variances known, and then with one or both of these variances estimated on-line, at the same time as the signal and weights.

Various combinations of the signal and noise types described in Sections 4.2.3 and 4.2.4 are used in the experiments. Ten repetitions of the noise are used to generate boxplots and p-values. Appropriate model structures are chosen for each data set: for the linear and neural network signals, the exact model structures used to generate the data are used by the dual EKF. In the case of the Ikeda signal, a feedforward network with 10 inputs, 8 hidden units, and one output (10-8-1) is used; a linear AR model with $M_n = 10$ is used for the pink noise.

4.7.1 Known Variances

In this set of experiments, the true process noise variance σ_v^2 is assumed known, as are the measurement noise statistics (either σ_n^2, or $\mathbf{w}_n$ and $\sigma_{v_n}^2$). As an example, Figure 4.23 shows the dual EKF estimation of the chaotic neural network signal in 3dB AS noise, using the $J^j(\mathbf{w})$ cost. The estimates are indicated by the heavy curve, the noisy data are shown by '+' signs, and the clean signal appears as a thin curve.

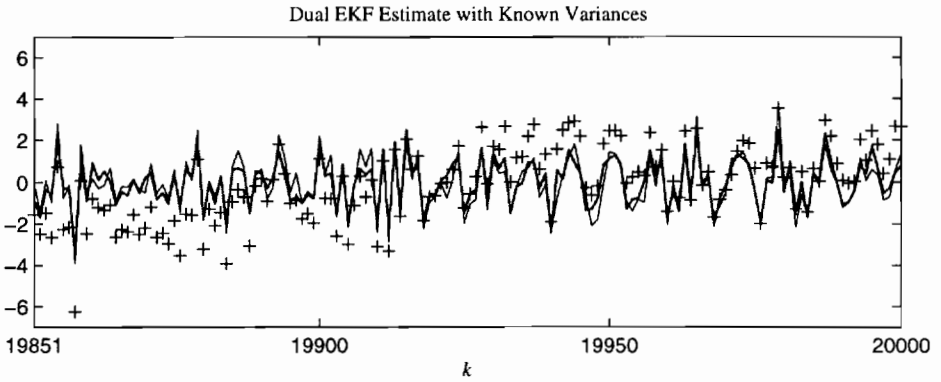


Figure 4.23: Example of dual EKF estimation of nonlinear time-series in 3dB colored noise, using the joint cost function and known variances. Only the last 150 points are shown.

In the following experiments, the weight costs are tested on six combinations of signal and noise, each at four different SNRs.

1. Linear AR-10 data corrupted by white stationary (WS) noise.
2. Limit cycle neural network data corrupted by WS noise.
3. Limit cycle neural network data corrupted by stationary AR-5 (AS) noise.
4. Chaotic neural network data corrupted by AS noise.
5. Chaotic neural network data corrupted by nonstationary AR-5 (AN) noise.
6. Normalized Ikeda data corrupted by stationary pink (PS) noise.

The boxplots in Figure 4.24 show the relative performances of the different cost functions on the linear AR-10 data in stationary white measurement noise. The performances of the $J^{ml}(\mathbf{w})$ and $J^{pe}(\mathbf{w})$ costs are very similar, although the former shows some advantage in terms of final 1000-point signal NMSE. Because the model is linear, trajectories of the MSE in the weight estimates can be plotted. Figure 4.25 shows that at low SNRs, $J^{pe}(\mathbf{w})$ provides faster convergence; although not shown, this results in a lower overall signal MSE than $J^{ml}(\mathbf{w})$. Furthermore, $J^{pe}(\mathbf{w})$ is somewhat more robust at 0dB SNR. In terms of signal NMSE, the $J^{j}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ costs perform significantly worse at all noise levels[3]. On the other hand, $J^{j}(\mathbf{w})$ produces results with much less spread in the weight MSE than the other methods; the bias in these weight estimates, however, translates to larger signal estimation NMSEs. Note that this result confirms the qualitative analysis

[3]In the context of the dual Kalman filter, we will use $J^{j}(\mathbf{w})$ as shorthand for the direct substitution joint cost, $J^{j}(\hat{\mathbf{x}}_1^k, \mathbf{w})$; the use of signal estimates is implied.

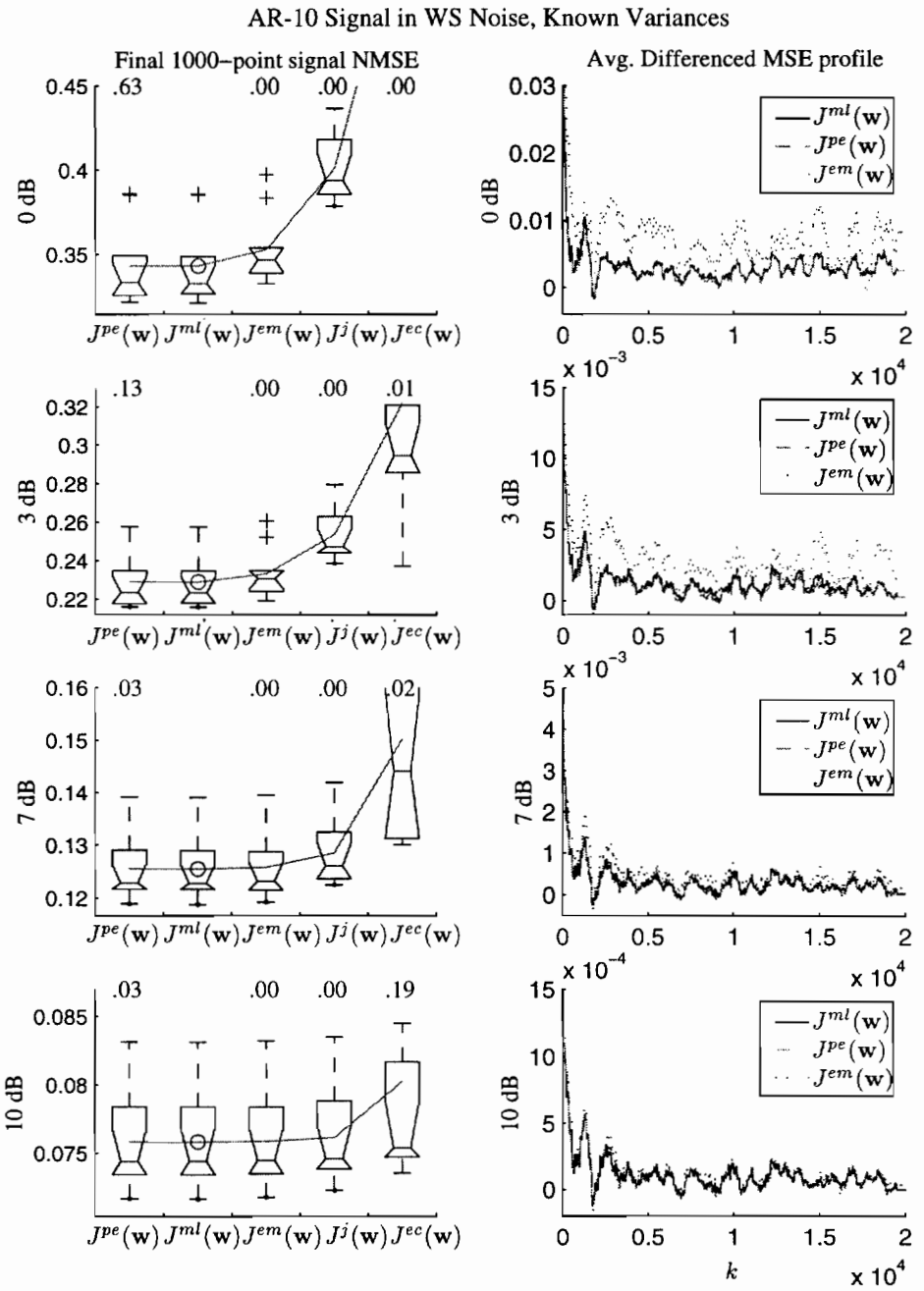


Figure 4.24: AR-10 data corrupted by white stationary noise at four different SNRs. On the left, boxplots show the final 1000-point NMSEs for the signal estimates. On the right, the average differenced MSE profiles are shown.

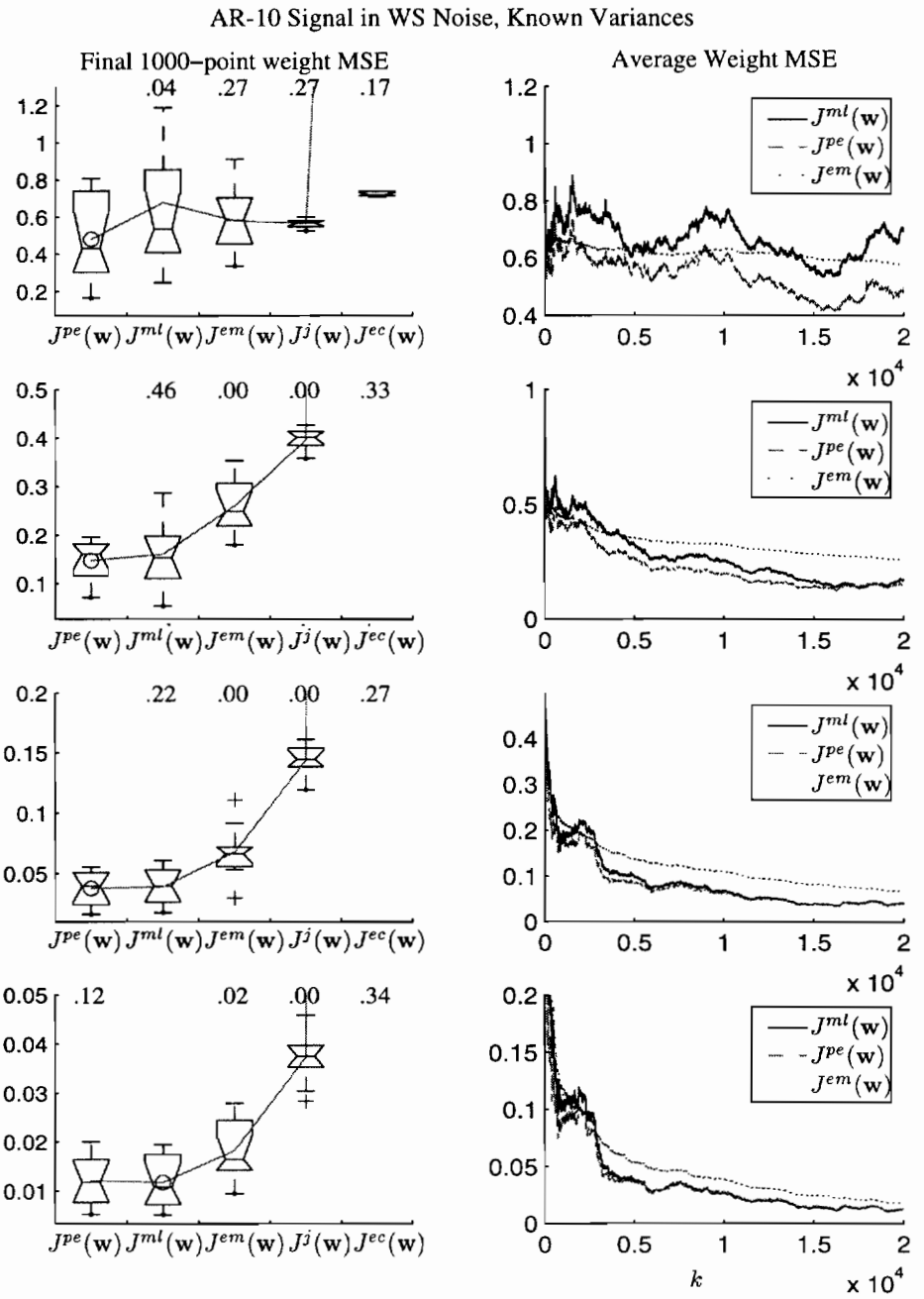


Figure 4.25: Weight estimate MSE trajectories for AR-10 data corrupted by white stationary noise at four different SNRs.

of the joint cost at the end of Chapter 2. The poor performance of the error-coupled joint cost can be interpreted in a similar way, but with an even larger bias than the $J^j(\mathbf{w})$ cost. Meanwhile the EM cost exhibits less volatility in the weight MSE trajectory, at the expense of slower convergence speed.

Similar results are obtained when the stationary white noise is used to corrupt the limit cycle neural network data, as shown in Figure 4.26 on the following page. As an aside, note that the NMSE levels for the limit cycle data are a factor of between 2 and 7 smaller than for the linear AR-10 data. This outcome is surprising because the nonlinear model should involve more approximations. However, it can be explained by the fact that the limit cycle data are more deterministic because its process noise variance, σ_v^2, is smaller than that of the AR-10 series (.04 *.vs* .09); the increased predictability of the series makes it inherently easier to estimate. A side-effect of a smaller value of σ_v^2 is that some of the algorithms encounter stability problems when using $Q_0 = .1$. Hence, $Q_0 = .01$ is used on this series instead. Recall from the discussion on page 62 that for small σ_v^2, $\mathbf{Q}_0$ needs to be smaller to keep $(\mathbf{Q}_1)^{-1}$ (and subsequent updates) invertible. The cost functions most prone to instability with $Q_0 = .1$ are $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$; coincidentally, these are the only costs that were not evaluated in Experiment 1 when determining the best choice of Q_0! However, the EM cost also shows a slight improvement in performance with $Q_0 = .01$, as well.

Figure 4.26 shows that at the lower two SNRs, $J^{pe}(\mathbf{w})$ and $J^{ml}(\mathbf{w})$ are not significantly different, while $J^{ml}(\mathbf{w})$ shows a small (but significant) advantage at higher SNRs. While the joint cost $J^j(\mathbf{w})$ performs better than in the linear model case, it is nonetheless significantly worse than $J^{pe}(\mathbf{w})$ at any SNR. Meanwhile, $J^{em}(\mathbf{w})$ performs very poorly at low SNRs, and is adequate only at the 10dB level. This implies the EM cost might be more sensitive to the EKF approximation required for nonlinear models (see Appendix D).

Figure 4.27 on page 154 shows the results on the limit cycle data in stationary AR-5 noise. Again, $Q_0 = .01$ is used to ensure stability; this drastically improves the performance of $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$, but has little effect on the other costs. At 0dB and 3dB SNR, the maximum-likelihood cost shows the best performance, although $J^j(\mathbf{w})$ is not significantly different at higher SNRs. Meanwhile, $J^{pe}(\mathbf{w})$ and $J^{em}(\mathbf{w})$ are completely unacceptable, and $J^{ec}(\mathbf{w})$ has unstable performance on at least one data repetition at 3dB.

Instability in a modified-Newton algorithm generally can occur when the approximate Hessian (represented by $\mathbf{Q}_k^{-1}$ in the dual EKF) becomes ill-conditioned, resulting in numerical problems during its inversion. This situation can arise when Q_0 is too large, as described above, or when the cost function surface changes much more rapidly in some parameter directions than in others. Hessian singularity problems in the context of maximum-likelihood estimation are discussed in [26].

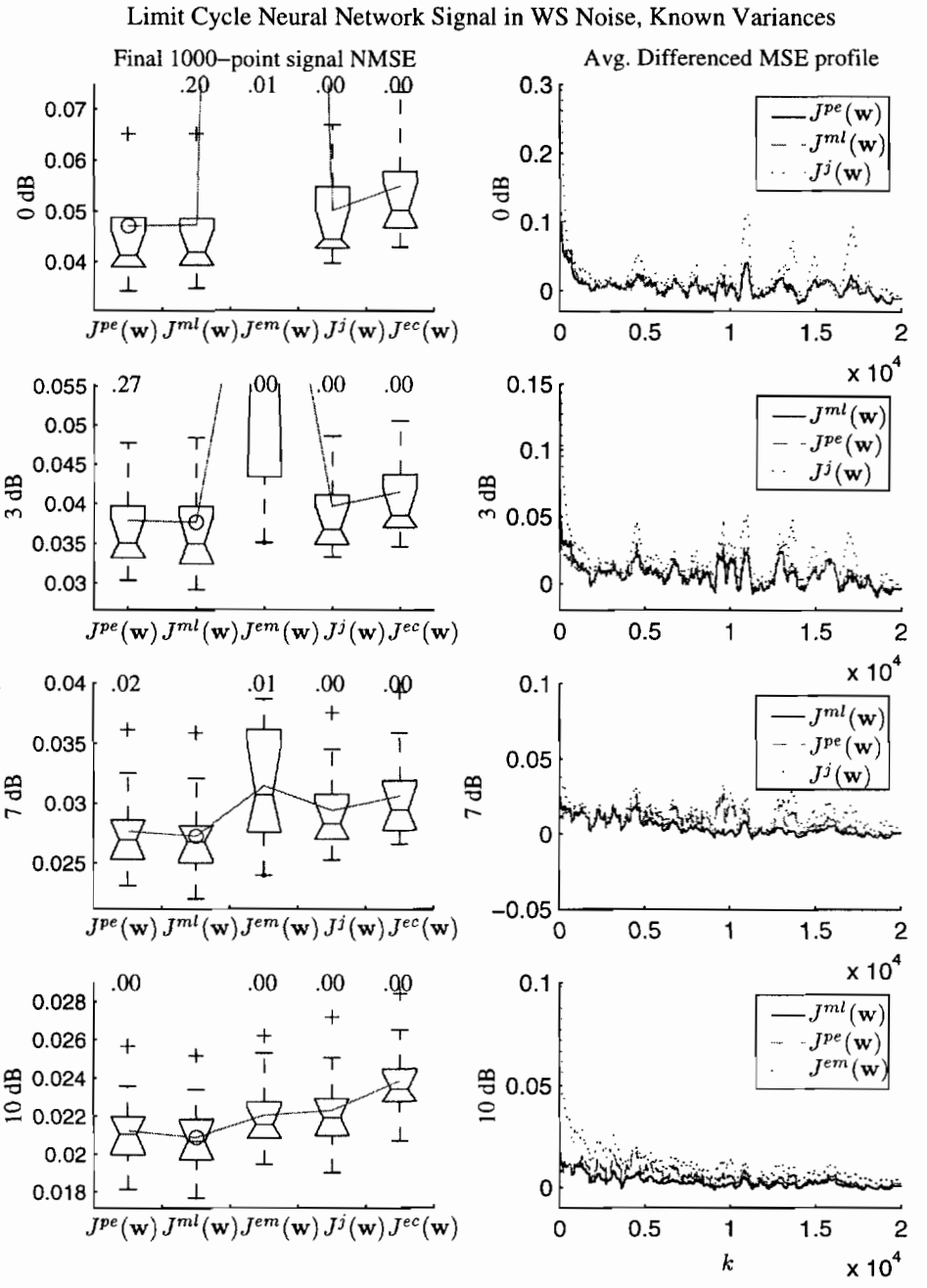


Figure 4.26: Limit cycle neural network signal, corrupted by white stationary noise at four different SNRs. On the left, boxplots show the final 1000-point NMSEs for the signal estimates. On the right, the average differenced MSE profiles are shown.

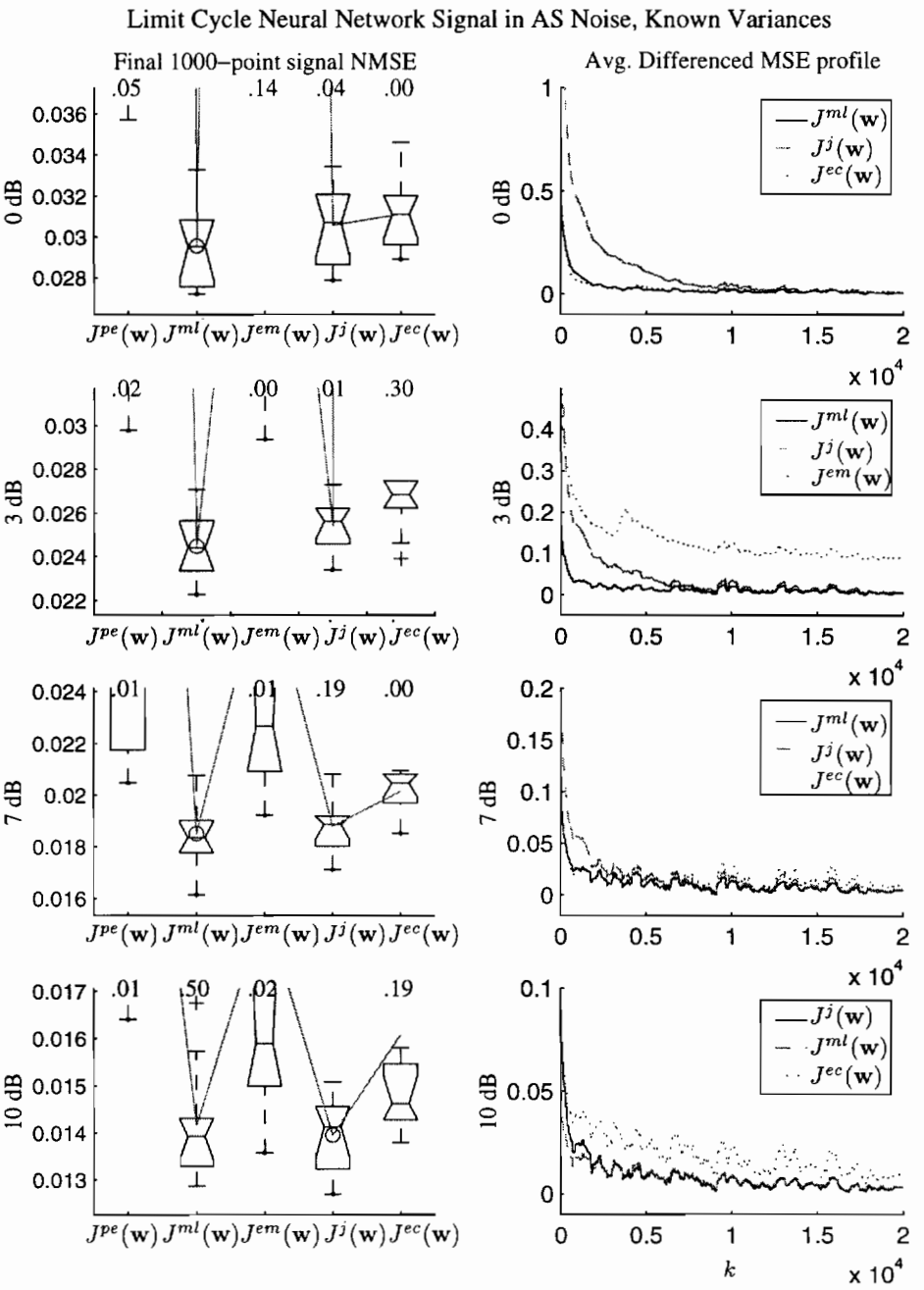
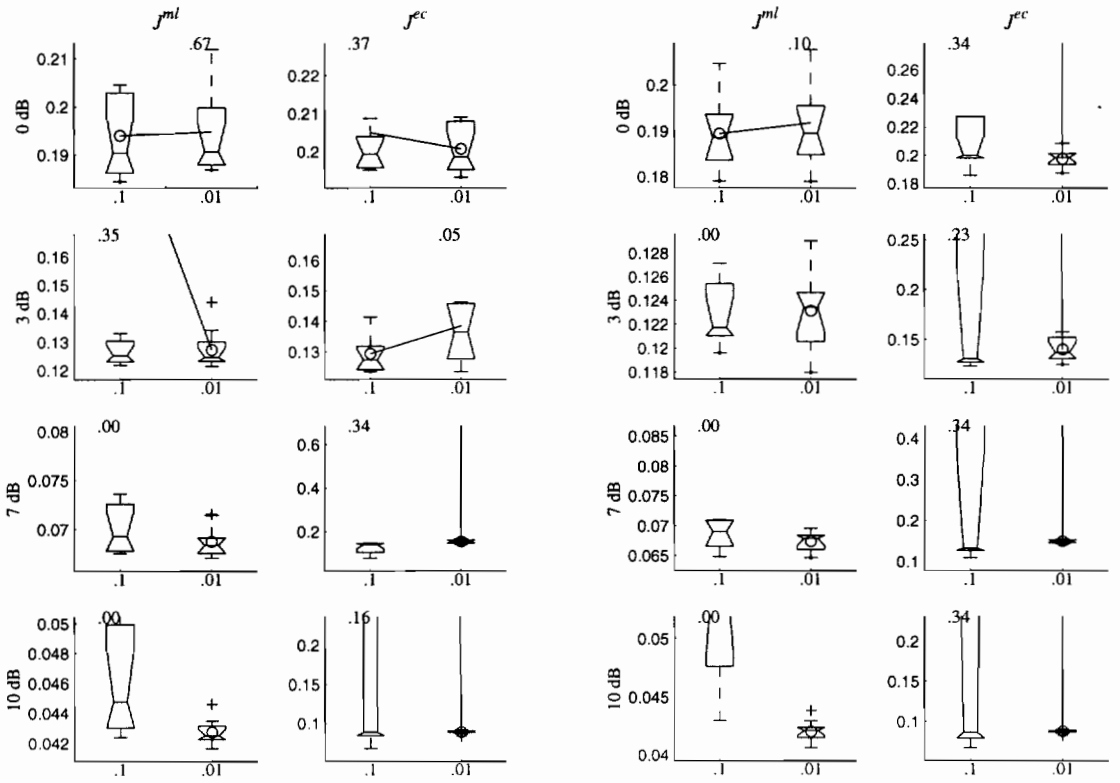


Figure 4.27: Limit cycle neural network data corrupted by stationary AR-5 noise at four different SNRs. On the left, boxplots show the final 1000-point NMSEs for the signal estimates. On the right, the average differenced MSE profiles are shown.

The instability of $J^{ml}(\mathbf{w})$ at high SNRs and on colored noise is most likely due to the denominator term, $\sigma^2_{\varepsilon_k}$, in the cost function becoming small, which causes the Hessian terms involving $\nabla_{\mathbf{w}}\sigma^2_{\varepsilon_k}$ to grow much larger than the $\nabla_{\mathbf{w}}\varepsilon_k$ terms (see Equation E.38 on page 264). An ill-conditioned Hessian will result if $\nabla_{\mathbf{w}}\sigma^2_{\varepsilon_k}$ is much smaller in some parameter directions than in others. Interestingly enough, the only other cost to exhibit stability problems is $J^{ec}(\mathbf{w})$, which has similar terms – involving $\nabla_{\mathbf{w}}g_k$ – in its Hessian (see Equation E.22). Instability of the maximum-likelihood cost is more likely to occur on colored noise, and in particular when the noise variances are small (see Equations 3.165 and 3.207).

However, stability can be usually be restored for the $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ costs by selecting a smaller value of Q_0, as was done for the limit cycle data. The effect of Q_0 on the stability of the $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ costs is investigated in Figure 4.28 using the chaotic neural network data



(a) Stationary AR-5 noise

(b) Nonstationary AR-5 noise

Figure 4.28: Effect of Q_0 on stability of dual EKF for the $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ costs. Boxplots show the overall signal NMSE when estimating the chaotic neural network signal in (a) AS noise and (b) AN noise.

in AS and AN noise. The figure shows boxplots of the overall signal NMSEs, and indicates that while both costs show instability using $Q_0 = .1$, both are stable with $Q_0 = .01$. The algorithms are generally unstable on only a few repetitions of the data, as evinced by the large slope on the line connecting the average NMSE values, or (with $J^{ml}(\mathbf{w})$) the presence of "not-a-number" (NaN) results. These NaN results prevent the calculation of an average NMSE, so no line is drawn. Note that for the repetitions which do not go unstable, the larger value of $Q_0 = .1$ can often produce better results because it provides faster convergence of the algorithm. Nonetheless, stability is of paramount importance; a value of $Q_0 = .01$ should clearly be used for these two costs on the chaotic neural network data.

In the remaining experiments, then, the weight covariance is initialized with $Q_0 = .01$ for $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$, and with $Q_0 = .1$ for other cost functions. The results on chaotic neural network data corrupted by stationary AR-5 noise are shown in the boxplots in Figure 4.29. The joint cost $J^{j}(\mathbf{w})$ and maximum-likelihood cost $J^{ml}(\mathbf{w})$ performs significantly better than the other costs at most SNRs. The two costs are generally equivalent in performance, although $J^{ml}(\mathbf{w})$ shows a significant advantage at 0dB SNR.

For the results on the chaotic neural network data corrupted by *nonstationary* AR-5 noise, the NMSE is calculated over all time $k \in [1, N]$ in order to evaluate both convergence and tracking performance of the cost functions. As indicated by the boxplots in Figure 4.30 on page 158, the maximum-likelihood cost $J^{ml}(\mathbf{w})$ is the best choice at all SNRs, although $J^{j}(\mathbf{w})$ shows a p-value of 15 at the 0dB level. At 10dB, an outlier contributes to the p-value of 14.

On the Ikeda data, the "known" value of σ_v^2 is found by training a neural network of the chosen architecture on the clean data. This is an attempt to find a value of σ_v^2 that accounts for the limited modeling capability of the network architecture. Because the true system is purely deterministic, the actual value of σ_v^2 is zero; however, this value is clearly inappropriate for the neural network model, and would in any case lead to instabilities in the Kalman filters. The chaotic Ikeda time-series is very difficult to model, and even more difficult when corrupted by noise. The NMSE values shown in the boxplots of Figure 4.31 on page 159 are therefore significantly greater than on the neural network data. Furthermore, the average MSE profiles are shown instead of their difference against the EKF result; this is because the "known" model result is actually worse than the dual EKF result in most cases. At low SNRs, the maximum-likelihood cost shows the best performance, while the $J^{pe}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ costs perform well at high SNRs. The $J^{ml}(\mathbf{w})$, $J^{em}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ costs all show instability at 7dB SNR. On these data, the use of $Q_0 = .01$ with $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ was not helpful: the costs showed as much instability (although at different SNRs) as with $Q_0 = .1$. This underscores the inherent stability problems of these two costs. On

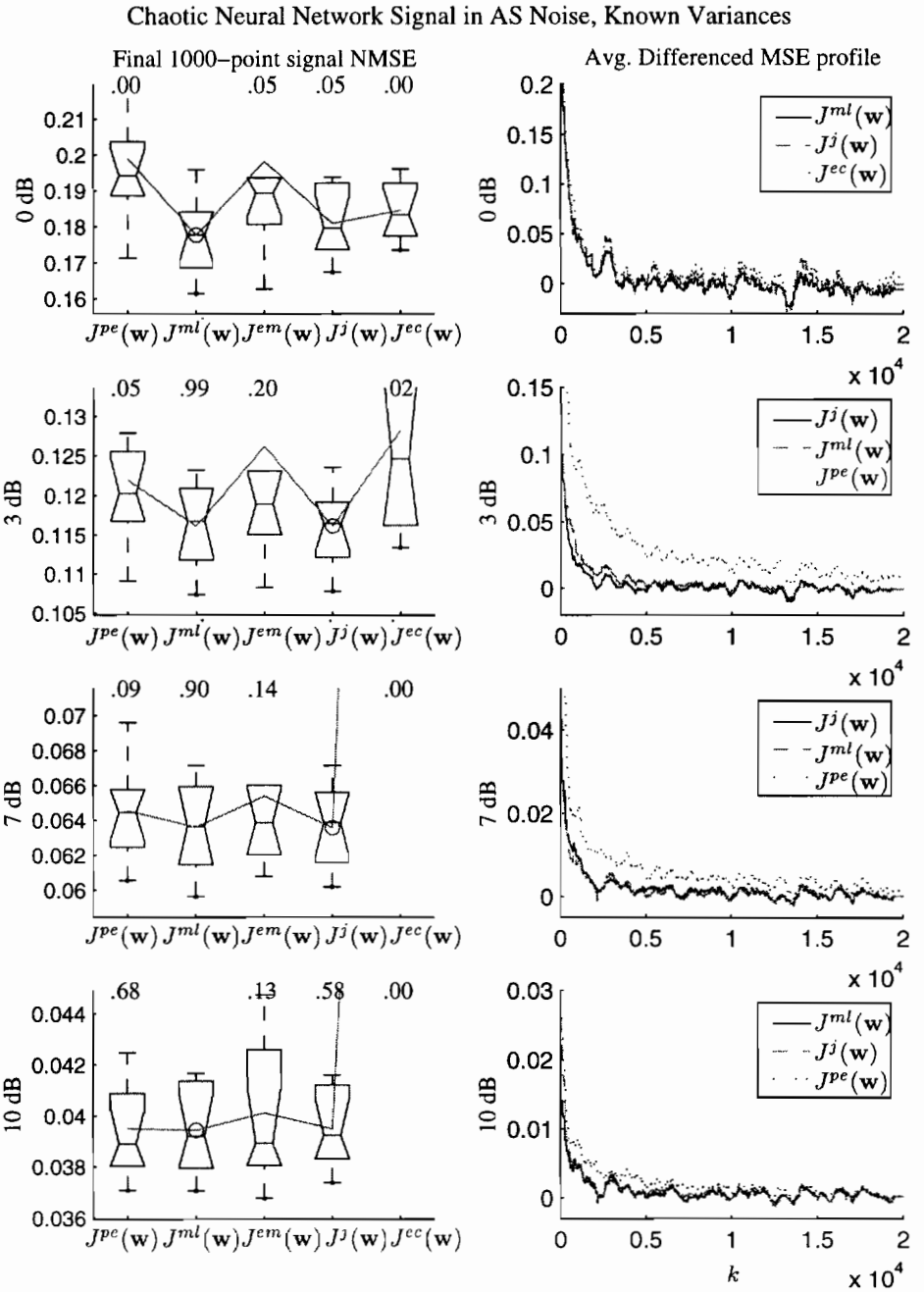


Figure 4.29: Chaotic neural network data corrupted by stationary AR-5 noise at four different SNRs. On the left, boxplots show the final 1000-point NMSEs for the signal estimates. On the right, the average differenced MSE profiles are shown.

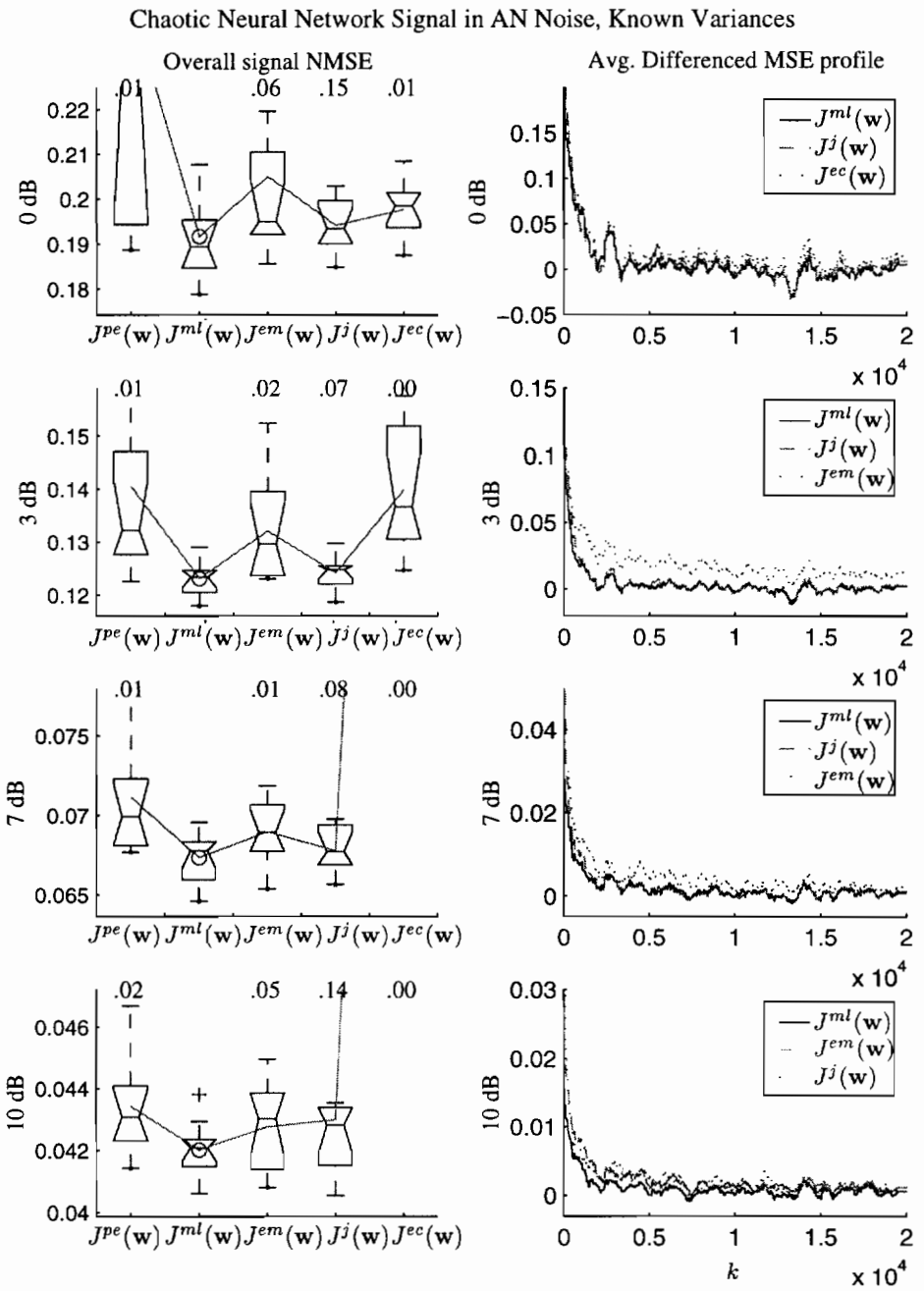


Figure 4.30: Chaotic neural network data corrupted by nonstationary AR-5 noise at four different SNRs. On the left, boxplots show the overall NMSEs for the signal estimates. On the right, the average differenced MSE profiles are shown.

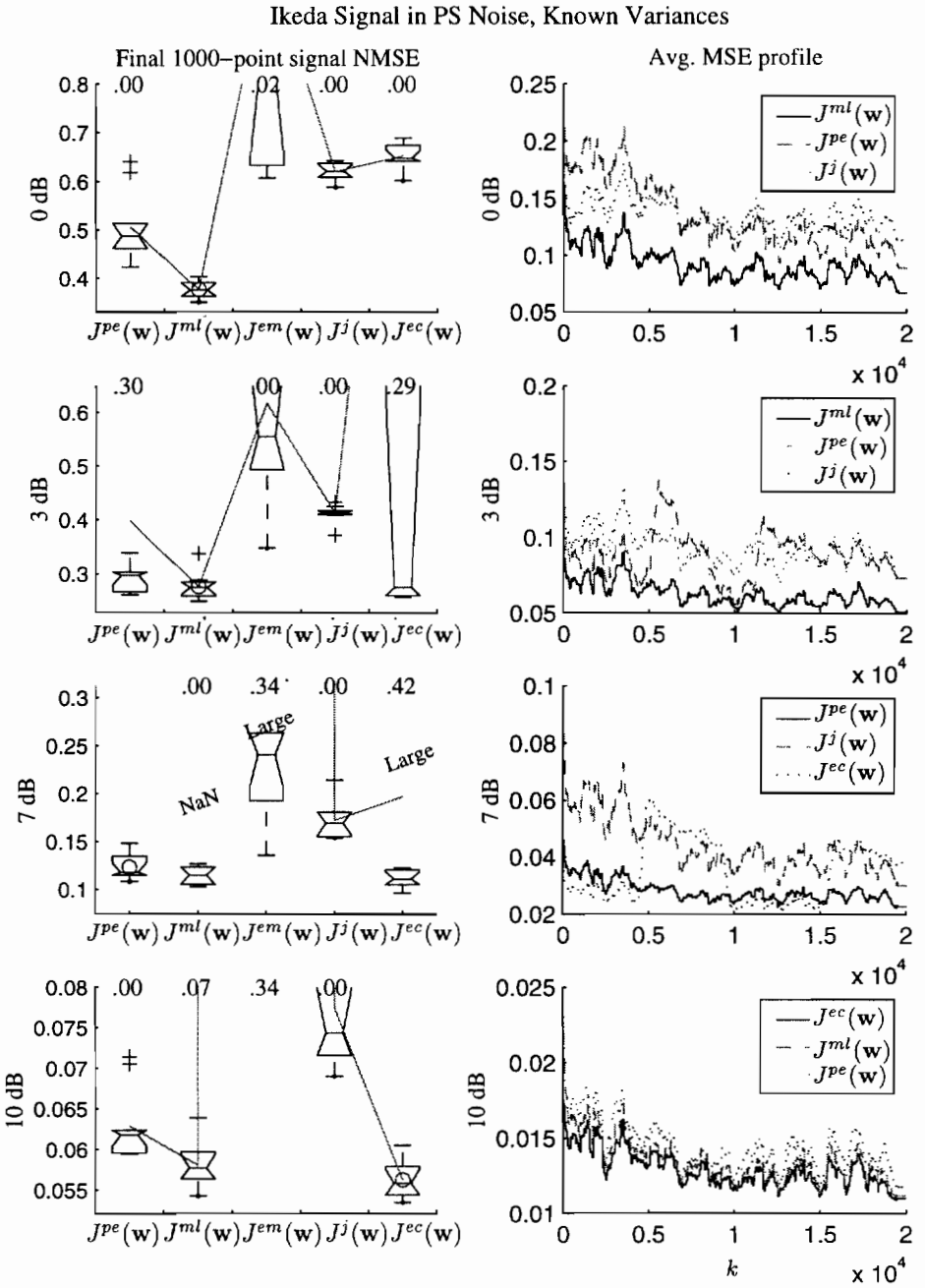


Figure 4.31: Normalized Ikeda data corrupted by stationary pink noise at four different SNRs. Boxplots show the initial, overall, and final NMSEs for the signal estimates.

the other hand, the behavior of $J^j(\mathbf{w})$ is consistently suboptimal, but never unstable. The EM cost generally shows poor performance.

It was hypothesized at the end of Chapter 2 that in comparison to the marginal estimation approaches, the joint cost functions offer the potential for reduced variance in the weight estimates, at the expense of higher bias. However, at that juncture the conditions under which one cost is favored over another could not be predicted. Table 4.4 provides a partial answer to this question by summarizing the best cost functions for the known variance case.

Table 4.4: Best dual estimation cost functions for estimating w when both noise variances are known. Column headings are abbreviations for the four data sets listed on page 149.

SNR	AR-10(WS)	NNlc(WS)	NNlc(AS)	NNch(AS)	NNch(AN)	Ikeda(PS)
0 dB	$J^{pe}(\mathbf{w})$	$J^{pe}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$
3 dB	$J^{pe}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{j}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$
7 dB	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{j}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{pe}(\mathbf{w})$
10 dB	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{j}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ec}(\mathbf{w})$

A few general trends are apparent from the results reported thus far:

1. $J^{ml}(\mathbf{w})$ generally provides the best performance on both white and colored noise.

2. $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ are much more sensitive to σ_v^2 and Q_0 than the other costs, and are more prone to stability problems.

3. $J^{pe}(\mathbf{w})$ can perform better than $J^{ml}(\mathbf{w})$ on low SNR white noise, or when $J^{ml}(\mathbf{w})$ is unstable. Otherwise, it does not perform as well.

4. $J^{em}(\mathbf{w})$ typically shows mediocre or poor performance, although it sometimes is adequate on high SNR data.

5. $J^{j}(\mathbf{w})$ can perform as well as $J^{ml}(\mathbf{w})$ on higher SNR colored noise, and is not as prone to instability. However, $J^{ml}(\mathbf{w})$ does significantly better at low SNR and on white noise.

6. $J^{ec}(\mathbf{w})$, like $J^{ml}(\mathbf{w})$, exhibits stability problems. It generally performs worse than $J^{j}(\mathbf{w})$, with the exception of the 10dB Ikeda series, on which it performs the best.

The degraded performance of the prediction-error cost relative to $J^{ml}(\mathbf{w})$ is somewhat expected because $J^{pe}(\mathbf{w})$ can be viewed as an approximation to $J^{ml}(\mathbf{w})$. Apparently, however, this approximation is less severe at lower SNRs, or when signal estimation is more difficult. Furthermore, the quadratic form of the prediction error cost conveys better stability properties.

Regarding the EM cost, although the use of purely static derivatives is justified from a theoretical perspective, this may be partially responsible for the cost's relatively unfavorable performance. In a sequential approach, such as the dual EKF, the information contained in the recurrent derivatives of the state estimate sequence with respect to the weights is fairly crucial. The importance of computing recurrent derivatives is investigated in Section 4.8 on page 180. Otherwise, the poor performance of $J^{em}(\mathbf{w})$ might be due to the approximations made in evaluating the EM cost sequentially.

On white noise and low SNRs, the potential benefits (lower variance) of $J^{j}(\mathbf{w})$ are outweighed by its short-comings (increased bias). At high SNR, the primary advantage of the joint cost over the maximum-likelihood approach is its superior stability and lower sensitivity to σ_v^2 and Q_0. At low SNRs, the signal estimates, $\hat{\mathbf{x}}_k$, may simply be too inaccurate, thereby increasing the bias of the joint cost. Hence, the marginal estimation costs $J^{pe}(\mathbf{w})$ and $J^{ml}(\mathbf{w})$ are favored because they are less sensitive to inaccurate signal estimates. Furthermore, this low SNR effect may also be responsible for the ranking on the Ikeda series, on which the signal NMSE values are especially high. In contrast, one reason for its relative success on colored noise may be that (with $\mathbf{w}_n$ known) for a given SNR, colored noise is effectively less random than white noise because it is partly deterministic. Thus, the bias problem may be less problematic, in general, for colored noise.

The error-coupled joint cost $J^{ec}(\mathbf{w})$ seems to suffer from conflicting requirements. On one hand, the algorithm requires reasonably good signal estimates, so that (as a joint method) its bias does not cause it to perform worse than the marginal approaches. On the other hand, however, the cost is designed to take estimation errors into account, so it will only outperform $J^{j}(\mathbf{w})$ when the estimation errors are fairly large. Furthermore, it relies somewhat on an assumption that the estimation errors are Gaussian: the approximation will be less severe for small errors than large ones. These factors may account for some of the unpredictability of the results.

4.7.2 Unknown Process Noise Variance

In the following set of experiments, the process noise variance σ_v^2 is unknown; it is estimated concurrently with the signal and weights using the modified variance filter shown in Formulae 3.12 and 3.13 on page 78. The measurement noise statistics (either σ_n^2, or $\mathbf{w}_n$ and $\sigma_{v_n}^2$) are again assumed known. The forgetting factor $\lambda_{\sigma_v^2} = .9993$ is used for variance estimation, and $\lambda_{\mathbf{w}} = .9999$, as before. Initial covariances of $\mathbf{P}_0 = \mathbf{I}$ and $\mathbf{Q}_0 = .1\mathbf{I}$ are used, with the exception of $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$, for which: $\mathbf{Q}_0 = .01\mathbf{I}$.

An example of the simultaneous estimation of the signal, weights, and process noise variance is shown in Figure 4.32, wherein dual EKF estimation of the chaotic neural network signal in 3dB

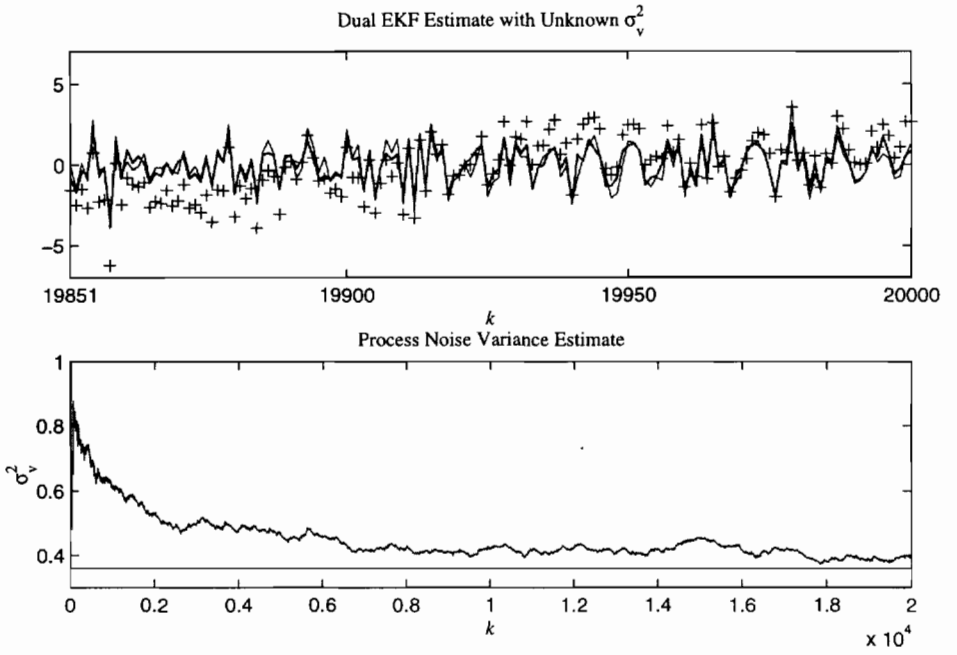


Figure 4.32: Example of dual EKF estimation of nonlinear time-series in 3dB colored noise, using the $J^j(\mathbf{w})$ and $J^{ml}(\sigma_v^2)$ cost functions. Only the last 150 points are shown. In the bottom plot, the $\hat{\sigma}_v^2$ trajectory is compared with σ_v^2 (horizontal line).

AS noise is performed with the $J^j(\mathbf{w})$ and $J^{ml}(\sigma_v^2)$ costs. The estimates are indicated by the heavy curve, the noisy data are shown by '+' signs, and the clean signal appears as a thin curve.

Each weight estimation cost is tested in conjunction with the three best costs for estimating σ_v^2, as determined previously in the known model case of Section 4.5. These are: $J^{pe}(\sigma_v^2)$ with $q_{v,0} = .1$, $J^{ml}(\sigma_v^2)$ with $q_{v,0} = .1$, and $J^j(\sigma_v^2)$ with $q_{v,0} = .01$. Because the limit cycle results in the previous section were shown to be consistent with other results using the same noise type, these data are omitted from the current set of experiments. Only the linear, chaotic neural network, and Ikeda series are considered here.

The boxplots in Figure 4.33 show the relative performance of the different cost function on the linear AR-10 data in stationary white measurement noise. For variance estimation, the success of the $J^{ml}(\sigma_v^2)$ cost is ubiquitous. For weight estimation, the performances of the $J^{ml}(\mathbf{w})$ and $J^{pe}(\mathbf{w})$ costs are very similar, although the prediction error cost has a significant advantage in all cases except the highest SNR. Because the model is linear, trajectories of the MSE in the weight estimates can be plotted along with the squared errors in $\hat{\sigma}_{v,k}^2$ in Figure 4.34. The advantage of the $J^{pe}(\mathbf{w})$ cost can be attributed to an interaction that produces faster convergence of both the weights and the variance estimate $\hat{\sigma}_{v,k}^2$.

Results for the chaotic neural network signal corrupted by stationary AR-5 noise are shown

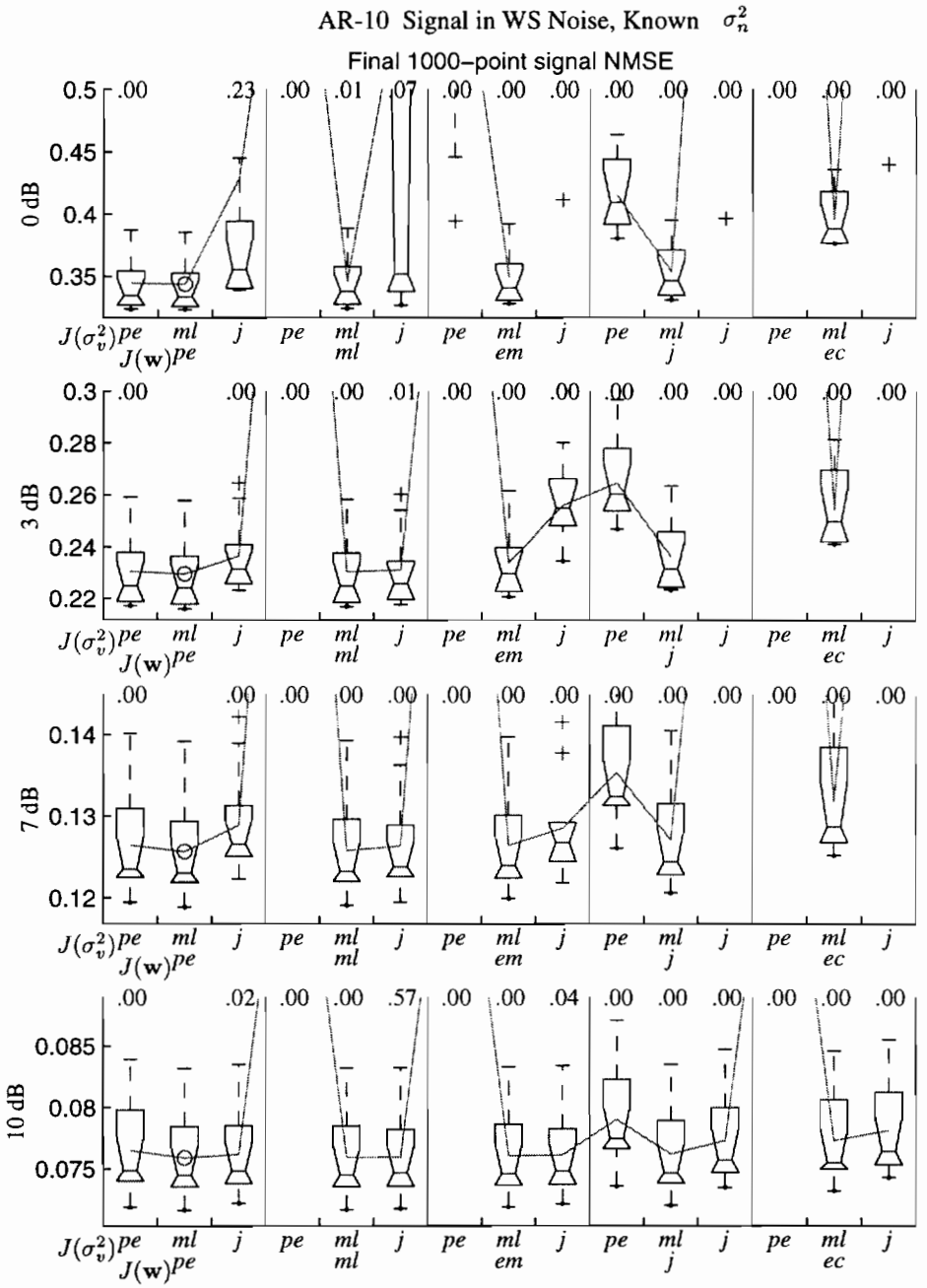


Figure 4.33: AR-10 data corrupted by white stationary noise at four different SNRs. Boxplots show the initial, overall, and final NMSEs for the signal estimates. As indicated, $J(\sigma_v^2)$ is varied within each panel, and $J(\mathbf{w})$ is varied across panels.

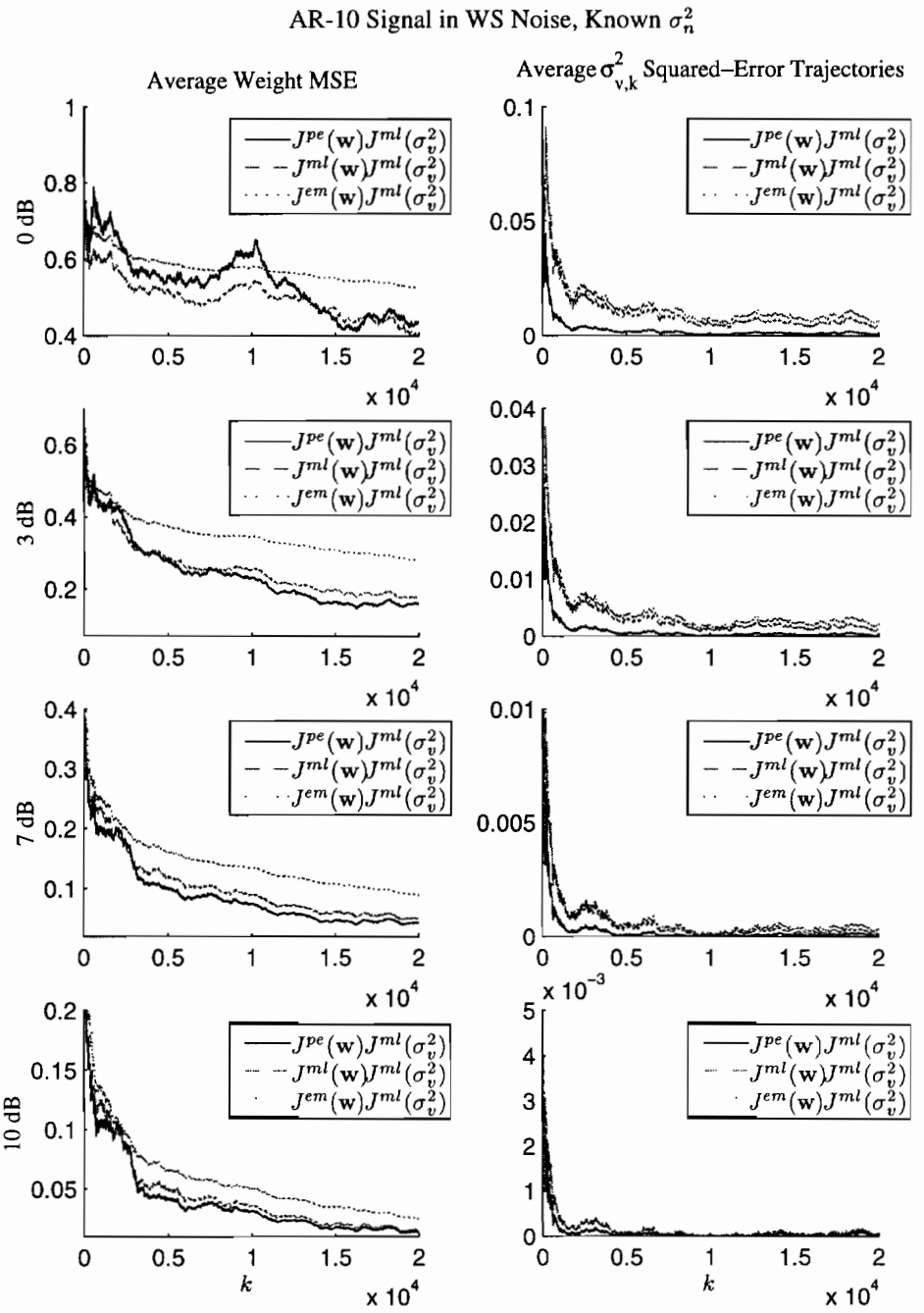


Figure 4.34: Weight estimate MSE trajectories, and σ_v^2 estimate squared error trajectories, for AR-10 data corrupted by white stationary noise at four different SNRs.

in the boxplots in Figure 4.35. Again, the variance estimation cost $J^{ml}(\sigma_v^2)$ is usually the best performer. For weight estimation, the maximum-likelihood cost $J^{ml}(\mathbf{w})$ gives the smallest average 1000-point NMSE at higher SNRs, while at 0dB $J^j(\mathbf{w})$ performs the best. Note that at most

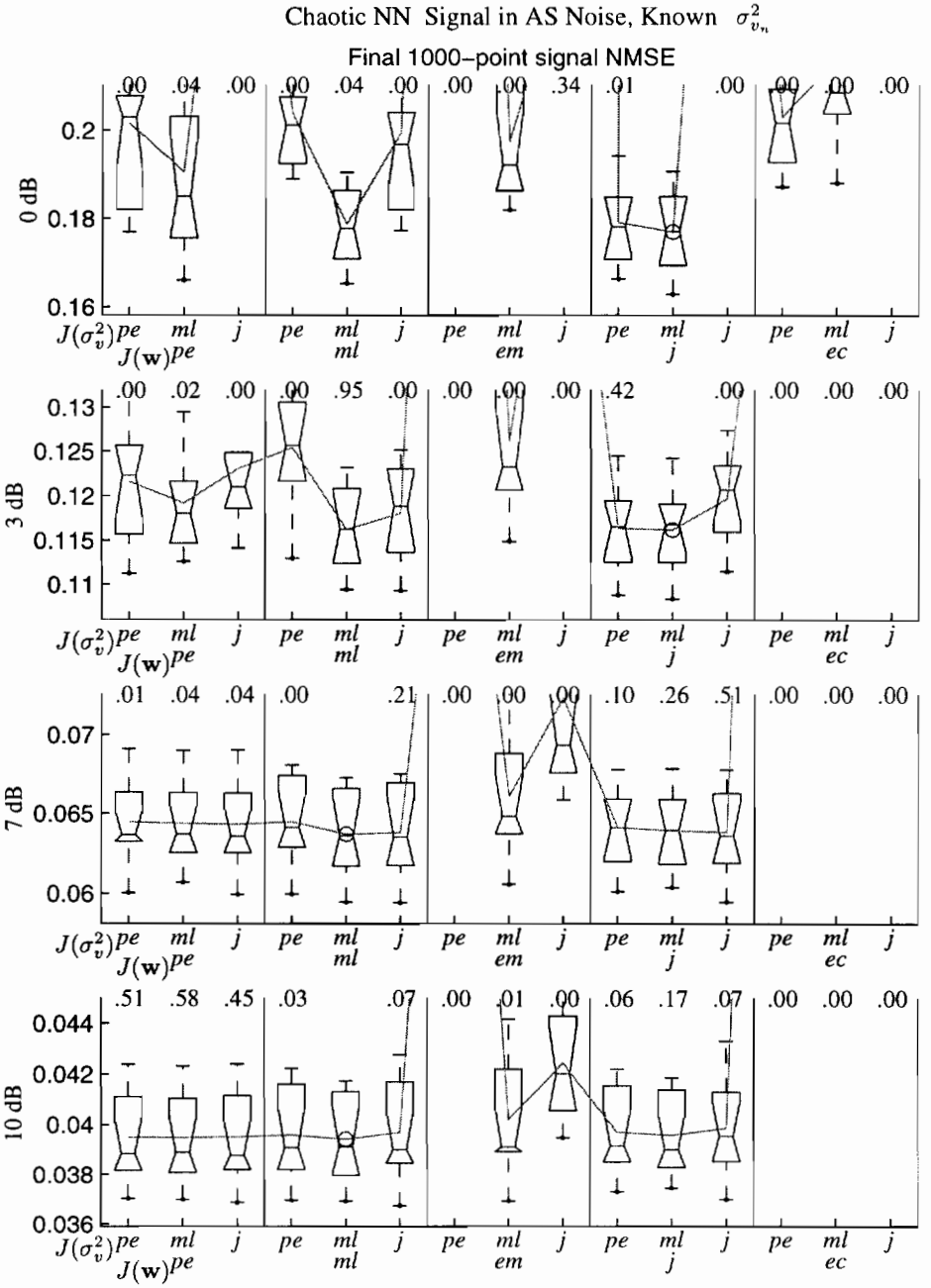


Figure 4.35: Chaotic neural network data corrupted by stationary AR-5 noise at four different SNRs. Boxplots show the initial, overall, and final NMSEs for the signal estimates. As indicated, $J(\sigma_v^2)$ is varied within each panel, and $J(\mathbf{w})$ is varied across panels.

SNRs, the joint cost $J^j(\mathbf{w})$ does not differ significantly from the best choice, making it a good general-purpose cost. The average differences between the dual estimation MSE profiles and the EKF profiles are shown in Figure 4.36, along with the average variance error trajectories.

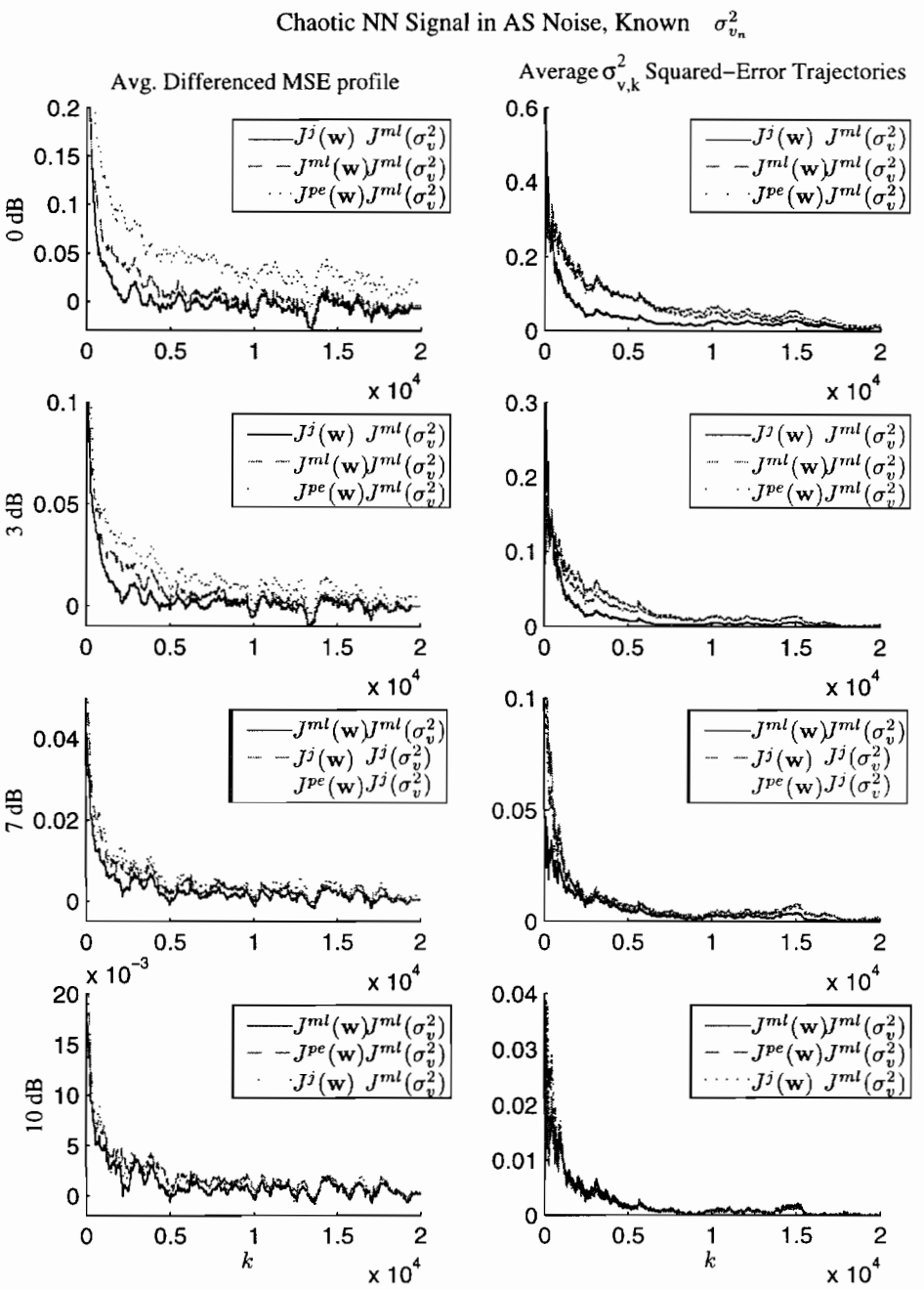


Figure 4.36: Ensemble averaged, differenced signal estimate MSE profiles, and variance estimate error trajectories, for chaotic neural network data corrupted by stationary AR-5 noise.

On *nonstationary* AR-5 noise added to neural network data, the results are very similar to those for stationary noise. As for the known variance case, the overall NMSE is used to rank the cost functions, rather than the final 1000-point NMSE. As indicated by the boxplots in Figure 4.37,

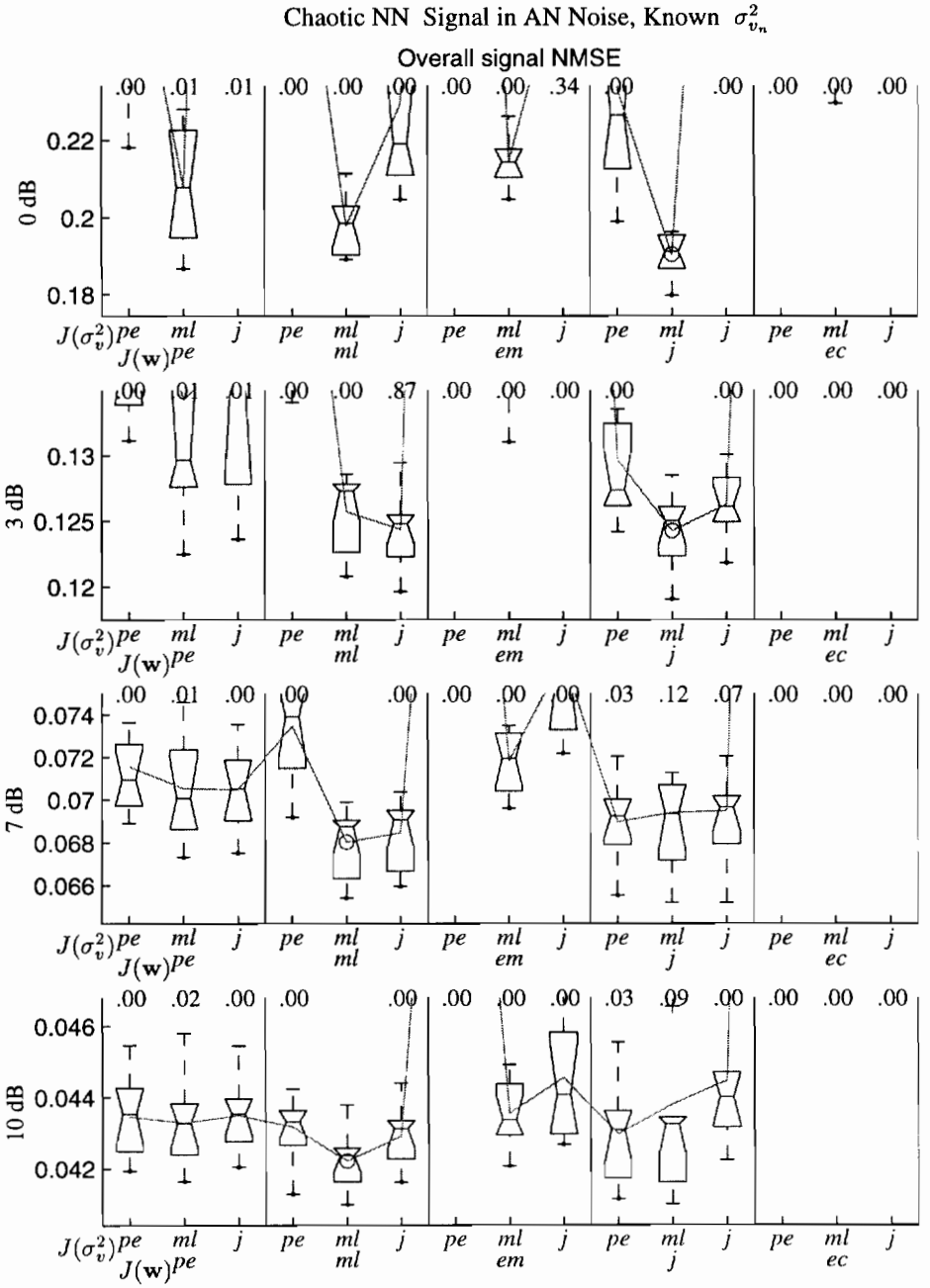


Figure 4.37: Chaotic neural network data corrupted by nonstationary AR-5 noise at four different SNRs. Boxplots show the initial, overall, and final NMSEs for the signal estimates. As indicated, $J(\sigma_v^2)$ is varied within each panel, and $J(\mathbf{w})$ is varied across panels.

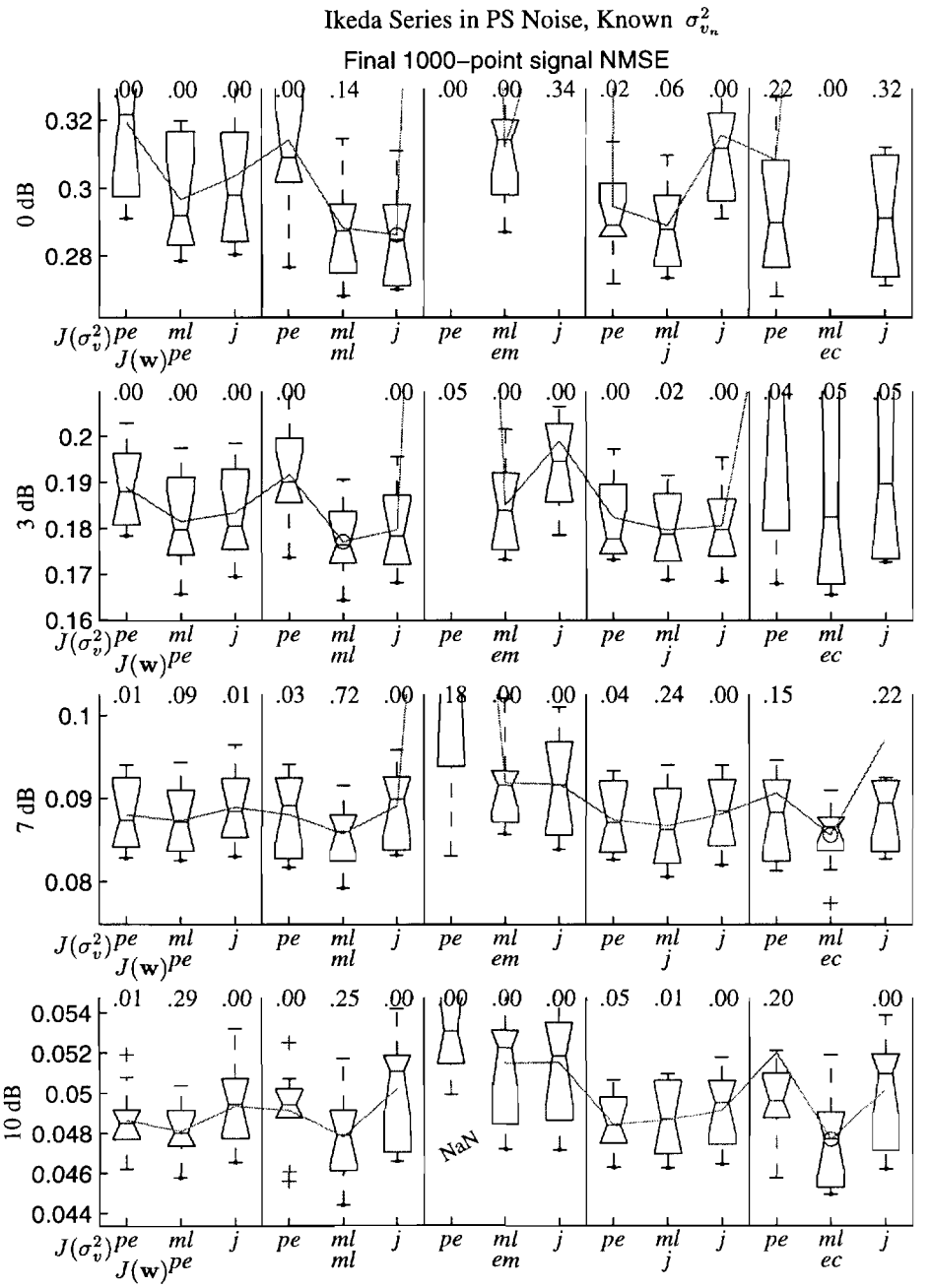


Figure 4.38: Normalized Ikeda data corrupted by stationary pink noise at four different SNRs. Boxplots show the overall and final 1000-point NMSEs for the signal estimates. As indicated, $J(\sigma_v^2)$ is varied within each panel, and $J(\mathbf{w})$ is varied across panels.

the joint cost $J^j(\mathbf{w})$ is the best choice at 0dB and 3dB SNR, and $J^{ml}(\mathbf{w})$ is better at the higher SNRs.

The NMSE values for the Ikeda data are shown in the boxplots of Figure 4.38. It is interesting to note that the performance here is actually better than in the "known σ_v^2" case, where a value obtained from training a predictor on clean data was used. Again, the maximum-likelihood weight cost $J^{ml}(\mathbf{w})$ is superior at 0dB SNR, whereas $J^{ec}(\mathbf{w})$ is better at higher SNRs. However, the advantage of the error-coupled weight cost over $J^{ml}(\mathbf{w})$ at the higher two SNRs is not very significant; although the $J^{ec}(\mathbf{w})$ is listed in Table 4.5, this choice is fairly arbitrary.

Table 4.5 summarizes the best cost functions when σ_v^2 is unknown. Because $J^{ml}(\sigma^2)$ is the best cost for estimating σ_v^2 in nearly all cases, it is not listed explicitly. Notice that the instability problem of the maximum-likelihood cost does not affect variance estimation, as the second derivative $\frac{\partial^2 J^{ml}}{(\partial \sigma^2)^2}$ is a scalar, and therefore cannot be "ill-conditioned." The weight cost choices largely mirror those given in Table 4.4, with only a few changes. Comparing the various boxplots with

Table 4.5: Best dual estimation cost functions for estimating $\mathbf{w}$ and σ_v^2 when the measurement noise statistics are known. Column headings are abbreviations for the data sets listed on page 162. In all cases, $J^{ml}(\sigma_v^2)$ is the best variance estimation cost.

SNR	AR-10(WS)	NNch(AS)	NNch(AN)	Ikeda(PS)
0 dB	$J^{pe}(\mathbf{w})$	$J^j(\mathbf{w})$	$J^j(\mathbf{w})$	$J^{ml}(\mathbf{w})$
3 dB	$J^{pe}(\mathbf{w})$	$J^j(\mathbf{w})$	$J^j(\mathbf{w})$	$J^{ml}(\mathbf{w})$
7 dB	$J^{pe}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ec}(\mathbf{w})$
10 dB	$J^{pe}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ml}(\mathbf{w})$	$J^{ec}(\mathbf{w})$

the known σ_v^2 case shows the weight costs are generally robust to initial inaccuracies in σ_v^2, and that the variance estimation filter is highly effective. In most cases, the final 1000-point NMSEs are not significantly different from when σ_v^2 is known. One exception is the improvement in the Ikeda results, discussed above. Another notable exception is that $J^j(\mathbf{w})$ actually *improves* its performance at lower SNRs when σ_v^2 is being estimated; this may be a result of a larger value of $\hat{\sigma}_{v,k}^2$ accounting for errors in the model, much in the way the $J^{ec}(\mathbf{w})$ cost was designed (but generally fails) to work. This effect seems to be responsible for the top ranking of $J^j(\mathbf{w})$ on the chaotic neural network in 0dB AN noise.

4.7.3 Both Variances Unknown

Recall that the signal is characterized by both the model weights, $\mathbf{w}$, and the variance of the process noise, σ_v^2. The process noise often represents the stochastic component of the dynamics not represented by the model; hence, the process noise variance depends on specification of the

model. Therefore, a scenario in which the process noise variance σ_v^2 is known, but the weights and measurement noise variance are not, is relatively unlikely. This case is therefore not investigated experimentally.

Rather, in this set of experiments both the process noise variance, σ_v^2, and measurement noise variance, σ_n^2 ($\sigma_{v_n}^2$ for colored noise), are assumed unknown. Each is estimated along with the signal and weights using a modified variance filter shown in Formulae 3.12 and 3.13 on page 78. For colored measurement noise, the model $\mathbf{w}_n$ is assumed known. The forgetting factors used for variance estimation are: $\lambda_{\sigma_v^2} = .9993$ and $\lambda_{\sigma_n^2} = .9993$.

An example of dual EKF estimation with unknown variance is shown in Figure 4.39. In the top plot, the estimates are indicated by the heavy curve, the noisy data are shown by '+' signs, and the clean signal appears as a thin curve.

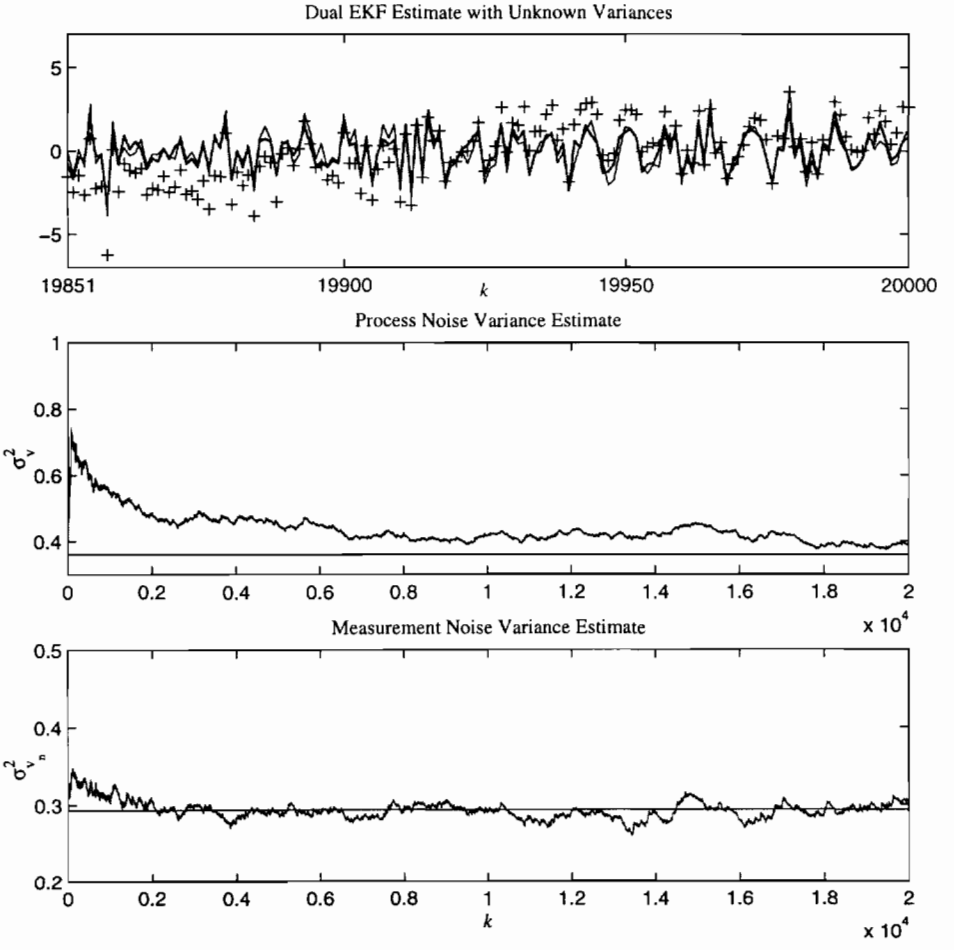


Figure 4.39: Example of dual EKF estimation of nonlinear time-series in 3dB colored noise, using the $J^j(\mathbf{w})$ $J^{ml}(\sigma_v^2)$ and $J^{ml}(\sigma_{v_n}^2)$ cost functions. Only the last 150 points are shown. In the bottom two plots, the $\hat{\sigma}_v^2$ and $\hat{\sigma}_{v_n}^2$ trajectories are compared against their true values (horizontal lines)

The following results provide a comparison the various dual EKF cost functions. As in the previous group of experiments, four data sets are used: AR-10 in WS noise, chaotic neural network in AS and in AN noise, and the Ikeda series in pink noise. Because the stability problems of $J^{ml}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ do not arise on the AR-10 and Ikeda signals, the smaller initial weight covariance $\mathbf{Q}_0 = .01\mathbf{I}$ is used for these costs only on the neural network data; $\mathbf{Q}_0 = .1\mathbf{I}$ is used everywhere else. The signal covariance is initialized by: $\mathbf{P}_0 = \mathbf{I}$.

On each of the four data sets, the three or four most promising weight estimation costs are tested in conjunction with the three best costs for estimating the measurement noise variance, as determined in Section 4.5. These are: $J^{pe}(\sigma_n^2)$, $J^{ml}(\sigma_n^2)$, and $J^{em}(\sigma_n^2)$, with initial error variance $q_{n,0} = .1$ used for white noise, and $q_{n,0} = .01$ used when the noise is colored. In light of the previous group of experimental results, the cost for estimating σ_v^2 is fixed at $J^{ml}(\sigma_v^2)$ with $q_{v,0} = .1$.

On the linear AR-10 signal corrupted by stationary white measurement noise, the weight costs $J^{pe}(\mathbf{w})$, $J^{ml}(\mathbf{w})$, $J^{em}(\mathbf{w})$, and $J^{j}(\mathbf{w})$ are tested. The boxplots in Figure 4.40 show the final 1000-point signal estimation NMSEs of the different combinations of cost functions. For weight estimation, the $J^{ml}(\mathbf{w})$ and $J^{pe}(\mathbf{w})$ costs perform similarly, although the maximum-likelihood cost shows a slight advantage at higher SNRs. Unfortunately, while the variance estimation costs show more distinct differences in performance, there is little consistency between noise levels: at 0dB, $J^{pe}(\sigma_n^2)$ has the best performance, although it shows the widest range in its weight MSEs; at 3dB SNR, $J^{pe}(\sigma_n^2)$ and $J^{em}(\sigma_n^2)$ are not significantly different, but $J^{pe}(\sigma_n^2)$ shows an advantage in terms of weight error; at 7dB, $J^{ml}(\sigma_n^2)$ and $J^{em}(\sigma_n^2)$ appear equivalent, but $J^{em}(\sigma_n^2)$ shows convergence to significantly lower weight MSE in Figure 4.41 on page 173; finally, at 10dB, $J^{ml}(\sigma_n^2)$ is significantly better than the other costs. Ideally, then, $J^{pe}(\sigma_n^2)$ should be used at lower SNRs, $J^{em}(\sigma_n^2)$ at medium SNRs, and $J^{ml}(\sigma_n^2)$ should be used at higher SNRs. This is clearly an undesirable situation because the SNR will generally not be known in advance (we are, after all, estimating σ_n^2). However, the penalty for selecting a suboptimal cost is not terribly high in this case; any of the top three costs will yield good performance.

The results for the neural network data in stationary AR-5 noise are considerably more consistent. As shown in Figure 4.42 on page 174, the final 1000-point NMSE is almost always lowest for the $J^{ml}(\sigma_{v_n}^2)$ variance cost. The $J^{j}(\mathbf{w})$ weight cost does the best at low SNRs, while $J^{ml}(\mathbf{w})$ is better at high SNRs.

In nonstationary colored noise, the results are ranked according to the overall signal NMSE. Figure 4.43 is largely consistent with the stationary noise results, with $J^{j}(\mathbf{w})J^{ml}(\sigma_{v_n}^2)$ the best choice at low SNRs, and $J^{ml}(\mathbf{w})J^{ml}(\sigma_{v_n}^2)$ better at 10dB. At 10dB, the joint cost actually appears to suffer from some stability problems, and gives its best performance in combination with the

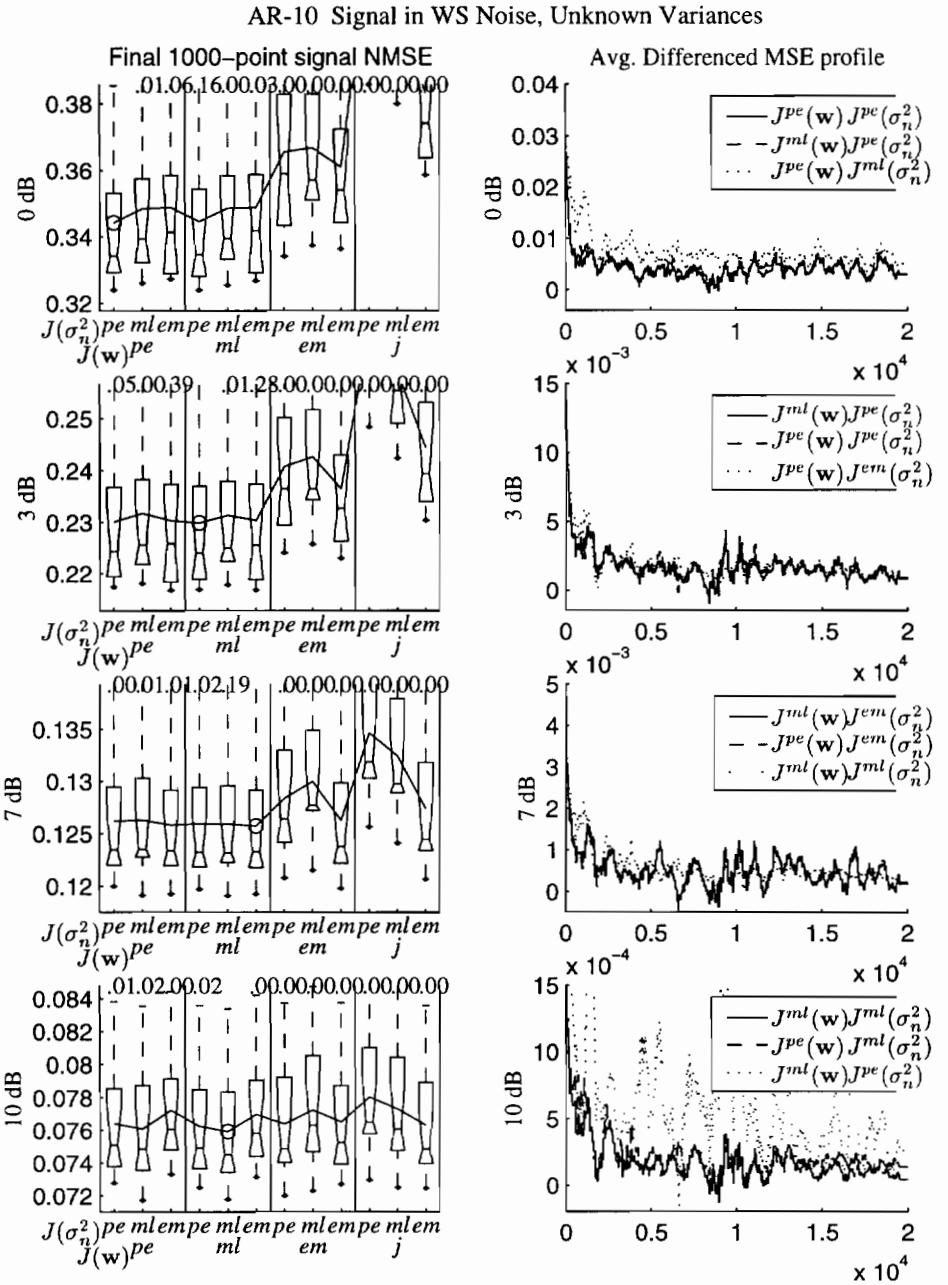


Figure 4.40: AR-10 data corrupted by white stationary noise at four different SNRs. Boxplots show the final 1000-point NMSEs for the signal estimates.

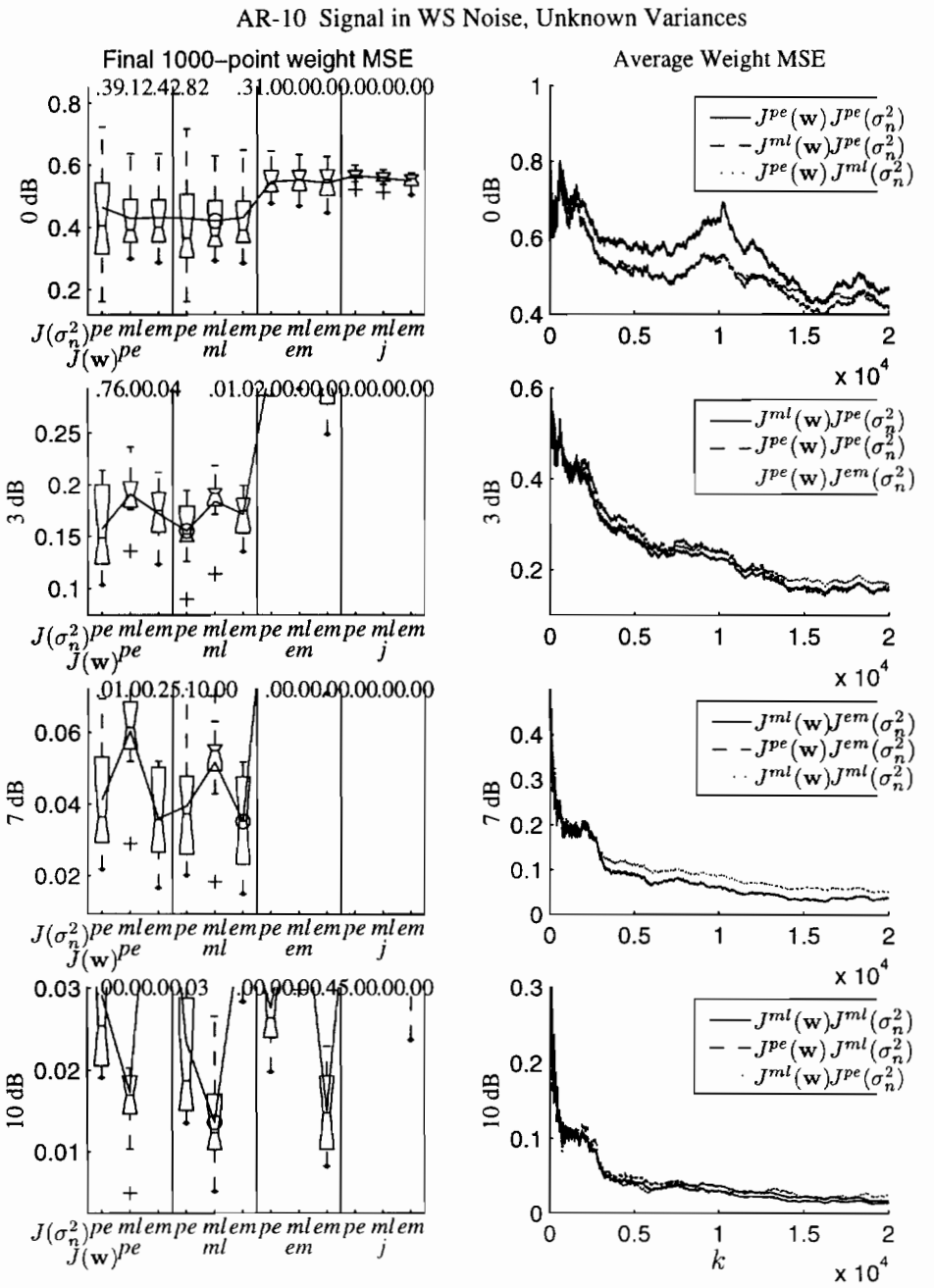


Figure 4.41: AR-10 data corrupted by white stationary noise at four different SNRs. Boxplots show the final 1000-point weight MSEs.

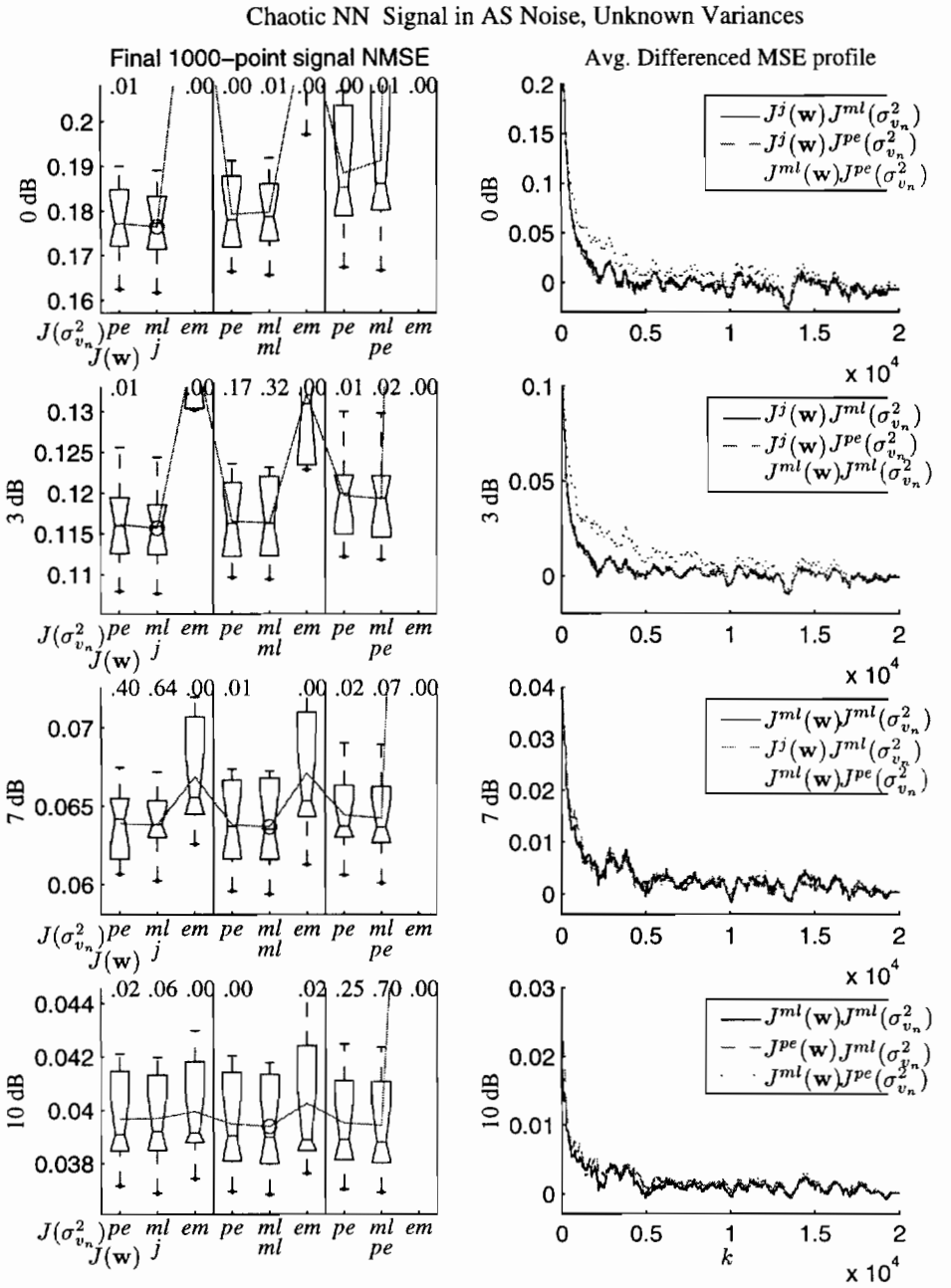


Figure 4.42: Chaotic neural network data corrupted by stationary colored (AR-5) noise at four different SNRs. Boxplots show the final 1000-point NMSEs for the signal estimates.

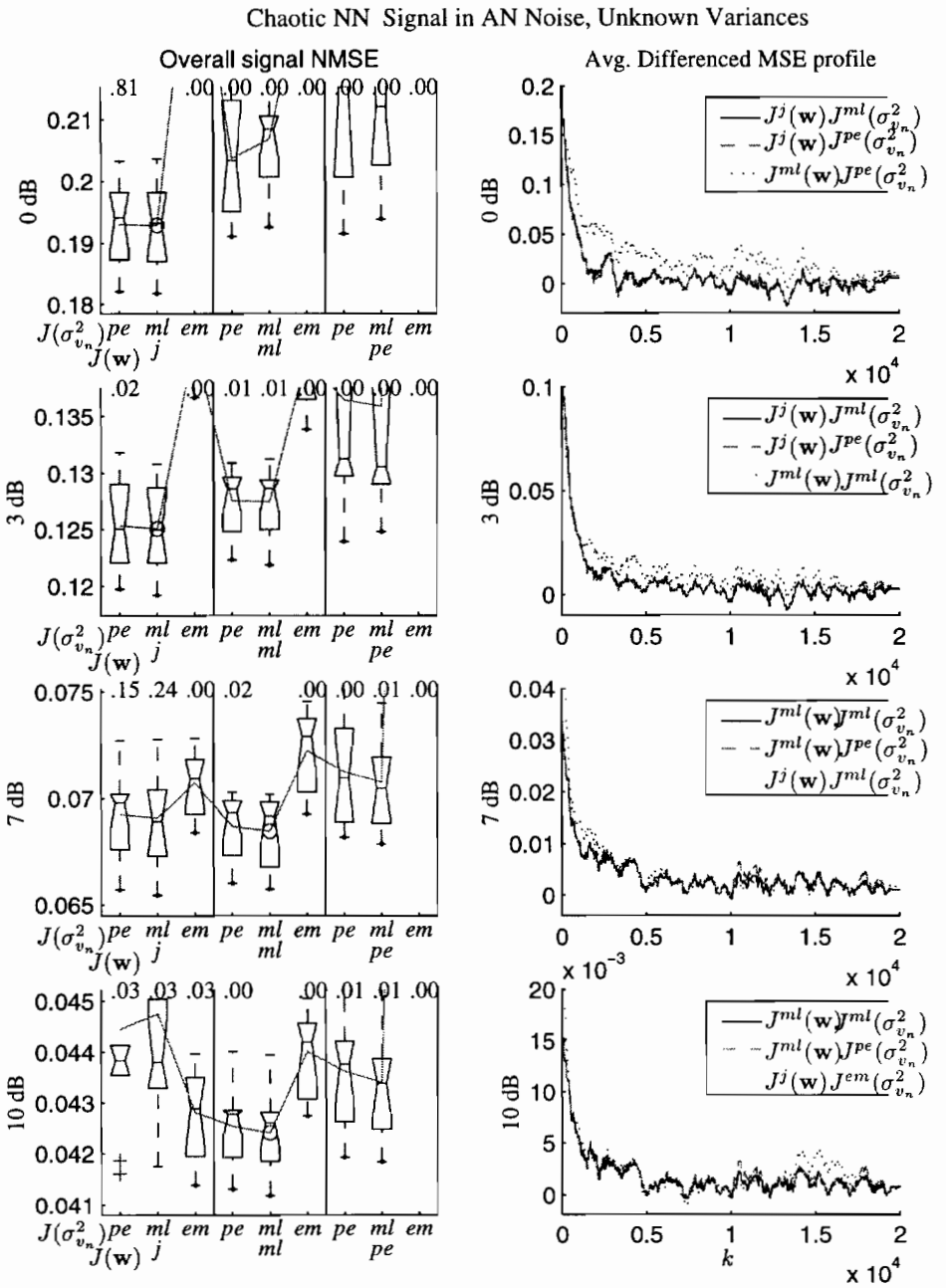


Figure 4.43: Chaotic neural network data corrupted by nonstationary colored (AR-5) noise at four different SNRs. Boxplots show the final 1000-point NMSEs for the signal estimates.

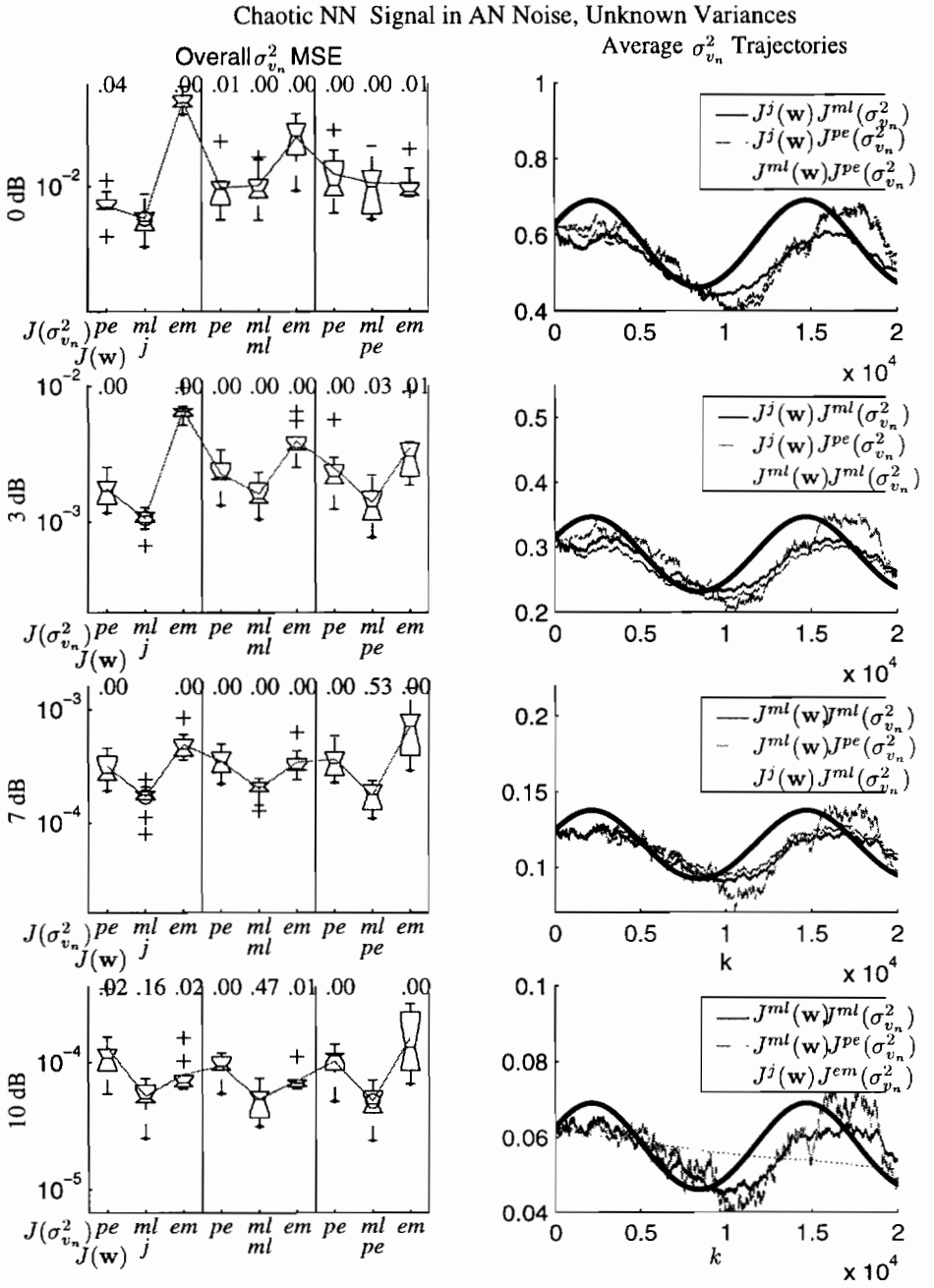


Figure 4.44: Chaotic neural network data corrupted by nonstationary colored noise at four different SNRs. Boxplots show the final 1000-point NMSEs for the variance estimates, $\hat{\sigma}^2_{v_n}$.

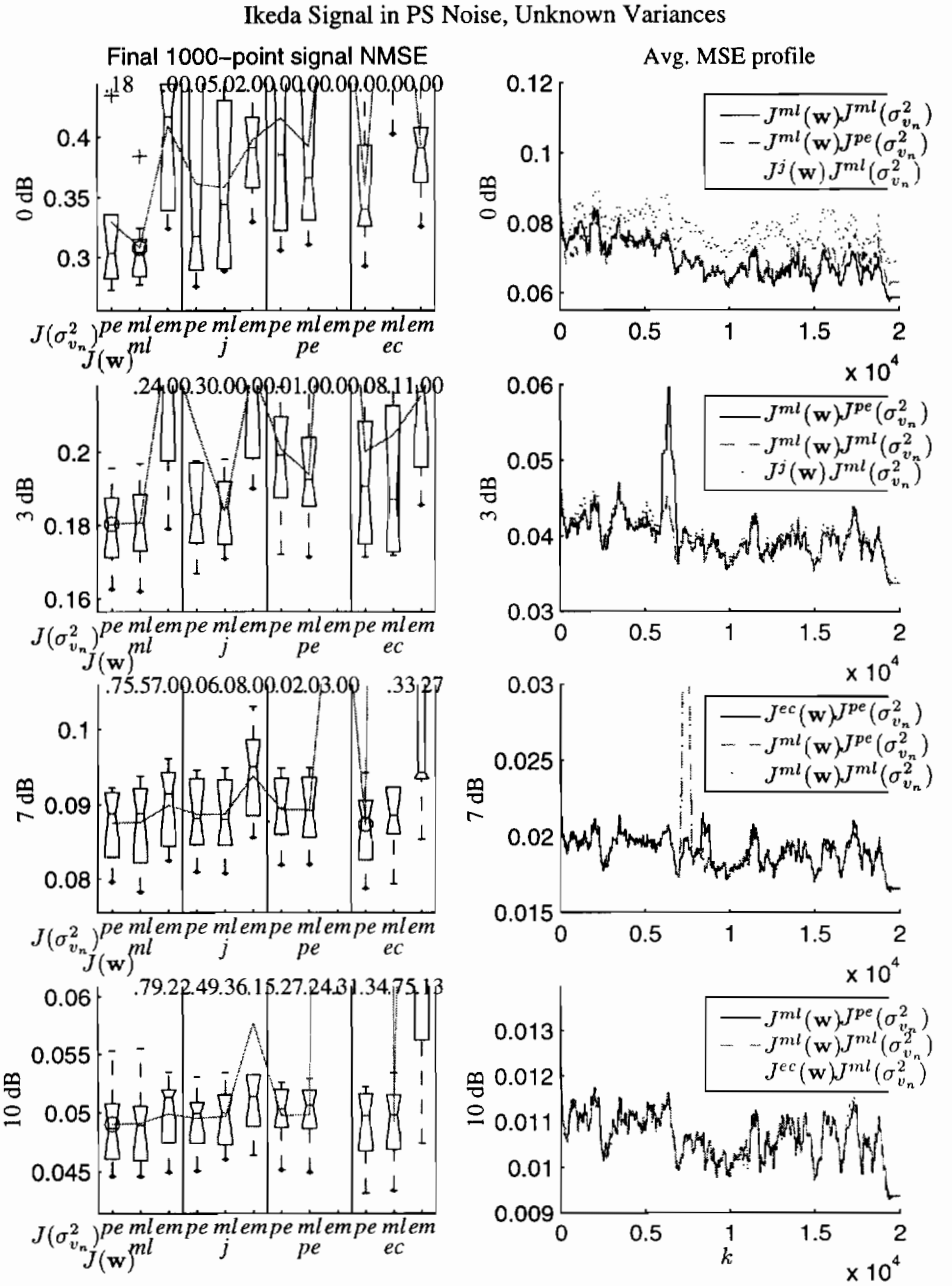


Figure 4.45: Normalized Ikeda map data corrupted by stationary pink noise at four different SNRs. Boxplots show the final 1000-point NMSEs for the signal estimates.

$J^{em}(\sigma^2_{v_n})$ variance cost, which is otherwise a suboptimal choice. As with $J^{ml}(\mathbf{w})$, better results might be obtainable by using $Q_0 = .01$ for the joint weight cost at 10dB.

Figure 4.44 on page 176 shows the overall $\sigma^2_{v_n}$ NMSEs, along with the average trajectories of $\hat{\sigma}^2_{v_n,k}$. The best performance is generally provided by $J^{ml}(\sigma^2_{v_n})$. Although the $J^{pe}(\sigma^2_{v_n})$ cost does a better job of tracking the nonstationarity, its higher volatility hurts its overall performance. Note that at 10dB the $J^{em}(\sigma^2_{v_n})$ cost displays a very slow time constant, resulting in significant lag misadjustment.

Figure 4.45 shows the results on the Ikeda series with additive pink noise. At 0dB and 3dB, $J^{ml}(\mathbf{w})$ is the best weight cost in terms of final 1000-point signal NMSE; at 7dB and 10dB SNR, it is not distinguishable from $J^{ec}(\mathbf{w})$. However, both the maximum-likelihood and error-couple costs shows signs of instability at the 3dB and 7dB noise levels, making comparison difficult. Although better performance can be achieved by using $Q_0 = .01$ for the initial weight covariance of these costs, this figure helps underscore their susceptibility to unstable behavior. For variance estimation, $J^{ml}(\sigma^2_{v_n})$ and $J^{pe}(\sigma^2_{v_n})$ show statistically equivalent performance at all SNRs (when paired with $J^{ml}(\mathbf{w})$). The EM variance cost is significantly worse at all but the highest SNR.

An examination of the boxplots shows again that the dual EKF costs are highly robust to inaccuracies in the noise variances, and that the variance estimation filters generally provide good performance. The best cost functions when both variances are unknown are summarized in Table 4.6.

Table 4.6: Best dual estimation cost functions when estimating $\mathbf{w}$, σ^2_v, and σ^2_n (or $\sigma^2_{v_n}$). The process noise is estimated using the cost $J^{ml}(\sigma^2_v)$, as determined previously. Each row indicates the SNR of the noisy data. Column headings are abbreviations for the data sets listed on page 171.

SNR	AR-10(WS)	NNch(AS)	NNch(AN)	Ikeda(PS)
0 dB	$J^{ml}(\mathbf{w})$ $J^{pe}(\sigma^2_n)$	$J^{j}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{j}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$
3 dB	$J^{ml}(\mathbf{w})$ $J^{pe}(\sigma^2_n)$	$J^{j}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{j}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$
7 dB	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_n)$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$
10 dB	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_n)$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$	$J^{ml}(\mathbf{w})$ $J^{ml}(\sigma^2_{v_n})$

4.7.4 Effect of Prior Knowledge

It is interesting at this point to stop and compare the best signal estimation results when estimating the signal, weights, and both variances using a dual Kalman filter or dual EKF, with results that can be obtained when the variances are known, or when applying a Kalman filter or EKF using the known model and noise variances. Figure 4.46 provides such a comparison in the form of boxplots of the final 1000-point signal NMSE, when estimating the AR-10 signal in white noise and the

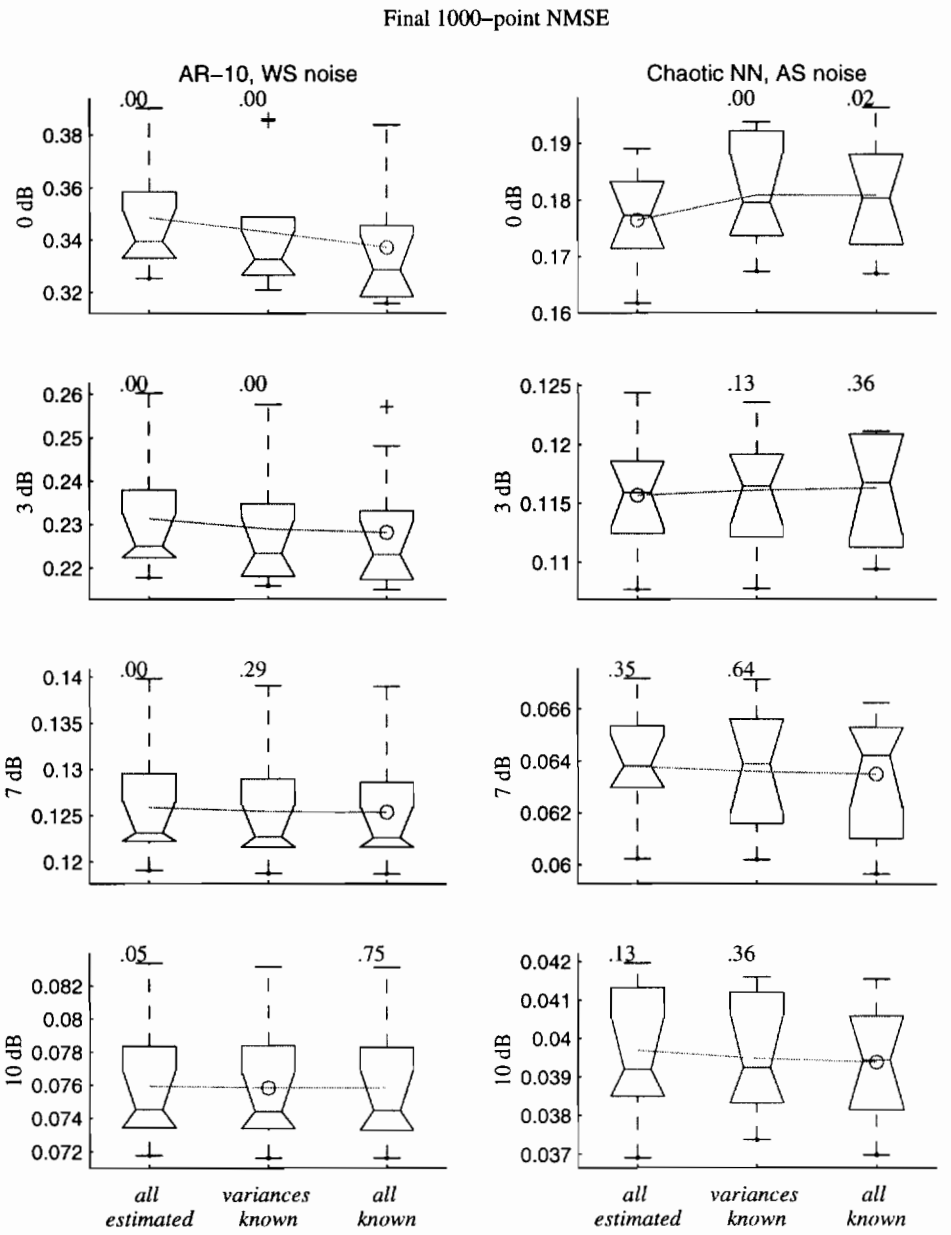


Figure 4.46: Final 1000-point signal NMSEs when the signal, model, and noise variances are all estimated (by a dual Kalman filter), when only the variances are known (signal and model estimated by dual Kalman filter), and when everything is exactly known (signal estimated by Kalman filter or EKF).

chaotic network signal in stationary colored noise, with $N = 20,000$ points of data at 4 different SNRs each. On the linear data, $J^{ml}(\mathbf{w})$ is used for weight estimation, and $J^{j}(\mathbf{w})$ is used on the neural network data; $Q_0 = .1$ is used in both cases.

The results on the linear data are as expected: prior knowledge of the true model and noise statistics gives a significant performance advantage (although the difference is too small to see from a plot of the signal estimates). This advantage diminishes slowly as the SNR increases and the signal estimation problem becomes less difficult. Interestingly, knowledge of the noise variances has at least as big an impact on performance as does knowledge of the weights; at 10 dB SNR, the known variance results and Kalman filter results are indistinguishable.

However, on the nonlinear data, something interesting occurs: at low SNRs, the dual EKF with unknown variances actually performs better than the EKF. Why is this so? The most plausible explanation is that the Taylor series approximations inherent in the EKF algorithm make it a suboptimal estimator. However, the inaccuracies in the filter are partially compensated for by adjusting the values of the noise variances, and to a lesser extent, adjusting the model itself. Furthermore, the approximations made by the EKF are more severe when the signal error covariances $\mathbf{P}_k$ are on the same scale as the curvature of the nonlinearities. This is more likely to happen with noisier data, or with strongly nonlinear signals. Note that the effect is absent at higher SNRs, and on the linear data.

Hence, it is evident from Figure 4.46 that the dual Kalman filter converges to solutions which are both reasonably close to the KF results on linear data, and which are potentially better than estimates produced by the EKF on noisy nonlinear data.

4.8 Experiment 5: Static Derivatives in the Dual EKF

Section 3.6.1 showed recursive equations for the derivatives of the prediction $\hat{\mathbf{x}}_{k+1}^{-}$ and estimate $\hat{\mathbf{x}}_k$ with respect to the weights $\hat{\mathbf{w}}$ and variances $\hat{\sigma}^2$. In some situations, the expense associated with these computations is too high, and a cheaper alternative must be considered. One such alternative simply ignores the dependence of the state estimate $\hat{\mathbf{x}}_k$ on the weights. This allows the derivative of the prediction to be computed as the partial derivative of the model with respect to the weights alone, and greatly reduces the computational cost. However, ignoring the derivative of the estimate, $\hat{\mathbf{x}}_k$, also has a rather significant effect on the form of the joint cost functions $J^{j}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$, both of which include the signal estimate $\hat{x}_k$ in one of their terms.

Figure 4.47 shows the effect of using static derivatives when estimating the chaotic neural network time-series in 3dB colored noise. The difference between the static and full derivative

estimates is slight enough that it is difficult to see on the scale of the estimates; the bottom plot shows the difference at 10 times the scale of the top plot.

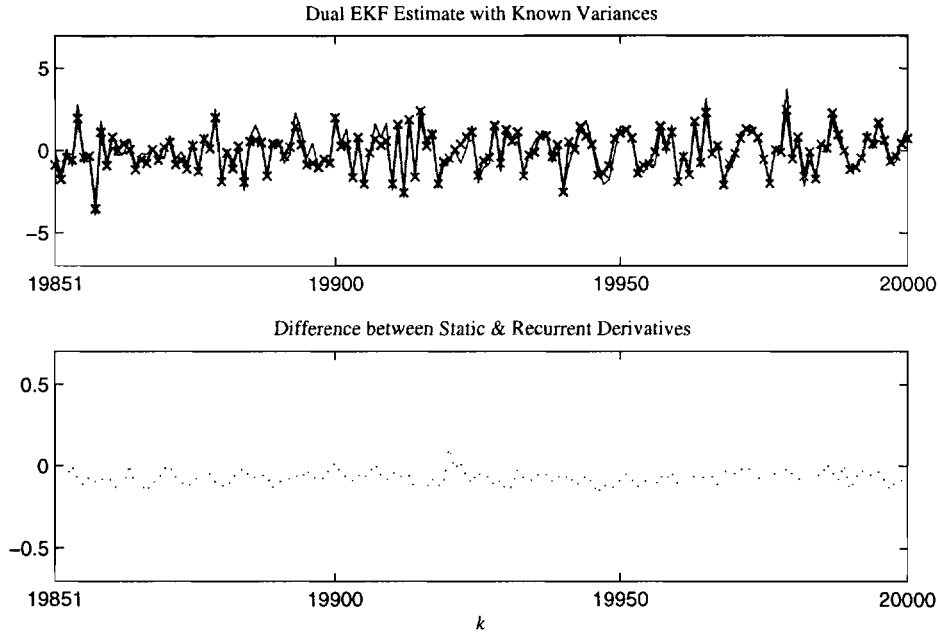


Figure 4.47: The effect of static derivatives on dual EKF estimation of the chaotic neural network time-series in 3dB colored noise, using the $J^j(\mathbf{w})$ cost and known noise variances. The bottom plot shows the difference between estimates using the static derivatives ('×' in top plot), and the full derivatives (heavy curve in top plot).

The general effect of the simplification on algorithm performance is shown in Figure 4.48, using the AR-10 and limit cycle data sets in white noise, the chaotic neural network data in both stationary and nonstationary AR-5 noise, and the Ikeda series in pink noise. In all cases, both variances are assumed known.

On each data set, the signal NMSE for one choice of cost function is shown, followed by the performance when that cost is used with static derivatives. The initial covariance is $\mathbf{Q}_0 = .1\mathbf{I}$ in all cases. On the two data sets with white noise, the effect is negligible, although the average NMSE is slightly higher with static derivatives in seven of the eight examples. However, the performance difference is much more noticeable on the three colored noise data sets, and is significant in nine of the twelve cases.

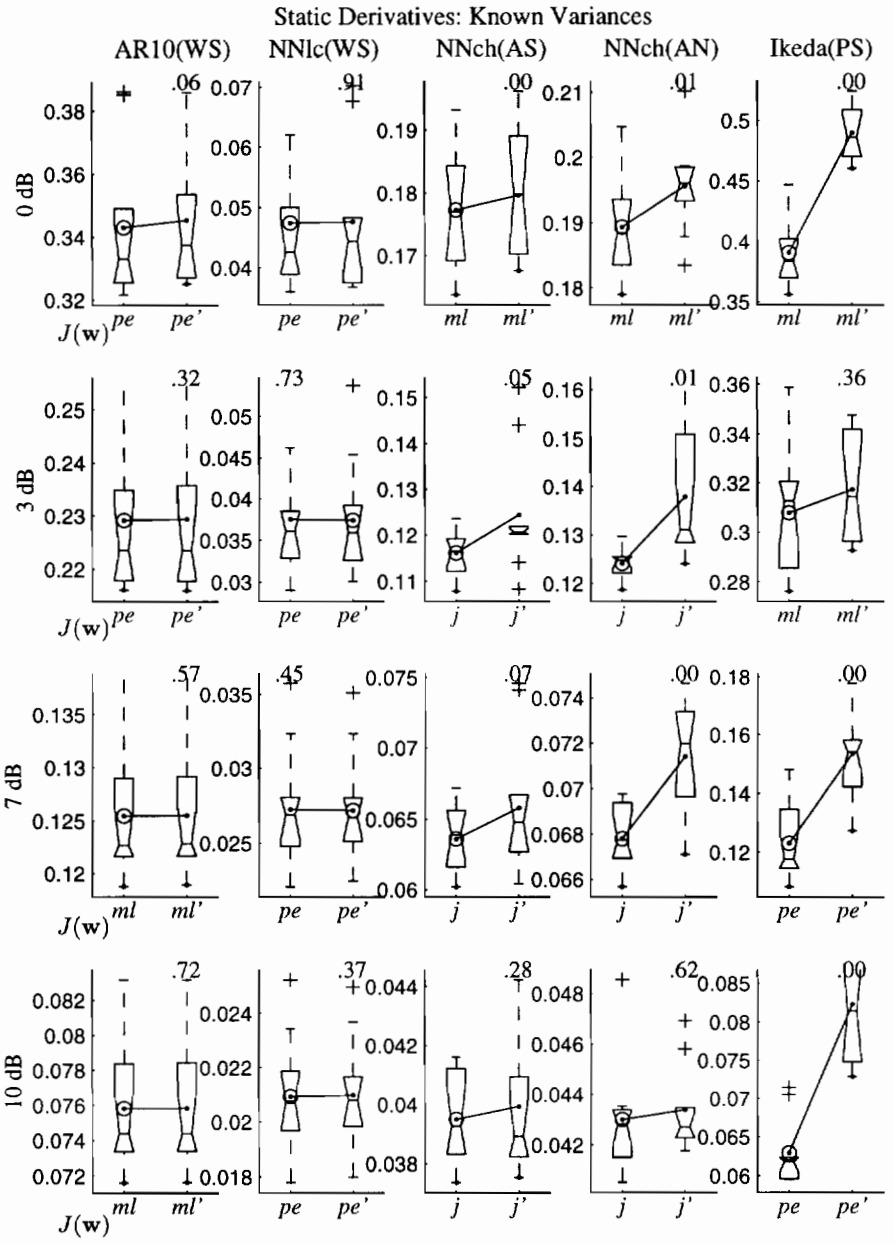


Figure 4.48: The effect of using static derivatives of $f(\cdot)$ in the dual EKF, rather than full recurrent derivatives. The static derivative plots are labeled with a prime (') in the superscript. Boxplots show the final 1000-point signal NMSEs on all data sets, except on the chaotic NN (AN) data (fourth column), in which the overall signal NMSE is used.

4.9 Experiment 6: Joint EKF Performance

In this section, the performance of the joint EKF (JEKF) algorithm is compared with that of the dual EKF (DEKF). The three cases: noise variances known; σ_v^2 unknown; and both noise variances unknown, are all considered using the same data and settings of parameters λ, $\mathbf{P}_0$, etc., as in Section 4.7. In each case, the joint EKF performance is compared against the performance of the best dual EKF cost function for that particular data set and SNR.

Following comments made in [45, 47, 61] regarding convergence problems of the joint EKF, and our own analysis (see Section 3.4.1) of the difficulties of this approach, the joint EKF might be expected to generally perform somewhat worse than the dual EKF. However, these experiments show that the joint EKF can, in fact, give the same or better performance as the dual EKF in many cases.

4.9.1 JEKF: Known Variances

Figure 4.49 shows the results when both noise variances are known. In the white noise experiments, shown in the left two columns, there is generally no significant difference in performance between the dual EKF and joint EKF. Nonetheless, the joint EKF is favored on the linear data by its slightly lower average NMSE in all cases (the advantage is significant at 3dB). The results are considerably more mixed on the limit cycle data. A similar story is told by the stationary and nonstationary AR-5 noise results, shown in the third and fourth columns. Here, the joint EKF shows a slight advantage at 0dB, and the dual EKF does somewhat better at 10dB. Note that the performance on the AN noise is measured in terms of overall NMSE, and so contains information about the tracking behavior of the algorithms.

The results are much more striking on the Ikeda series; here, the dual EKF shows a distinct advantage at every SNR. There are a couple of possible causes for this outcome. First, the nonlinearity of the Ikeda map may actually be severe enough so that the additional nonlinearity of the joint state space hurts the JEKF's performance. Second, recall that the process noise variance σ_v^2 is not accurately known for the Ikeda series; it is possible that the JEKF is more sensitive to this inaccuracy than the DEKF, and its performance suffers as a result. This hypothesis is in agreement with an observation in [45] about the JEKF's sensitivity to the noise variances.

4.9.2 JEKF: Unknown Process Noise Variance

The case of σ_v^2 unknown is considered in Figure 4.50. The results are very similar to the known variance case, except that the significance of the JEKF's advantage is generally increased in the

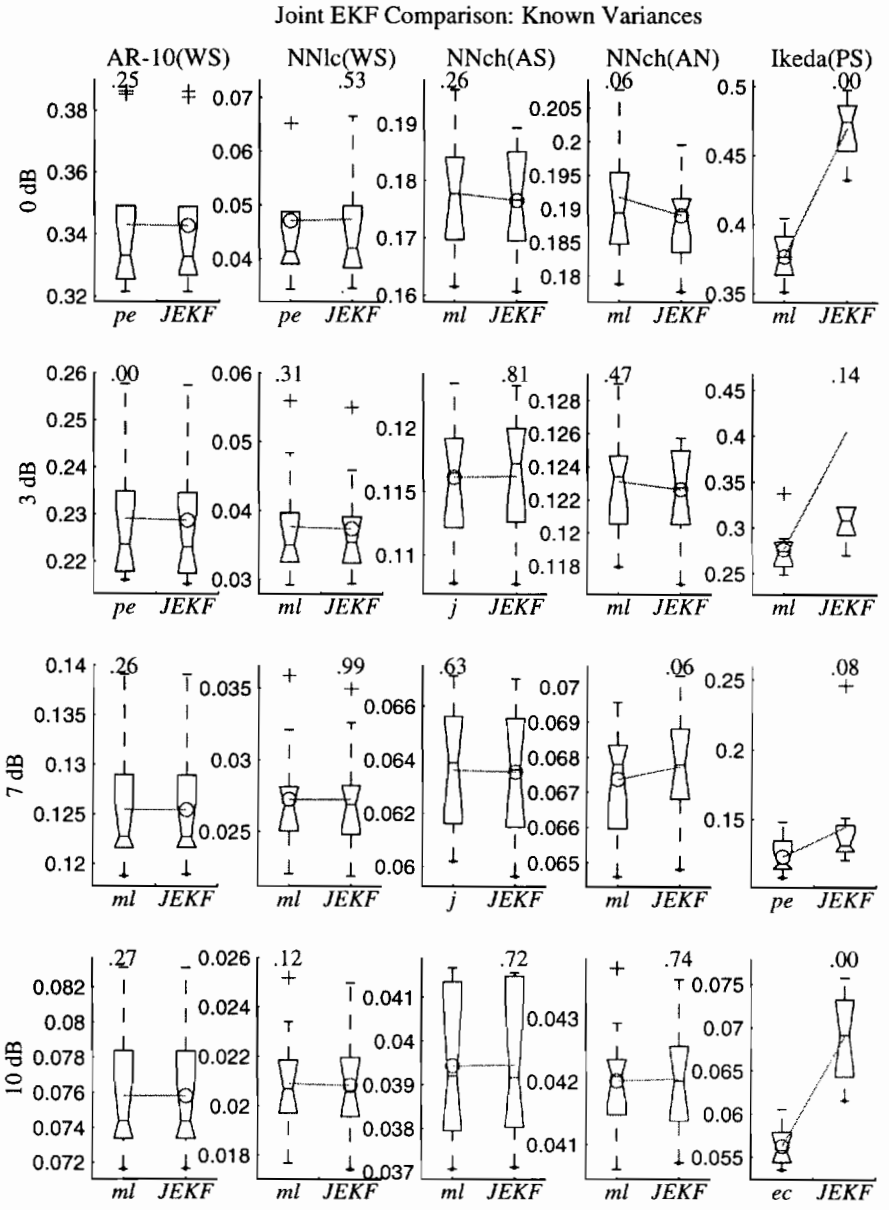


Figure 4.49: The performance of the joint EKF compared with the best dual EKF cost functions, when both noise variances are known. Boxplots show the final 1000-point signal NMSEs on all data sets, except on the chaotic NN (AN) data (fourth column), in which the overall signal NMSE is used.

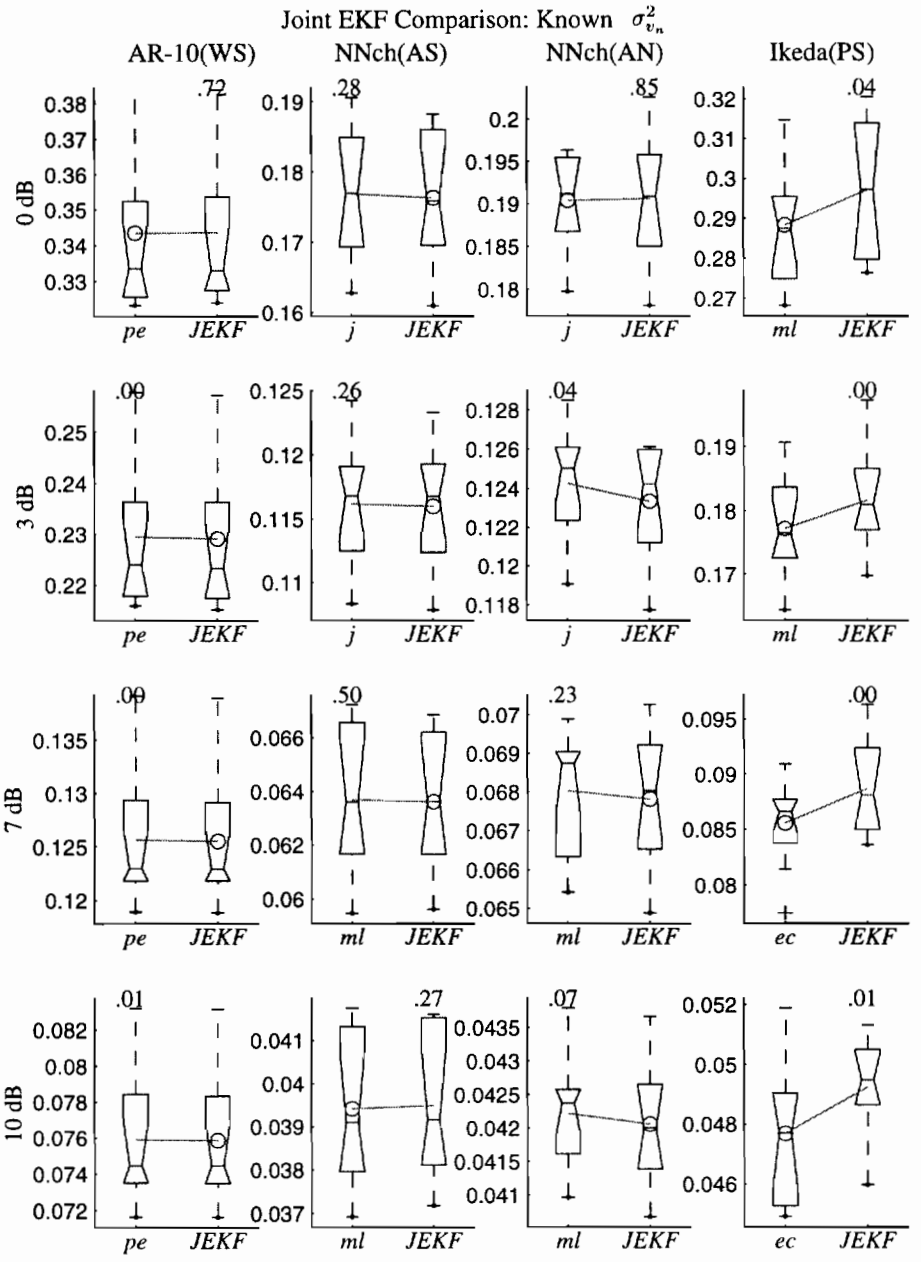


Figure 4.50: The performance of the joint EKF compared with the best dual EKF cost functions, when only the measurement noise statistics are known. Boxplots show the final 1000-point signal NMSEs on all data sets, except on the chaotic NN (AN) data (fourth column), in which the overall signal NMSE is used.

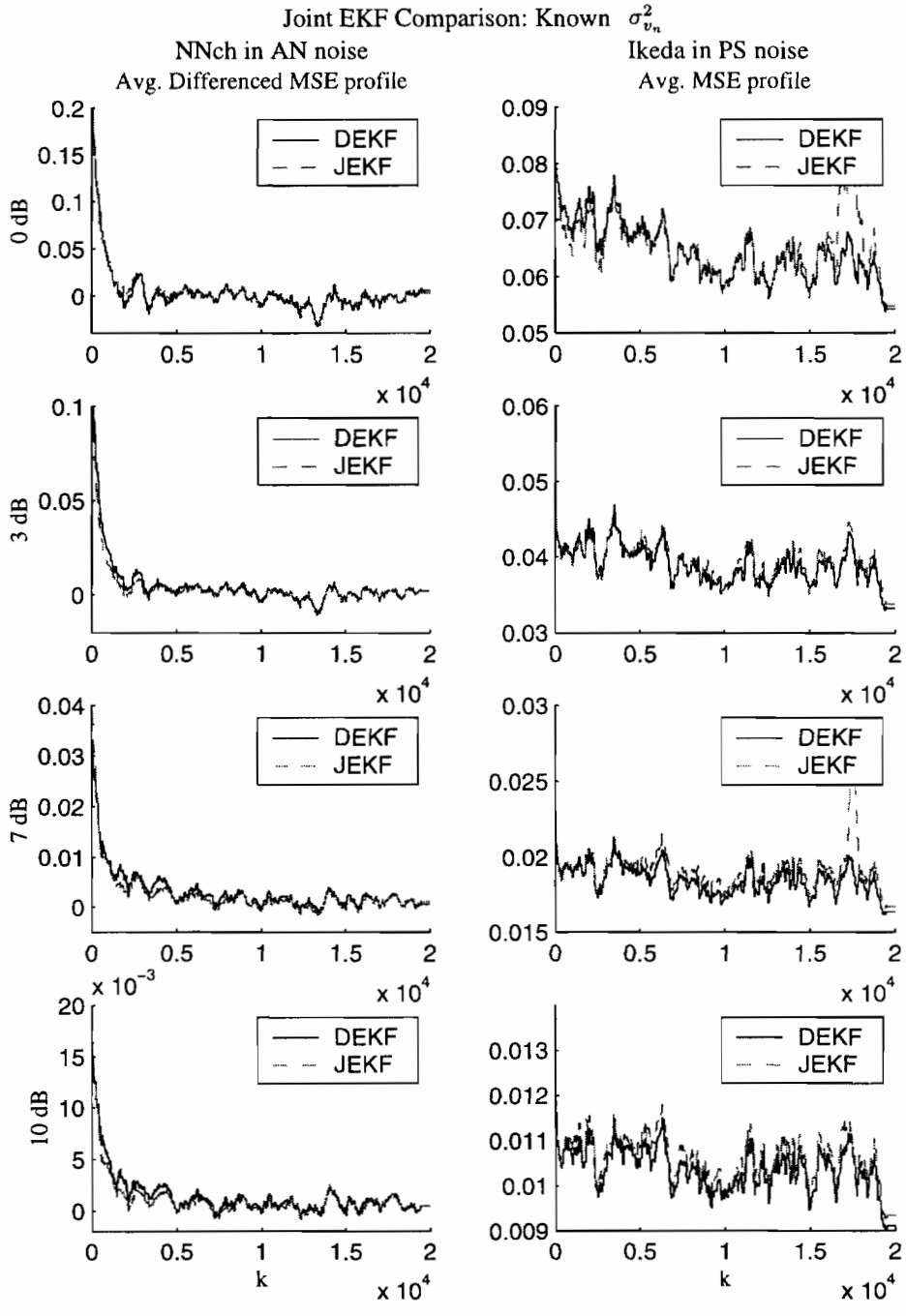


Figure 4.51: Averages of the differenced signal MSE profiles on NNch(AN) data (left), and the signal MSE profiles on the Ikeda(PS) data (right). Only the measurement noise statistics are known. In each plot, the joint EKF is compared against the best dual EKF cost, as listed in Figure 4.50.

first three columns (the limit cycle data were not tested). On the WS noise examples, the p-value is less than 2% for SNRs of 3dB and higher; the DEKF and JEKF results remain statistically equivalent at 0dB. On the chaotic neural network data (second and third columns), the relative performance of the JEKF improves noticeably from the known variance case.

On the Ikeda data, the DEKF retains its significant advantage, although the JEKF appears to perform more consistently than before, and at a smaller deficit in its average NMSE than when the variance is (incorrectly) known. This last observation agrees with the conjecture that the JEKF is not robust to inaccuracies in σ_v^2. Nonetheless, as shown by the average MSE profiles in the right side of Figure 4.51, the JEKF is less robust on the Ikeda data than the DEKF; at 0dB and 7dB SNR, noticeable spikes appear in the JEKF profiles near $k = 18,000$. Meanwhile, the left side of the figure shows the JEKF's superior convergence properties on the chaotic neural network data in nonstationary noise (at most SNRs).

4.9.3 JEKF: Both Variances Unknown

Finally, the joint EKF and dual EKF are compared when both noise variances are estimated online. Figure 4.52 shows that on the white noise data (left column), the JEKF is significantly less robust to inaccuracies in σ_n^2 than the dual EKF algorithm: the dual EKF with $J^{ml}(\mathbf{w})$ shows a definite advantage at all SNRs. On the autoregressive noise (AS and AN, middle columns), as well as on the Ikeda series in pink noise (right column) the relationship between the two algorithms shows little change from when $\sigma_{v_n}^2$ is known: the joint EKF maintains its advantage on the neural network series, while the dual EKF does significantly better on the Ikeda data.

Figure 4.53 shows the ensemble averages of the differenced MSE profiles for the JEKF and dual EKF on the AR-10 (WS) data and neural network (AS) data. Note that the relative performance of the joint EKF appears to improve with increasing SNR.

The experimental results in this section can be summarized by the following observations:

1. The performances of the two algorithms are similar when both the model structure and measurement noise variance are known exactly, although the joint EKF shows a slight (usually insignificant) advantage.

2. On the Ikeda data, the dual EKF algorithm provides significantly better performance. Here, the nonlinearities are more severe, and the appropriate model structure for the neural network is uncertain.

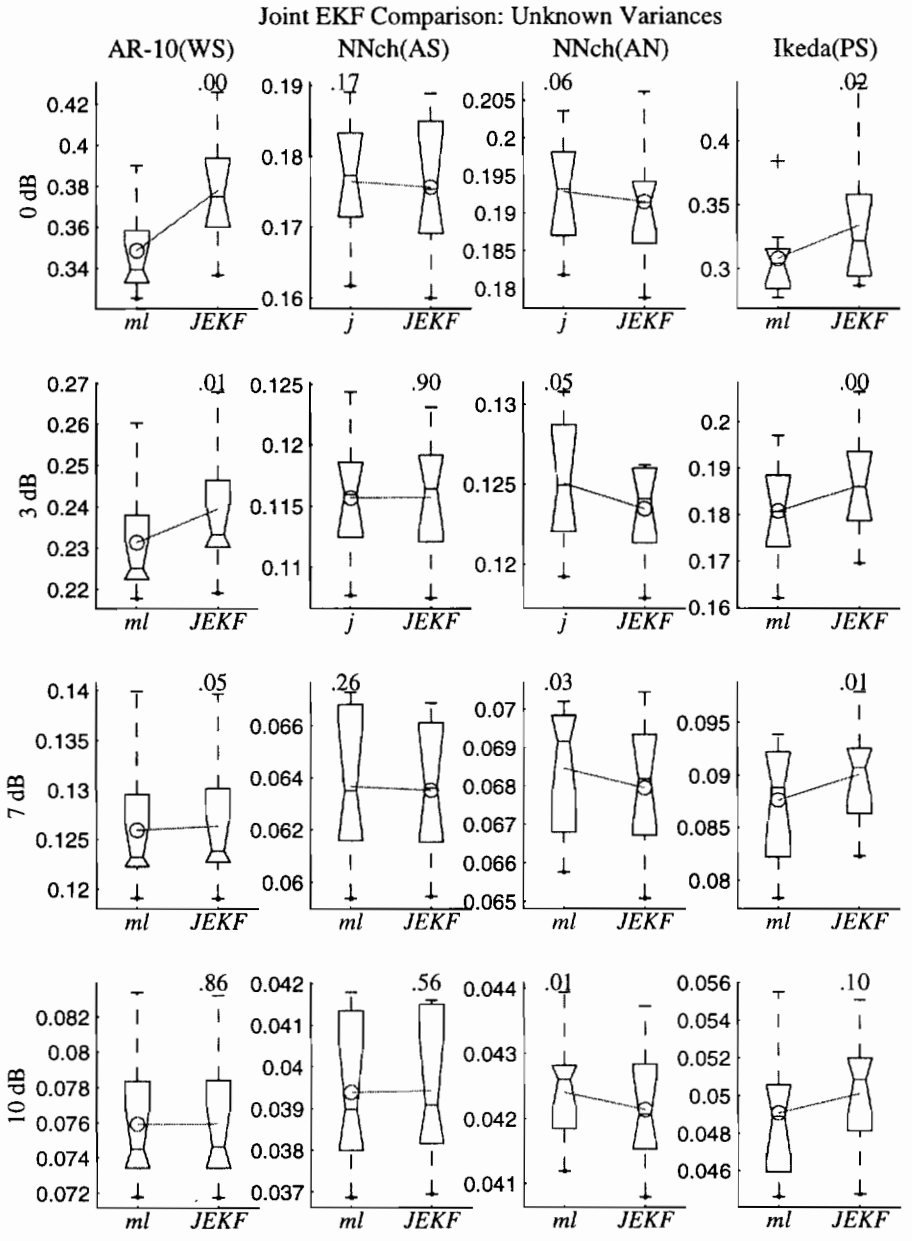


Figure 4.52: The performance of the joint EKF compared with the best dual EKF cost functions, when estimating both noise variances. Boxplots show the final 1000-point signal NMSEs on all data sets, except on the chaotic NN (AN) data (fourth column), in which the overall signal NMSE is used.

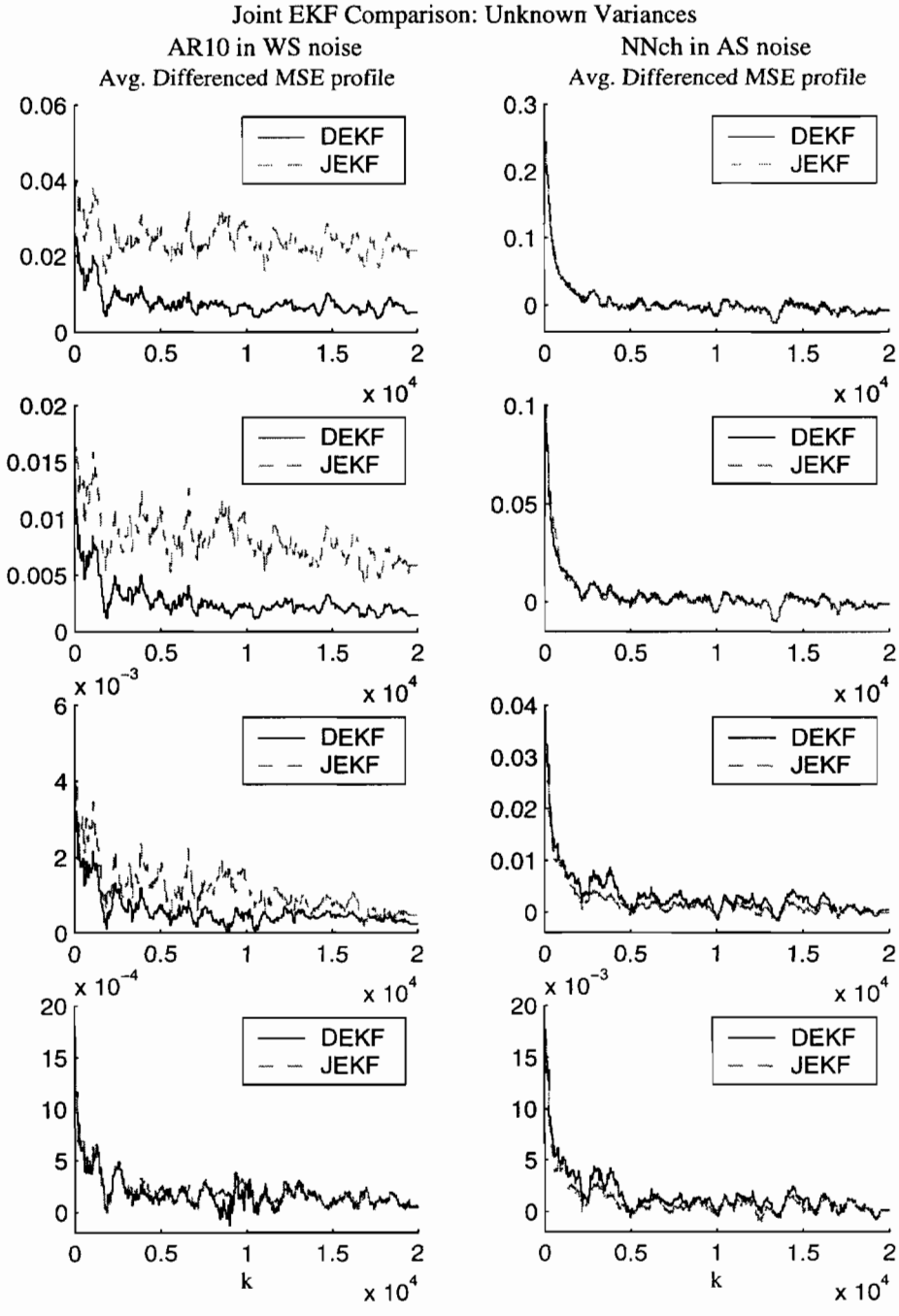


Figure 4.53: Averages of the differenced signal MSE profiles on AR-10(WS) data (left), and neural network (AS) data (right). In each plot, the joint EKF is compared against the best dual EKF cost, as listed in Figure 4.52.

3. When the process noise variance is unknown, the joint EKF gains a significant advantage for white noise at medium to high SNRs. Although the stationary colored noise results do not favor either algorithm, the results in nonstationary noise indicate the joint EKF possesses superior tracking performance when σ_v^2 is unknown. The dual EKF maintains its advantage on the Ikeda data.

4. When both variances are unknown, the advantage of the joint EKF on WS noise is completely reversed. However, the algorithms remain indistinguishable on AS noise, and the joint EKF exhibits better robustness on AN noise. The ranking of the Ikeda results is mostly unchanged.

Overall then, the joint EKF appears more sensitive than the dual EKF to factors that increase estimation error, such as: low SNR, incorrect noise variances, uncertain model structure, and highly nonlinear dynamics. In fact, all of these effects can be interpreted in terms of the additional source of nonlinearity – and hence, linearization error – inherent to the joint EKFs concatenated state-space realization. The larger the state covariance $\overline{\mathbf{P}}_k$ (relative to the scale of the underlying nonlinearities), the more severe the approximation imposed by the linearization of the EKF. Of course the dual EKF also requires linearization; the point is that the *additional* nonlinearity of the joint state-space model makes this effect more pronounced for the joint EKF.

Nevertheless, the performance of the joint EKF is quite good in many cases. Furthermore, its lack of recurrent derivative computations can mean computational savings, although these are potentially offset by the larger dimension of the joint state vector (see Section 3.6.5 on page 112).

4.10 Experiment 7: Model Mismatch Effects

In most applications, the most appropriate model structure for the data is not known beforehand. The experimental results in this section address the robustness of the algorithms and cost functions in the face of model structure uncertainty.

Experiments are run on the chaotic neural network series corrupted by stationary AR-5 noise, at 3dB SNR. As just shown, the performances of the dual EKF (using $J^{ml}(\mathbf{w})$ and $J^{j}(\mathbf{w})$) and joint EKF can be compared using the known 10-5-1 architecture that generated the signal, and the known noise parameters: $\mathbf{w}_n$, $M_n = 5$. Here, the performances are evaluated using two additional (incorrect) model structures:

1. A 5-2-1 neural network architecture, and $M_n = 3$ order noise model.

2. A 10-8-1 neural network architecture, and $M_n = 10$ order noise model.

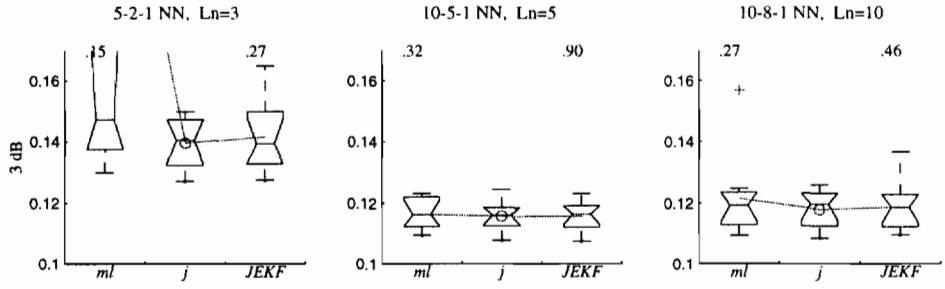


Figure 4.54: The effect of incorrect model structure on the relative performances of the joint EKF and two dual EKF cost functions, when estimating both noise variances. Boxplots show the final 1000-point signal NMSEs on the chaotic neural network series in 3dB AS noise. The middle plot represents the model structure actually used to generate the data.

The first structure is too rigid for the data; the second structure is overly flexible. As an additional source of error, the noise model $\mathbf{w}_n$ is estimated in both cases from a fairly short section (500 point) of the noise data. Both of the variances: σ_v^2 and $\sigma_{v_n}^2$, are estimated on-line using $J^{ml}(\sigma^2)$, as before.

The lefthand plot of Figure 4.54 shows the final 1000-point signal NMSE of the three algorithms on model structure (1); results for the "correct" structure appear in the middle plot; the righthand plot shows the performance with structure (2). As expected, the performance of all algorithms is noticeably degraded by using the inappropriate model size. However, it is also clear that the joint EKF is considerably less robust to these changes than the dual EKF ($J^j(\mathbf{w})$), as indicated by the long top whisker of the joint EKF boxplots. Furthermore, notice that the maximum-likelihood cost function exhibits stability problems for both the underparameterized and overparameterized structures.

The averaged, differenced MSE profiles of $J^j(\mathbf{w})$ and the joint EKF are shown for the two incorrect model structures in Figure 4.55; these profiles can be compared with the corresponding plot in Figure 4.53 on page 189.

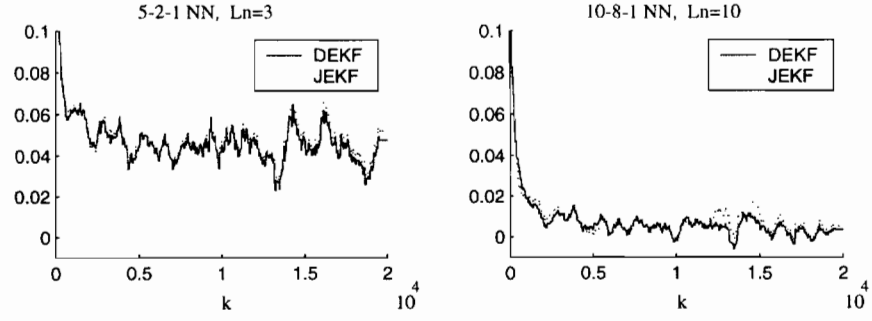


Figure 4.55: Averages of the differenced signal MSE profiles of the dual EKF ($J^j(\mathbf{w})$) and joint EKF for the two incorrect model structures.

4.11 Experiment 8: Over-Training

All of the experiments unto this point are performed in a purely "online" setting, in which each data point is used only once, as soon as it made available. As described in Section 3.6.3, it is sometimes possible to make more than one pass over the data.

In situations where processor speed is high relative to the rate at which observations become available, a large number of training epochs can be performed before the next measurement arrives. In such contexts, *over-training* is a concern because the empirical distribution of the repeated data becomes increasingly biased, relative to the true distribution.

The present set of experiments is designed to investigate several algorithms – including the dual EKF, joint EKF, and a couple of iterative algorithms – with regard to their potential for over-training. As such, the algorithms are evaluated in terms of both *training-set* NMSE, calculated over the portion of the series used during training, and *test-set* NMSE, calculated over data not used during training. The algorithms can be compared by plotting their training and test set NMSEs as functions of the training epoch. Both estimation NMSEs ($x_k - \hat{x}_k$) and prediction NMSEs ($y_k - \hat{x}_k^-$) provide useful information.

The experiments are performed on the Mackey-Glass chaotic series (described on page 127) corrupted by stationary white Gaussian noise at 3dB SNR. A 5-18-1 neural network architecture (5 inputs, 18 hidden units, and one output) is used to model the dynamics, based loosely on findings in [40]. Both noise variances are initialized with the ad-hoc procedure of Section 3.6.2, and are estimated along with the signal and weights using the $J^{ml}(\sigma^2)$ cost function. Initial covariances: $\mathbf{P}_0 = \mathbf{I}$, $\mathbf{Q}_0 = .1\mathbf{I}$, $q_{v,0} = q_{n,0} = .1$, and forgetting factors: $\lambda_{\mathbf{w}} = .9999$, $\lambda_{\sigma^2} = .9993$ are used by the dual EKF and joint EKF.

The dual EKF and joint EKF algorithms are compared against an iterative algorithm similar to that in [10] which alternates between EKF signal estimation and backpropagation model estimation (BP-EKF). The BP-EKF algorithm performs EKF signal estimation followed by 100 epochs of a gradient descent algorithm minimizing the prediction error $(\hat{x}_k - \hat{x}_k^-)^2$. The variances were estimated using the sequential variance filters also used by the dual EKF and joint EKF. Although an iterative generalized EM algorithm was also implemented, it produced relatively poor results, and so is not included in the figures.

A known-model performance benchmark is obtained by training a neural network predictor on 4000-points of the clean series using gradient descent, and stopping when the prediction error on the remaining 1000 points begins to increase (25,300 training epochs are used). The resultant innovations variance for the trained model is 4.8×10^{-04}, which can be used as the value of σ_v^2.

The sample variance of the noise is used to obtain σ_n^2. Using these "known" values of the weights and variances, an EKF is applied to the noisy series to produce benchmark NMSE values against which to compare the various dual estimation algorithms.

Because over-training is affected by the length of the training window, N_{win}, two different situations are considered. In the first, $N_{win} = 2000$ with a test set of the subsequent 3000 points. in the second, $N_{win} = 500$, with 1500 points of test data; in both cases, it is assumed that enough processing power is available to make numerous passes over the data before the next data point arrives.

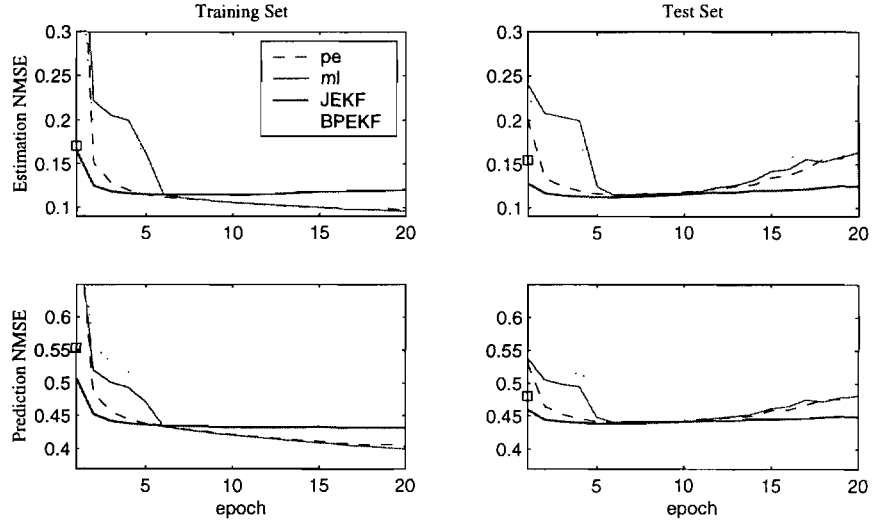


Figure 4.56: The average estimation and prediction NMSE trajectories on noisy Mackey-Glass data, using a 2000-point training-set, and 3000 points of test data. The average EKF result is indicated by a small square.

The average NMSE trajectories for the longer training set are shown in Figure 4.56. The superior training-set performance of the joint EKF and dual EKF with $J^{ml}(\mathbf{w})$ and $J^{pe}(\mathbf{w})$ costs is consistent with the on-line results shown previously for white measurement noise. Nearly all the algorithms exhibit over-training to some extent, as exhibited by the increase in the test-set NMSE after some number of epochs. Unique to the task of dual estimation is the possibility of such an increase appearing on the training set NMSE as well (as exhibited by the joint EKF in this case). This is because of the unsupervised nature of the task; the clean data are not available even during training, so the neural network can begin to model the noise in the training data to some extent.

In some circumstances, a block of data can be used during the training process to monitor out-of-sample prediction NMSE, and control over-training. These data are usually referred to as a *validation set* to distinguish it from a true test-set, which should not be used even for this purpose.

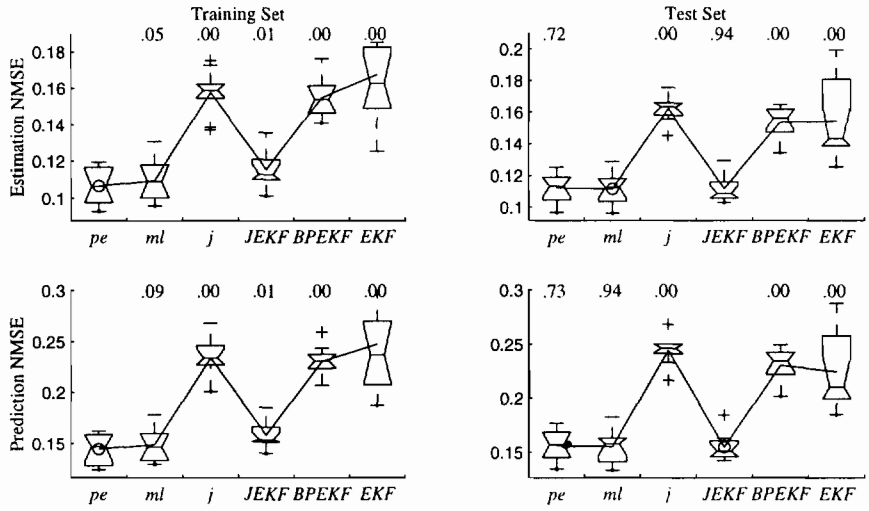


Figure 4.57: Boxplots of estimation and prediction NMSEs, obtained from 2000 points of Mackey-Glass data while using the 3000 point test-set for early-stopping validation.

When available, the validation NMSE can be used to select the training epoch with the lowest error, and thereby avoid over-training. Seeing that the clean signal is not available to compute an estimation NMSE, the prediction NMSE can be used, where the prediction error is defined relative to the noisy data, as: $(y_k - \hat{x}_k^-)$. In Figure 4.56, the joint EKF and dual EKF with the $J^{ml}(\mathbf{w})$ and $J^{pe}(\mathbf{w})$ costs represent the best methods if the 1500 point "test data" are used for validation in this way. The boxplots in Figure 4.57 show the relative performance of the algorithms under this assumption. Note that the EKF with the known model is significantly outperformed by several of the dual estimation algorithms. As discussed in the on-line case, this is most likely due to the suboptimality of the EKF on nonlinear data.

However, it is not always possible to provide a validation set. In particular, when the amount of data is limited, there will not be enough to train on and still provide a reliable estimate of performance on unseen data. One scenario in which this situation arises is in the use of short windows to process speech, or other time-varying signals. As shown in Figure 4.58, the problem of over-training is exacerbated by a shorter training window of 500 points.

Hence, it is precisely when a validation set is most needed (*i.e.*, with scarce data) that one is least likely to be available. In these situations, one might consider using the dual EKF with $J^j(\mathbf{w})$ because it appears to be less susceptible to over-training. However, the joint cost clearly produces suboptimal performance on these data (white noise). A better result can sometimes be obtained by using the maximum-likelihood cost or the joint EKF algorithm, and stopping early at some previously-chosen epoch.

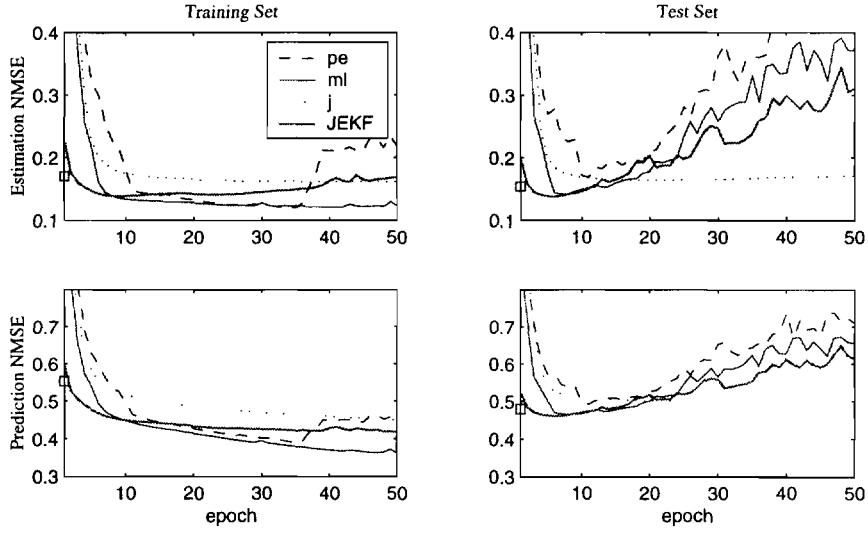


Figure 4.58: The average estimation and prediction NMSE trajectories on noisy Mackey-Glass data, using a 500-point training-set, and 1500 points of test data. The BP-EKF results are not shown.

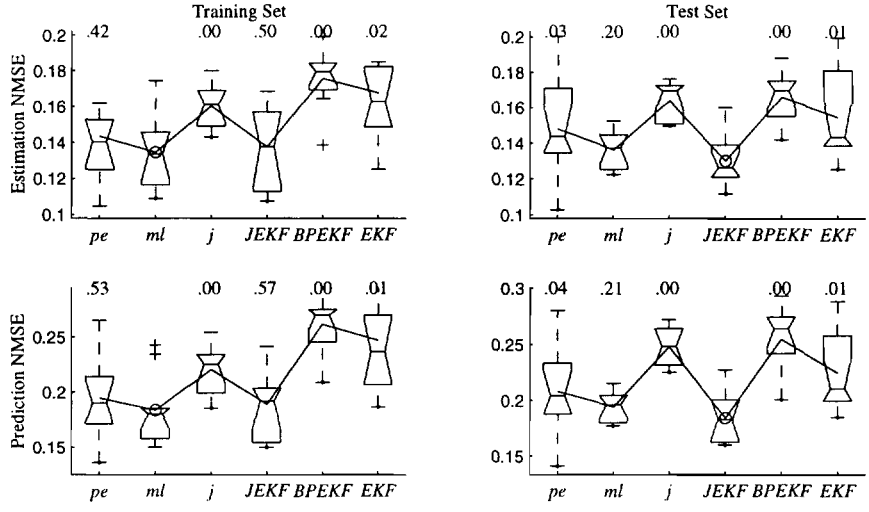


Figure 4.59: Boxplots of estimation and prediction NMSEs, obtained from 500 points of Mackey-Glass data while using the 1500 point test-set for early-stopping validation.

Note that even if the particular application does not require generalization to new data, the issue of over-training is still significant; the dual-estimation algorithm can begin to incorporate the noise into its signal estimates, so that performance begins to degrade after some number of epochs. Although the $J^{ml}(\mathbf{w})$ cost continues to minimize the training set NMSE in the preceding examples, this will not generally occur. In most applications over-training is of real concern, and should be guarded against if possible.

A suitable choice of forgetting factor can be used to help prevent over-training. As described

previously, a choice of $\lambda < 1$ effectively controls the amount of data used for parameter estimation. The forgetting factor $\lambda_{\mathbf{w}}$ used in this set of experiments corresponds to a time-constant of around 10,000 points for the data window (see Figure 3.3 on page 59). A time-constant of around 2,500 points might have prevented some of the over-training seen in Figure 4.58. However, early-stopping is a more direct approach to the problem.

4.12 Discussion

The experiments in this chapter, while fairly extensive, are necessary for making some important observations about the dual EKF and joint EKF.

- Estimation of the noise variances can generally be done very successfully by the dual EKF, and is best performed by minimizing the maximum-likelihood variance cost functions.
- For weight estimation, $J^{ml}(\mathbf{w})$ can produce the best results in many situations, but it is unfortunately prone to numerical problems related to the inversion of the approximate Hessian.
- For white noise data, the prediction-error cost $J^{pe}(\mathbf{w})$ produces very good estimates (often not significantly worse than $J^{ml}(\mathbf{w})$), without stability problems.
- In colored noise, the joint cost $J^{j}(\mathbf{w})$ gives good, stable solutions, sometimes better than the less stable $J^{ml}(\mathbf{w})$ cost.
- The $J^{em}(\mathbf{w})$ and $J^{ec}(\mathbf{w})$ methods do not work well in general. This may be due to the approximations made by the dual EKF, as described in Chapter 3, or in development of the costs themselves. For example, the error-coupled cost is developed using a Gaussian assumption on the dynamics error $\tilde{f}_k$, and the EM cost is usually computed noncausally using a smoother, rather than in the sequential manner of the dual EKF.
- The joint EKF is not seen to suffer from the convergence problems reported in the literature. However, its higher sensitivity to inaccuracies in noise variances and model structure information makes it a less robust alternative to the dual EKF in real-world dual estimation applications.
- Iterating the dual EKF over the same data set is likely to cause over-training. Early-stopping is a must, and can be implemented with either a cross-validation approach, or using a predetermined number of iterations (*e.g.*, 5 epochs). This issue is explored further in Chapter 5.

In the next chapter, the above observations are put into practice when approaching several time-series estimation and prediction problems.

Chapter 5

Applications

The controlled experiments in the previous chapter provide an empirical comparison of cost functions and algorithms on several of different types of time-series data. In this chapter, many of the experiments described are performed on real-world data, for which the clean signal is not available. A range of application domains are considered, including estimation of river flow, enhancement of speech, and prediction of economic time-series.

The purpose of these experiments is to illustrate the use of the dual EKF in some realistic signal processing settings, and to demonstrate the potential advantage of the dual estimation approach. Some of the applications considered here have been studied extensively in the literature, with researchers incrementally improving their results over the years using a variety of model structures and training methods. The purpose of this chapter is not necessarily to supersede these published results; in some cases this is unlikely, as the experiments herein are limited to the autoregressive model structure discussed in Chapter 1 of this thesis. However, by showing the advantage of the dual EKF used with AR models, the results in this chapter underscore the potential of the dual EKF to improve upon previously published results when used with the alternative model structures described in the literature.

Because the clean signal and true model are generally not available in these experiments, the only objective criterion is prediction error; in applications for which estimation error is of primary concern, only a subjective evaluation of the results is possible. An exception is the speech enhancement section, in which several controlled experiments are included (with the clean speech available) in addition to actual recordings of noisy speech. Results of another controlled experiment – on a known discrete-time chaotic map – are presented in the next section.

5.1 Chaotic Hénon Map

The study of chaos has far reaching applications in the study and analysis of real-world systems. A chaotic system can be characterized by the dimensionality and appearance of its attractor. A model of a chaotic system can therefore be evaluated in terms of its ability to reproduce the attractor of the original system [84]. This first experiment considers a well-known, but artificial, chaotic system, and demonstrates the benefit of a dual estimation perspective of modeling the dynamics in the presence of noise.

In 1976, Michele Hénon proposed the following system of equations for modeling chaos in two dimensions:

$$a_{k+1} = 1 - 1.4 \cdot a_k^2 + b_k \tag{5.1}$$

$$b_{k+1} = 0.3 \cdot b_k. \tag{5.2}$$

The map takes points (a, b) through three successive transformations: a bending; a compression in the a-direction; and a reflection through the diagonal, $a = b$. 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 (in the upper left part of Figure 5.1) shows the chaotic attractor. A neural network can be trained as a predictor on this signal, using an EKF training algorithm. The network is then iterated – feeding back the predictions of the network as future inputs – to produce the attractor shown in the upper right plot. The individual data points are of course not the same as the original data, but it is clear that the dynamics have been captured by the (5 input, 7 hidden unit) neural network.

However, if the signal is corrupted by white noise at 10dB SNR, and a neural network with the same architecture is trained on these noisy data, the dynamics are not adequately captured. The iterated predictions of the neural network trained on noisy data are shown in the bottom left part of the figure. While the general outline of the original attractor is apparent, the dynamics 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 the bottom right plot of Figure 5.1. 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 .1, and the initial signal covariance is $\mathbf{P}_0 = \mathbf{I}$. Forgetting factors are: $\lambda_{\mathbf{w}} = .9999$, and $\lambda_{\sigma_v^2} = .9993$. As with both of the EKF-trained networks, a separate validation set is used for early-stopping. Although the attractor is not reproduced with total fidelity, its general structure has been extracted

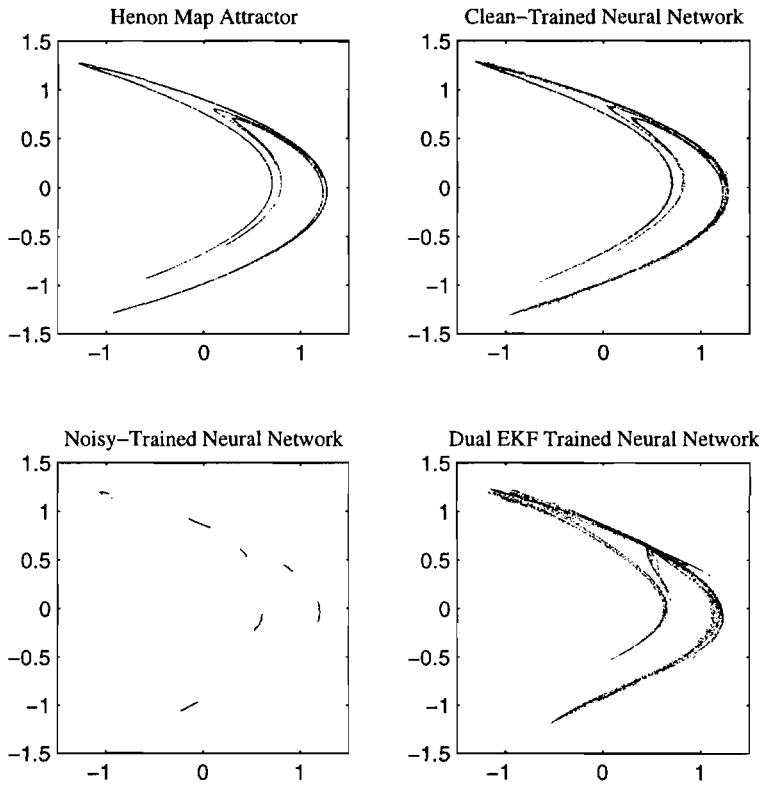


Figure 5.1: Phase plots of: x_{k+1} versus x_k for the original Hénon series (top left); the series generated by a neural network trained on x_k (top right); the series generated by a neural network trained on y_k (bottom left); the series generated by a neural network trained on y_k, using the dual EKF (bottom right).

from the noisy data.

5.2 Willamette River Flow

The data in this experiment were first published by Percival and Walden ([64], 1993), and are available in digital form at the Carnegie Mellon University StatLib web site [9]. The data consist of the log of the monthly average flow in the Willamette River, as measured daily near Salem, Oregon for about 33 years. The series, shown at the top of Figure 5.2, contains only 395 points, so the dual EKF must be iterated over the data to obtain a solution.

The problem is made more difficult by the limited amount of data, and the lack of prior information about its collection, such as the reliability of the sensors, or an estimate of the noise which might have been gleaned from the original daily measurements. Nonetheless, the data are useful for demonstrating the dual estimation approach.

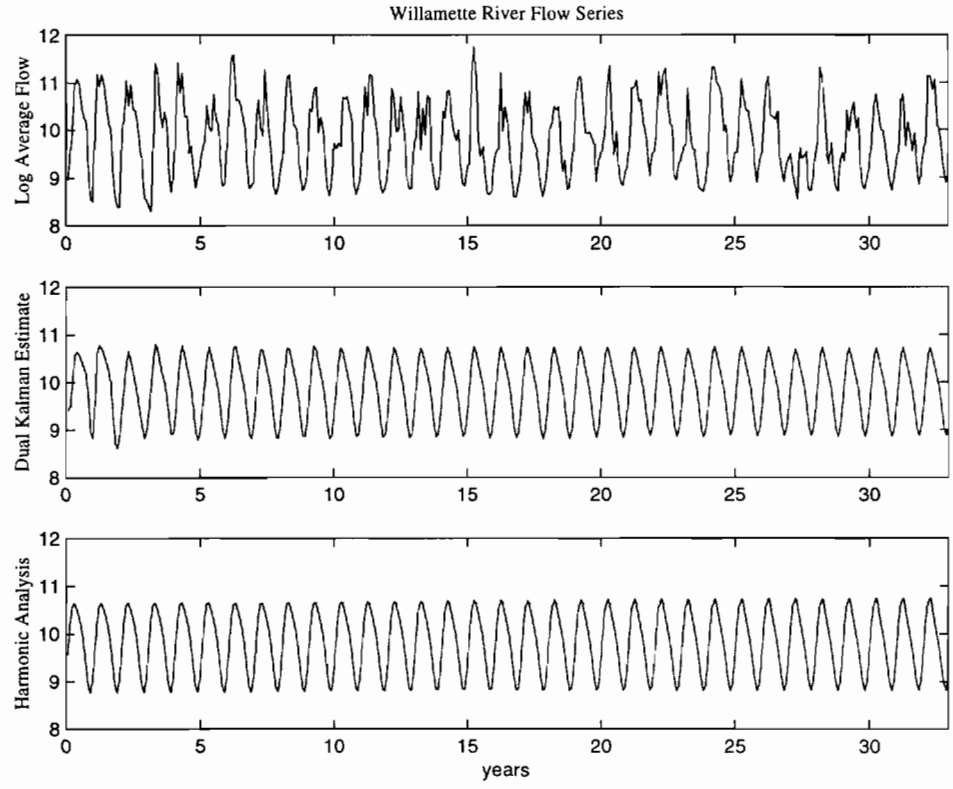


Figure 5.2: The log of the monthly average Willamette River flow, as measured daily near Salem, Oregon (top). The dual EKF estimate of the series (middle) captures its annual periodicity, and agrees well with the result published in [64] using a harmonic analysis approach (bottom).

The first challenge is to select a suitable model structure for the data (see Figure 1.3 on page 4). With so few data available, a proper model validation set is not a possibility. However, a small validation set (*e.g.*, the last 95 points) can be used as a sort of guide during the model selection process. A model can be selected by trying several structures, and picking one that seems to extract as much of the structure in the signal as possible, and provide reasonable generalization in the validation set. This trial and error process should ideally be replaced with a more rigorous approach, but it produces good results, nevertheless.

In particular, the middle plot of Figure 5.2 is obtained with the dual EKF using a 20 input, 5 hidden unit, single output neural network, the maximum-likelihood costs for weight and variance estimation. Forgetting factors: $\lambda_{\mathbf{w}} = .9997$ and $\lambda_{\sigma^2} = .9993$ are used to control over-training. 10% of the data are chosen randomly for cross-validation, to determine that iteration should be stopped after 5 epochs. The algorithm is subsequently trained with the full data set to obtain a prediction NMSE of 0.3258 and an estimation residual $(y_k - \hat{x}_k)$ with variance 0.1801 (unnormalized). The estimate of the measurement noise variance is around $\hat{\sigma}_n^2 = 0.184$.

Results published in [64] using statistical harmonic analysis provide an additional, external form of validation. As shown by the bottom plot of Figure 5.2, the dual EKF results agree quite well with those in [64]. Both approaches uncover the seasonal regularity of the data; moreover, the dual EKF estimate contains some additional structure and variation among the annual cycles. Note that the estimation residual is comprised of both actual measurement noise, and unmodeled global climatic and weather fluctuations. These additional factors are considered to be noise only in the sense that the data set is insufficient to model their dynamics.

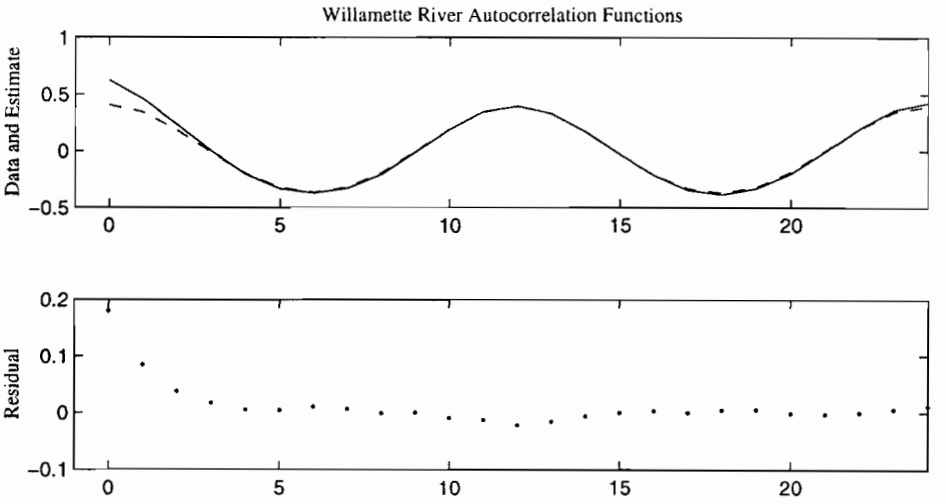


Figure 5.3: Autocorrelation of the river flow data (solid line, top plot), estimates (dashed line, top) and residuals (bottom plot) over two annual cycles. The dual EKF is able to extract the periodic structure of the data, and produces a residual with little temporal correlation (nearly white).

The autocorrelations of the original data, the dual EKF estimates, and the estimation residuals are provided in Figure 5.3. The strong periodicity of the data is indicated by the autocorrelation of the time-series and signal estimates in the top plot. The bottom plot shows that the autocorrelation of $(\hat{x}_k - y_k)$ is strongly peaked at the 0^{th} tap, suggesting that there is very little temporal structure left in this residual.

5.3 Sunspot Prediction

Since the year 1700, the number of sunspots visible from the Earth have been counted and recorded on an annual basis. A method devised by Rudolph Wolf incorporates the number of sunspots and the number of sunspot groups, and continues to be used today (see Figure 5.4). Daily numbers are computed as a weighted average of observations at locations around the globe, and can be summed to produce the annual series. Because the number of sunspots is a good indicator of solar activity,

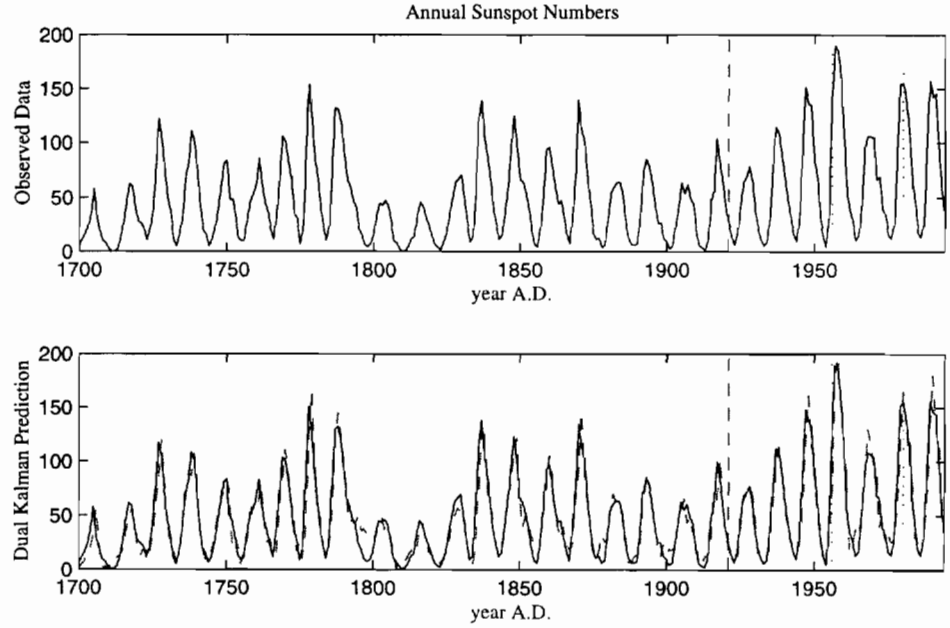


Figure 5.4: The annual sunspot series, from 1700 to 1994 (295 points). Data up to 1920 are typically used for training, and the remaining data are divided into three test sets: 1921-55, 1956-79, and 1980-1994. The bottom plot shows estimates (solid line) and predictions (dashed line) generated by a dual KF, using a linear model and the $J^{pe}(\mathbf{w})$ cost.

the series has been the focus of much study over the years (*e.g.*, see [97, 91, 86]). Of course, more accurate indicators of solar activity are in use today, but none with as lengthy an historical record.

However, even with three centuries of annual sunspot numbers, there is not enough data to build good models of the series. Furthermore, there are good reasons to believe that significant measurement noise exists in the data: the sunspot numbers derive from a crude integer count of an underlying process which is continuously valued; counts are highly subjective and depend on the atmospheric conditions between the sun and the observer; the series does not differentiate between larger and smaller diameter sunspots; other relevant information, such as the duration of the sunspots, is not considered. These errors make the data somewhat stochastic, and a good candidate for a dual estimation approach.

Recent approaches to sunspot prediction in the literature include linear AR models, as well as neural network predictors [91], and committee machines [86]. Typically, the 221 points from 1700 through 1920 are used for training the predictor, and the remaining data are used for testing purposes. The test data is often subdivided into three parts: 1921-55, 1956-79, and 1980-94, with prediction MSEs reported on each of these periods separately, as well as on the entire test set. To facilitate comparison with previously published results, the MSEs are all divided by the constant

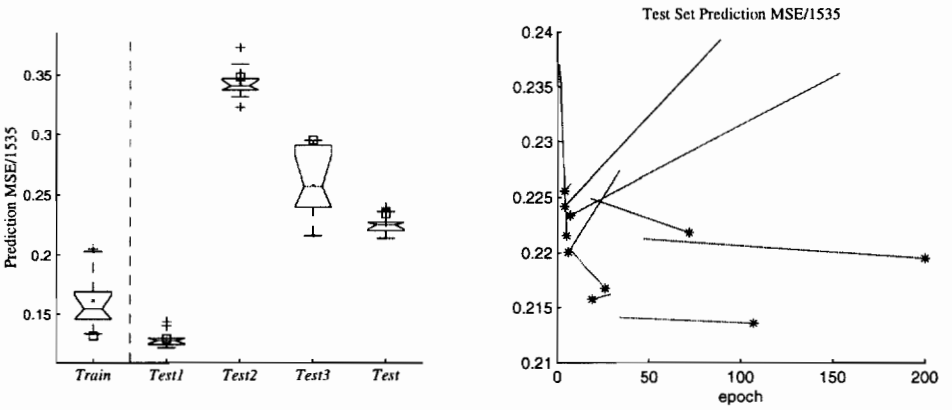


Figure 5.5: The boxplot on the left shows the performance of the dual KF over ten runs, compared with that of a standard AR-12 predictor (squares). "Test1","Test2", and "Test3", correspond to the three test subsets listed in the text. In the right plot, the optimal errors and early-stopping epochs (*) are shown, connected with lines to the corresponding errors and epochs returned by the cross-validation approach.

1535, instead of a traditional NMSE measure.

For the dual Kalman filter approach, a linear AR-12 model is chosen. Although nonlinear model structures have been used successfully in the literature, they tend to incorporate nonstandard architectural features which make them difficult to reproduce here. The linear AR-12 model produces reasonably good predictions, is sufficient for demonstrating the dual Kalman approach.

The prediction-error cost function is used for weight estimation, and the maximum-likelihood cost is used for estimating both variances. Parameter covariances are initialized at .1, and the initial signal covariance is $\mathbf{P}_0 = \mathbf{I}$. Forgetting factors of $\lambda_{\mathbf{w}} = .9993$ $\lambda_{\sigma_v^2} = .999$, $\lambda_{\sigma_n^2} = .999$ are selected to control over-training on the extremely short data set. The series is scaled to fall between 0 and 1, but is not otherwise normalized.

The dual KF is iterated over the training set, with early-stopping implemented by the cross-validation approach described in Section 3.6.3 on page 108: 10% of the training set is held out. Because the validation set is selected at random, results vary from one run to the next. Therefore, the experiment is repeated 10 times, and the resultant prediction MSEs (normalized by 1535) are shown graphically in Figure 5.5. Also shown are the MSE values for a standard AR-12 model, trained on the noisy data using a forward-backward least-squares approach; these MSEs are indicated by superimposed squares in the boxplot. Although the dual KF results are generally better than for the least-squares AR-12 model, the variance is fairly high, and the performance is actually worse on some individual runs. A large part of the problem is indicated in the right plot of Figure 5.5; this shows the optimal test set MSE/1535 of each of the ten runs, positioned at

the appropriate epoch. A line is drawn from this point to the epoch and MSE actually returned for that run using the cross-validation scheme. The length and steep slope of several of these lines indicate that the attempt at early-stopping is not very effective. In some cases training was stopped more than 100 epochs too early, while in other cases training was stopped too late.

As an alternative, the cross-validation approach is abandoned, and the dual KF is stopped after 5 epochs, based on the results of the iterative Mackey Glass experiment in the previous chapter. Again, this has the advantage of allowing all the training data to be used for model adaptation. As always, the model is initialized using a least-squares fit to the noisy data, so there is no variation between runs in this case. The results are shown in Table 5.1, along with those for the standard least-squares predictor, and the average of the cross-validation results over 10 repetitions.

Table 5.1: Sunspot prediction MSE/1535. Standard AR-12 predictor results are compared with the dual KF using a 10% cross-validation set (CV), and using 5 epochs of training.

	Train	Test1	Test2	Test3	Test
AR-12	0.1319	0.1295	0.3485	0.2951	0.2341
Dual KF (CV)	0.1614	0.1295	0.3437	0.2578	0.2250
Dual KF (5 ep)	0.1374	0.1257	0.3518	0.2431	0.2228

As with the river flow data, the autocorrelation of the estimation residual $(y_k - \hat{x}_k)$ can be used to determine the amount of left-over structure not contained in the signal estimates. Similarly, the autocorrelation of the prediction error $(y_k - \hat{x}_k^-)$ shows the degree to which the predictions can be improved. As presented in Figure 5.6, these autocorrelations show that the dual KF has captured

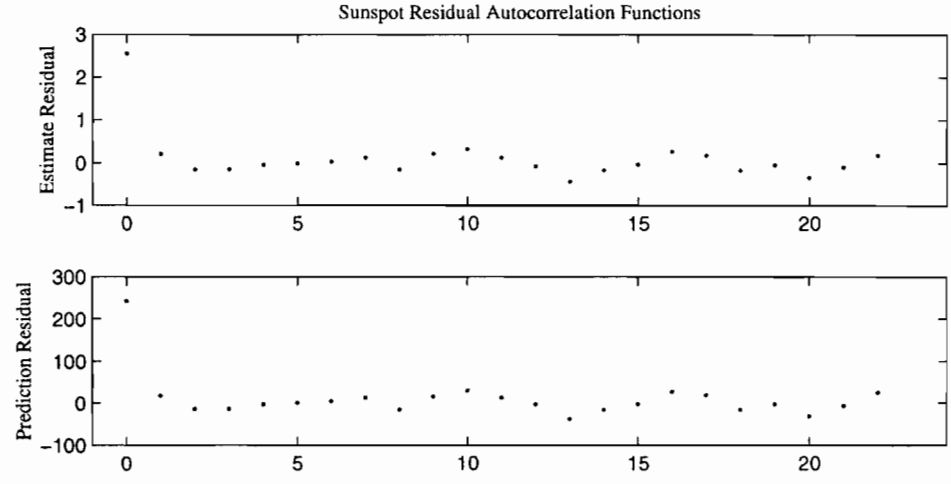


Figure 5.6: Autocorrelation functions of the sunspot estimation residual (top) and prediction error (bottom) over two 11 year cycles show little temporal structure in the terms: $(y_k - \hat{x}_k)$ and $(y_k - \hat{x}_k^-)$, respectively.

most of the structure in the data, leaving nearly white estimation and prediction residuals.

This experiment underscores the difficulty associated with building predictive models from short data sets. The dual Kalman filter has the potential to produce more accurate predictions, but it requires that attention be given to the problem of over-training. These issues are reinforced in the next experiment on a macroeconomic time-series.

5.4 Index of Industrial Production

The level of economic activity in the country is of great interest to policy makers and companies, as it influences many aspects of our lives, such as the unemployment rate, the stock market, demand for goods and real estate, inflationary pressures, and the general mood of the populace.

The two primary measures of economic activity in the U.S. are the *gross domestic product* and 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. Moody ([55], 1995) cites several reasons for the difficulty in forecasting such series. Among them are: the lack of prior (analytical) models for the data; high levels of noise due to unmodeled disturbances and inexact survey techniques; nonstationarity due to changes in the world economy and changes in the definition of the series itself; and the possible nonlinearity of the dynamics, which makes simpler linear modeling techniques inadequate.

Nonetheless, many economists have used linear regression techniques to build empirical models that predict the IP using several other economic series as inputs. An important baseline approach is to predict the IP from its past values, using a standard autoregressive model; *e.g.*, results with an AR-14 model are reported by Moody et al. ([56], 1993). This linear AR model is well suited

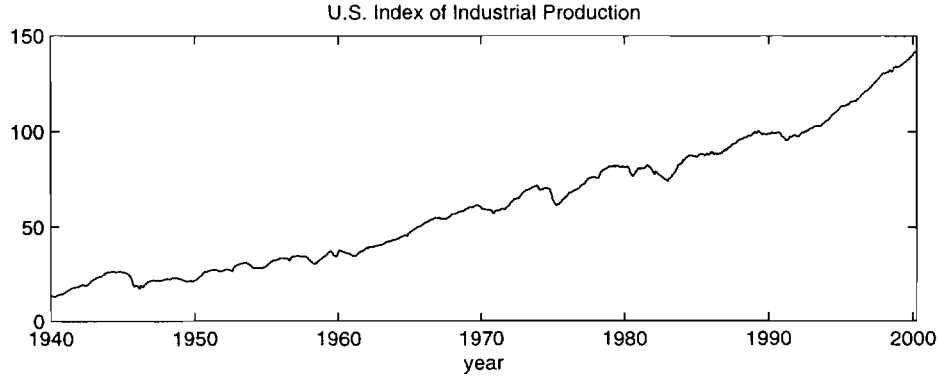


Figure 5.7: Index of Industrial Production in the U.S., from January 1940 through March 2000. Data available from Federal Reserve [20].

for investigating the benefit of dual estimation to this problem, and it is tested in the following experiments along with a neural network model. The high level of noise, and the presence of nonlinear dynamics make the IP an excellent candidate for testing the dual EKF.

The monthly IP data is shown in Figure 5.7. To remove the trend, the differences between the $\log_2$ values for adjacent months are computed. This is called the IP monthly rate of return, and is shown at the top of Figure 5.8 for January 1950 to January 1990.

Both a linear AR-14 model and neural network (14 input, 4 hidden unit) model are tested. Consistent with experiments reported in [56], 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.

To obtain the predictions shown in Figure 5.8, the weights are estimated with the joint cost $J^j(\mathbf{w})$; the costs $J^{pe}(\mathbf{w})$, $J^{ml}(\mathbf{w})$ are also tested. Both the noise variances are estimated using the maximum-likelihood cost: $J^{ml}(\sigma^2)$. All initial parameter covariances are set to .1, and the initial

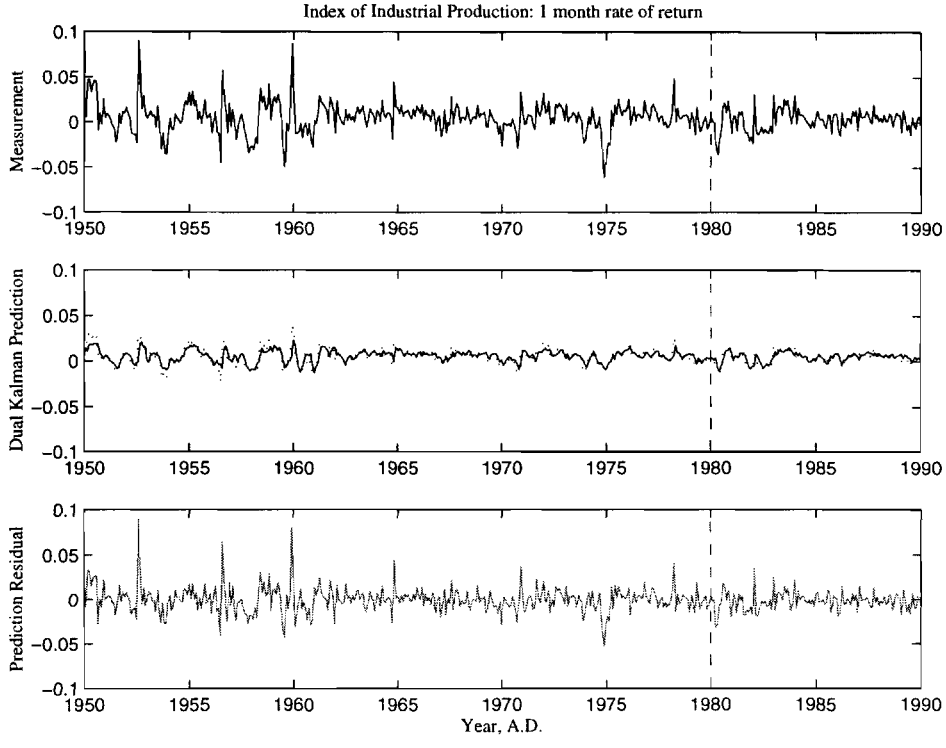


Figure 5.8: Monthly rate of return of the Index of Industrial Production in the U.S., 1950-1990 (top). The dual KF prediction for a typical run (middle), is shown along with the signal estimates (dotted line). The prediction residual is also shown (bottom).

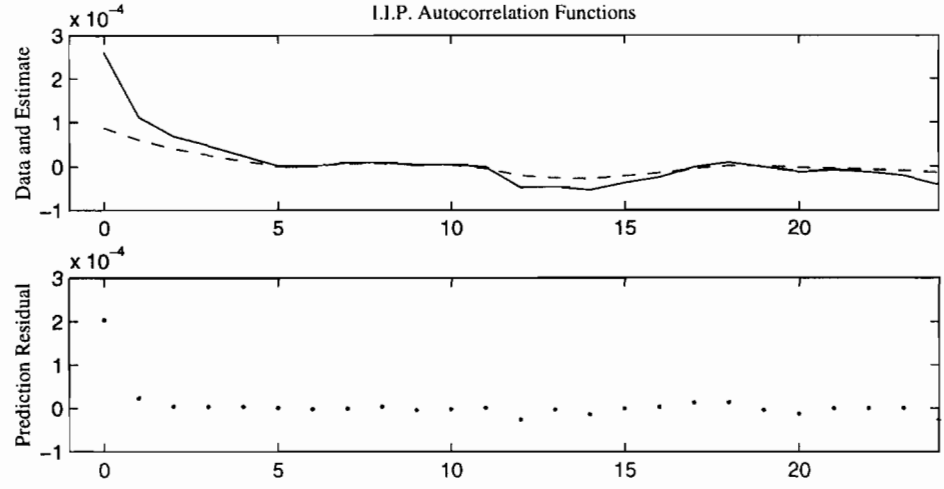


Figure 5.9: Autocorrelation functions of the Index of Industrial Production time-series (solid) and signal estimate (dashed) over two years (top plot). The autocorrelation of the prediction error is also shown (bottom plot).

signal covariance is $\mathbf{P}_0 = \mathbf{I}$. As in the previous examples, the autocorrelation functions (shown in Figure 5.9) show very little structure in the prediction error $(y_k - \hat{x}_k^-)$.

As expected from Section 4.11, over-training is a serious concern because the algorithm is being run repeatedly over a very short training set (only 360 points). The scarcity and nonstationarity of the data makes the use of a validation set highly problematic. Based on experience with other types of data, and the results in Experiment 8, only 5 training epochs are used.

Nonetheless, the effect of over-training is shown in Figure 5.10, for the neural network model with the maximum-likelihood and prediction-error costs, in particular. The experiment is repeated 10 times with different initial weights, $\hat{\mathbf{w}}_0$, to produce the boxplots in the left part of the figure. The result of training an AR-14 model with least-squares (LS) is included as a benchmark, and clearly indicates the advantage of dual estimation. Results for a neural network predictor trained with an EKF weight filter on the noisy data indicate that there is little advantage to using a nonlinear model on the original series. However, the dual EKF with $J^j(\mathbf{w})$ cost produces significantly better results with the neural network, although the potential for over-training actually hurts the performance of the $J^{pe}(\mathbf{w})$ and $J^{ml}(\mathbf{w})$ costs.

Although better results are reported on this problem [56] using models with external inputs from other series, the dual EKF results are quite competitive. While the dual EKF can in principle be applied to models that incorporate exogenous inputs, the investigation of these possibilities is beyond the scope of this thesis.

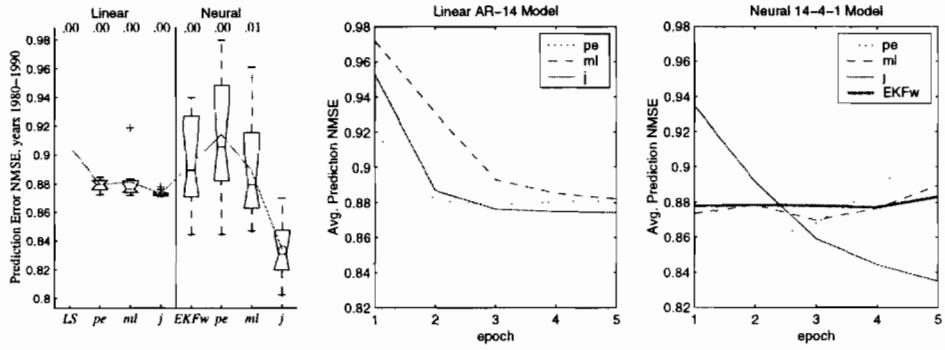


Figure 5.10: Boxplots of the prediction NMSE on the test set (1980-1990) are shown in the left plot. The middle and right plots show average convergence behavior of linear and neural network model structures, respectively.

5.5 Speech Enhancement

This section considers the removal of noise from speech signals. Speech enhancement has many applications, ranging 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. [33]) such methods frequently result in audible distortion of the signal, and are somewhat unsatisfactory in real-world noisy environments.

Recent neural network based filtering methods utilize data sets where the clean speech is available as a target signal for training. These methods are often effective within the training set, but tend to generalize poorly for actual sources with varying signal and noise levels (a review of neural based approaches can be found in [89]). Furthermore, the network models in these methods do not fully take into account the nonstationary nature of speech.

5.5.1 Dual Estimation Approach

The dual estimation algorithms developed in this thesis have the advantage of generating estimates using only the noisy signal itself. To address its nonstationarity, the noisy speech is windowed into shorter, approximately stationary sections, as described in Section 3.6.4 on page 110. The dual estimation algorithms are then iterated over each window to generated the signal estimate. Effectively, a *sequence* of time-series models is trained on the specific noisy speech signal of interest, resulting in a nonstationary model which can be used to remove noise from the given signal. For linear models, $f(\cdot)$, this basically reduces to the classic *linear predictive coding* (LPC) model of speech.

This approach to speech enhancement poses some problems for dual estimation algorithms. First, it is clear that the appropriate model structure will vary (with the complexity of the signal dynamics) from one window to the next, depending on whether the current window comprises, for example, a fricative, vowel, or silent interval (noise-only). However, the nature of the application generally precludes (or at least makes undesirable) the use of a system identification loop (see Figure 1.3 on page 4) to determine the model structure. Ideally, a Bayesian approach to model selection might be used as an integral part of the dual estimation process, but this is beyond the scope of this thesis. Hence, in this section, a model structure is chosen which in some way is a compromise between the various levels of dynamic complexity encountered in the signal.

A second difficulty inherent to speech signals is that they contain long segments of silence, for which the process noise variance is effectively zero. However, a Kalman filter will diverge if $\sigma_v^2 = 0$, because the data get completely ignored, and numerical inaccuracies accumulate. This problem is usually overcome by setting σ_v^2 to some small positive value. In the present context, σ_v^2 is estimated online, so the difficulty is overcome by putting a lower limit (*e.g.*, 10^{-8}) on $\hat{\sigma}_{v,k}^2$, within the variance estimation filter.

A third problem is the need for large amounts of data to achieve low model variance and avoid over-training on one hand, and the need for short windows to address the nonstationarity of the signal on the other hand. The problem of data scarcity can be ameliorated somewhat by using the parameters learned in one window to initialize the next window; the overlap between windows makes this especially appropriate. The problem of within-window nonstationarity can be partially addressed by using model structures of higher complexity (*e.g.*, a neural network instead of an LPC model). Of course, higher complexity models also require more data, so this is a partial solution at best.

Fourth, proper normalization of the speech signal is difficult, because of the large variation in signal levels. This makes the appropriate choice of parameters such as P_0 and Q_0 problematic. While each window could be normalized individually to have zero mean and unit variance, this introduces radical changes in the dynamics from one window to the next. Thus, the speech signals are normalized in their entirety (amplitude variation notwithstanding), as was done for the stationary signals considered earlier in this chapter.

Finally, the dual estimation approach *per se* does not address the large body of knowledge about human perception of speech that has been developed in the literature of speech processing and psychology. For example, a great deal is known about the effects of masking, phase distortion, critical bands, *etc.*, on speech perception [57]. However, the flexibility of the dual estimation

approach offers the potential for incorporating much of this knowledge: *e.g.*, by use of perceptually constrained cost functions, or by independent processing of different critical bands. These possibilities remain as promising areas of future research.

Before proceeding with the experimental results, it is worth mentioning that the windowing approach to processing speech is only one of several possibilities. Although using overlapping windows is fairly straightforward, the approach introduces an inherent delay in the enhancement process, making it unsuitable for real-time applications (computational requirements notwithstanding). An alternative approach, mentioned in Section 3.6.3, involves sliding the window by only one point each time, so that an estimate of the current value of the signal is always available. However, this increases the computational expense considerably, and is not likely to improve the quality of the overall speech estimate. As a topic of future research, the windowing scheme might be avoided altogether by finding a way to track the changing dynamics of the signal, perhaps by using a parameterized model of the state-transition function for $\mathbf{w}$, instead of the identity map. This approach is discussed further in Chapter 6.

5.5.2 Evaluation of Speech

Because a human listener is often the end-user of a speech enhancement system, proper evaluation of performance is very difficult to perform. This is because objective measures, such as SNR, are poor indicators of speech quality or intelligibility, as perceived by humans. Although several "perceptual" objective quality measures have been developed (*e.g.*, Itakura-Saito distortion, weighted-spectral slope, log area ratio, log-likelihood ratio, etc. [28]), they are not adequate for making a definitive evaluation of speech enhancement algorithms. To date, the only effective means of comparison is subjective testing with human listeners (*e.g.*, calculating mean opinion scores). However, such tests are time consuming and expensive to perform, so they are not frequently used.

The algorithms developed in this thesis are designed to minimize mean squared error, or increase SNR. Hence, although they are evaluated in terms of the objective measures listed above, SNR is the primary criterion used for selecting an algorithm. Rather than compute the SNR of the entire signal at once, however, a more perceptually relevant measure is used, known as *segmental SNR*. This is computed as the average of the SNRs computed within 240-point windows, or *frames* of speech:

$$SSNR = \frac{1}{\#\text{frames}} \sum_i \max(SNR_i, -10\text{dB}). \tag{5.3}$$

Here, SNR_i is the SNR of the i^{th} 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) [28], and is expected to improve the measure's perceptual relevance.

5.5.3 Controlled Comparisons

In addition to their limited perceptual relevance, a major drawback of segmental SNR and the other objective measures is that they require the clean speech signal as a reference. Therefore, the algorithms are compared by testing them with a controlled experiment similar to those presented in the previous chapter. To reduce the processing requirements of multiple repetitions, a short section of speech is used, corresponding to a single word ("tool") spoken by a woman with a British accent. The speech is sampled at 8 kHz. Ten repetitions of white Gaussian noise are added at 3 dB SNR to produce the noisy speech. The measurement noise variance, σ_n^2, is estimated from the first N_{win} points of the signal, whereas the process noise variance σ_v^2 is estimated on-line using the maximum-likelihood cost function. Although σ_n^2 could be estimated on-line as well, this is typically unnecessary in a stationary noise environment.

Of course, more extensive testing on longer signals with various speakers should ideally be performed; as mentioned above, perceptual testing by human subjects is also required for an adequate ranking of algorithms. However, the purpose of these experiments is more limited in scope: we wish to determine how the results in early sections for more generic signals translate into the speech domain.

In some initial experiments, over-training was found to be a serious problem, with the maximum value of SNR_i occurring after around 5 epochs, and decreasing thereafter (even as the weight cost continues to improve). The effect is fairly independent of window length and model structure; hence, the number of epochs is fixed at 5. The parameter covariances are initialized at .1, except $\mathbf{Q}_0 = .01\mathbf{I}$ for $J^{ml}(\mathbf{w})$; the initial signal covariance is $\mathbf{P}_0 = \mathbf{I}$. Forgetting factors are $\lambda_{\mathbf{w}} = .9997$ $\lambda_{\sigma_v^2} = .9993$.

Apart from choosing an appropriate cost function for speech enhancement, an appropriate value for the window length, N_{win}, must be determined, as must a model structure for $f(\cdot)$. Two model structures are tested: a tenth order linear AR model; and a neural network with 10 inputs, 4 hidden units in a single layer, and one output. Furthermore, two different windowing schemes are tried:

1. Windows of length 512, shifted by 64 points.

2. Windows of length 128, shifted by 32 points.

For each of the four cases, both the dual EKF algorithm (with weight costs: $J^{pe}(\mathbf{w})$, $J^{ml}(\mathbf{w})$, and $J^{j}(\mathbf{w})$), and the joint EKF algorithm are tested. In addition, the batch EM algorithm is tested with the linear architecture (the nonlinear batch GEM algorithm is not effective). Finally, the traditional method of spectral subtraction is tested, using code developed by Levent Arslan at Duke University[1]. This is intended for benchmarking purposes only; more sophisticated forms of spectral subtraction have been developed, and would most likely produce more competitive results. Measures for the original noisy speech are also computed to indicate relative improvement of the enhancement algorithms.

For each algorithm or cost function, the perceptual measures are averaged across all frames and compiled in the boxplots of Figure 5.11. The top plot shows the segmental SNR, which is the only measure for which larger numbers indicate better performance. For most quality measures, the result using $N_{win} = 512$ compares favorably with the corresponding result using the shorter window, with the exception of the Itakura-Saito measure. The advantage of the longer window is related to the amount of noisy data required to estimate the parameters. It is possible that shorter windows would be sufficient for processing speech that is less noisy.

The Itakura-Saito and weighted spectral slope measures are immediately suspect, because the distance for unprocessed speech (labeled "y"), is in many cases lower than that of the processed speech. The degradation in the Itakura-Saito measure is localized to several silent (non-speech) frames, but these values greatly inflate the average distance, nonetheless. These two measures are disregarded in the rest of the discussion.

On the longer window, the neural network model shows a higher segmental SNR than the linear model; this advantage is less pronounced on shorter windows, which is probably due to the larger number of parameters used in the neural network. The outcome is less conclusive for the other measures.

In terms of the segmental SNR, the best results are obtained by the dual EKF with $J^{ml}(\mathbf{w})$ cost, using a neural network and longer window. The advantage of the maximum-likelihood cost over $J^{j}(\mathbf{w})$ is expected in the white noise case, based on experiments in the previous chapter. The weaker performance of the joint EKF is very likely a result of the model structure errors inherent to the speech enhancement problem, and inaccuracies in the noise variance estimates. Another factor might be lack of data reweighting for weight training (see page 110) in the joint EKF.

As expected, the spectral subtraction shows very poor segmental SNR performance; however, the algorithm is an average performer in terms of the log area ratio and log likelihood measures.

[1]The default settings of $N_{win} = 128$ Hanning windows, shifted by 64 points are used.

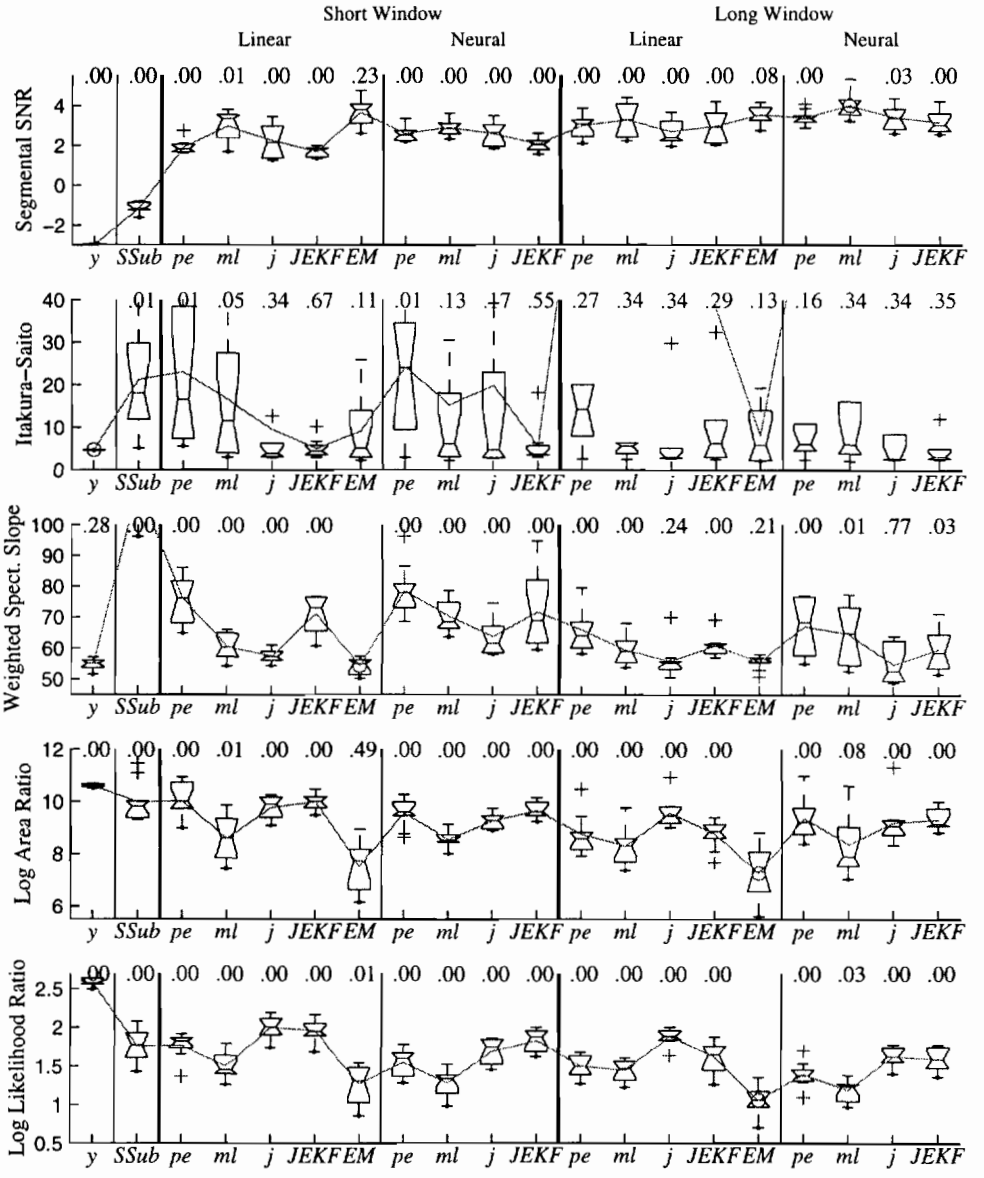


Figure 5.11: Boxplots of various perceptual metrics, obtained using several different speech enhancement algorithms on a single word in 10 repetitions of 3 dB white noise. The left three panels show the results using a 128 point window, while the results using a 512 point window are shown in the right two panels. Results are further divided into linear and neural network models, as indicated. Values for the noisy speech and spectral subtraction result are shown on the left.

It is interesting to note the excellent performance of the linear batch EM algorithm, especially as rated by the log area ratio and log likelihood measures. Recall that the EM algorithm uses a Kalman smoother to improve the signal estimates; it shows significantly higher segmental SNR than the other linear algorithms. For the shorter window the result has a p-value of 23% relative

to the top-performing dual EKF algorithm.

This highlights the potential advantage of using noncausal processing in off-line applications; one possible approach is a dual forward-backward Kalman filter investigated in [87], but this has not been fully developed. A simpler form of noncausal estimation could be performed by taking the signal estimates $\hat{x}_{k+1}$ from the last element of the state vector $\hat{\mathbf{x}}_{k+M}$, thereby using a limited amount of future data to estimate the signal. This type of processing is sometimes referred to as "fixed-lag" smoothing.

5.5.4 Digit Recognition

Apart from increasing the perceptual quality and intelligibility of speech for human listeners, an important application of speech enhancement technology is as a front-end to automatic speech recognition (ASR) systems. Often ASR systems are trained to recognize relatively clean speech, but must deal with noisy environments when put into use. Such noise might originate from a factory setting, an automobile, or even computer fan noise.

One way of increasing the robustness of an ASR system to noisy speech is by preprocessing the speech with an enhancement algorithm. The effectiveness of the dual EKF in this application is demonstrated using speech corpus and ASR system[2] 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 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 5.2, the recognition rates on the clean telephone speech are quite good. However, adding white Gaussian noise to the speech at 6dB significantly reduces the performance. As a benchmark, the standard spectral subtraction routine described in the previous section was used to enhance the noisy speech, resulting in a significant improvement in recognition. In addition, an enhancement algorithm built into the speech codec TIA/EIA/IS-718 for digital cellular phones (published by the Telecommunications Industry Association) was used, with the compression features of the algorithm disabled. Although the perceptual quality of the IS-718 enhancement is considerably better than the spectral subtraction result, the recognition rates are significantly worse.

The dual EKF algorithm is applied with maximum-likelihood costs for estimating the weights and process noise variance, and with static derivatives to reduce the computational expense. The

[2]The author wishes to thank Edward Kaiser for his invaluable assistance in this experiment

Table 5.2: 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)

measurement noise variance is estimated from the first window (512 points) of the noisy signal. The neural network architecture and other parameters are chosen as in the previous experiment. As shown by Table 5.2, the dual EKF outperforms both the IS-718 and spectral subtraction recognition rates by a significant amount. The improvement in terms of correctly recognized sentences is even more dramatic.

5.5.5 SpEAR Data

As mentioned earlier, computing perceptual quality measures of enhanced speech requires access to clean speech waveforms. Often, then, enhancement is performed on artificially corrupted speech, wherein a noise waveform is added in digital form to the clean speech waveform. While this provides access to the clean speech, the results are somewhat questionable, because the noise was not part of the same acoustic environment as the speech.

To increase the level of realism of the noisy speech, and yet still provide access to the clean speech waveform, a database of *acoustically corrupted* speech is under development as part of CSLU's Speech Enhancement Assessment Resource (SpEAR [13]). Noisy speech files in this database were created by simultaneously playing both noise and speech waveforms in the same room, and recording the acoustic combination clock-synchronously to produce a noisy speech waveform. A reference to the clean speech is also created by playing the speech waveform in the room (without noise) and re-recording it. This allows for segmental SNR to be computed for both the noisy speech and the enhanced speech.

A portion of the SpEAR database was processed by the dual EKF in order to evaluate the algorithm on a broader array of noise types. A variety of noise sources are acoustically combined with two different sentences, spoken by an American male and an American female, respectively. The clean speech files originate from the TIMIT database. Noise sources from the SPIB database [69] are:

- Noise recorded from the co-pilot's seat in a two-seat F-16, traveling at a speed of 500 knots, and an altitude between 300 and 600 feet. The sound level during the recording process was 103 dBA.
- Factory noise recorded in an automobile production hall.
- Noise recorded inside a Volvo 340 in 4th gear on an asphalt road, at 120 km/h in rainy conditions.

In addition, pink noise, stationary white noise, and nonstationary (bursting) white noise are used. Note that the spectra of all noise sources are altered by the acoustics of the SpEAR recording environment, which was a carpeted room with painted plaster-board walls. For these experiments, the 16kHz SpEAR data was downsampled to 8kHz before processing.

In most cases, the noise parameters $\mathbf{w}_n$ and $\sigma_{v_n}^2$ ($L_n = 10$) are estimated from a 512 point window of noise at the beginning of each recording. For the Volvo noise the model ($L_n = 12$) is estimated using the entire noise file, available by subtracting the clean reference. The model of the bursting white noise ($L_n = 10$) is estimated using a long segment of stationary white noise, and the value of $\sigma_{v_n}^2$ is estimated online using $J^{ml}(\sigma_{v_n}^2)$. This requires that the algorithm track the noise level from one window to the next; because there are portions of the waveform with no measurement noise, the value of $\hat{\sigma}_{v_n}^2$ was thresholded at a minimum value of 10^{-4}.

In all cases, the dual EKF is used with $J^j(\mathbf{w})$ and $J^{ml}(\sigma_v^2)$ costs, the usual choices of initial covariances, and with $\lambda_{\mathbf{w}} = .9997$ $\lambda_{\sigma_v^2} = .9993$, $\lambda_{\sigma_n^2} = .9993$. Table 5.3 presents the results in terms of average segmental SNR. The segmental SNR is shown for the noisy speech, and for the enhanced speech using both the standard and static derivative forms of the algorithm. In most cases, the full recursive derivative produces somewhat better results; however, the static derivative results are often quite close or better. In particular, the results on the low frequency Volvo noise

Table 5.3: Dual EKF enhancement results using a portion of the SpEAR database. 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.

	Male Voice (Seg. SNR)			Female Voice (Seg. SNR)		
	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

favor the static derivative results, for reasons that are unclear.

The application of the dual EKF to some "real-world" noisy speech signals is considered next.

5.5.6 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 speech is enhanced by the dual EKF with costs: $J^j(\mathbf{w})$, $J^{ml}(\sigma^2)$, and parameters: $P_0 = 1$, $Q_0 = .01$, $q_{v,0} = .1$. The measurement noise is modeled with an AR-12 model using a separate portion of the signal which does not contain speech. The process noise variance σ_v^2 is estimated on-line within the dual EKF framework. Following the results in Section 5.5.3, the longer window length (512 points) is used, with a feedforward neural network architecture of 10-4-1.

The results appear in Figure 5.12, along with the noisy signal. Spectrograms of both the noisy speech and estimated speech are included to aid in the comparison. 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[3].

The noise reduction is most successful in non-speech 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. The spectrogram suggests that better results might be obtained by processing different frequency bands individually and combining the results. This would potentially suppress the residual noise at high frequencies, and wherever no speech signal is present in a particular band. However, this issue is left as a topic of future research.

5.5.7 Richard Nixon

On November 17, 1973, during the height of the Watergate scandal, President Richard Nixon spoke to the American people in an attempt to reassure them of his innocence. A portion of the speech, in which Nixon states, "... because people have gotta know whether or not their President's a crook; well, I'm not a crook," is represented by the waveform and spectrogram at the top of Figure 5.13. In this example, the additive noise appears to be subject to quantization effects, and clearly violates the assumption of Gaussianity. The noise level is much lower than in the car

[3]Thanks to J. A. du Preez at the University of Stellenbosch for MATLAB code used to compute these spectrograms.

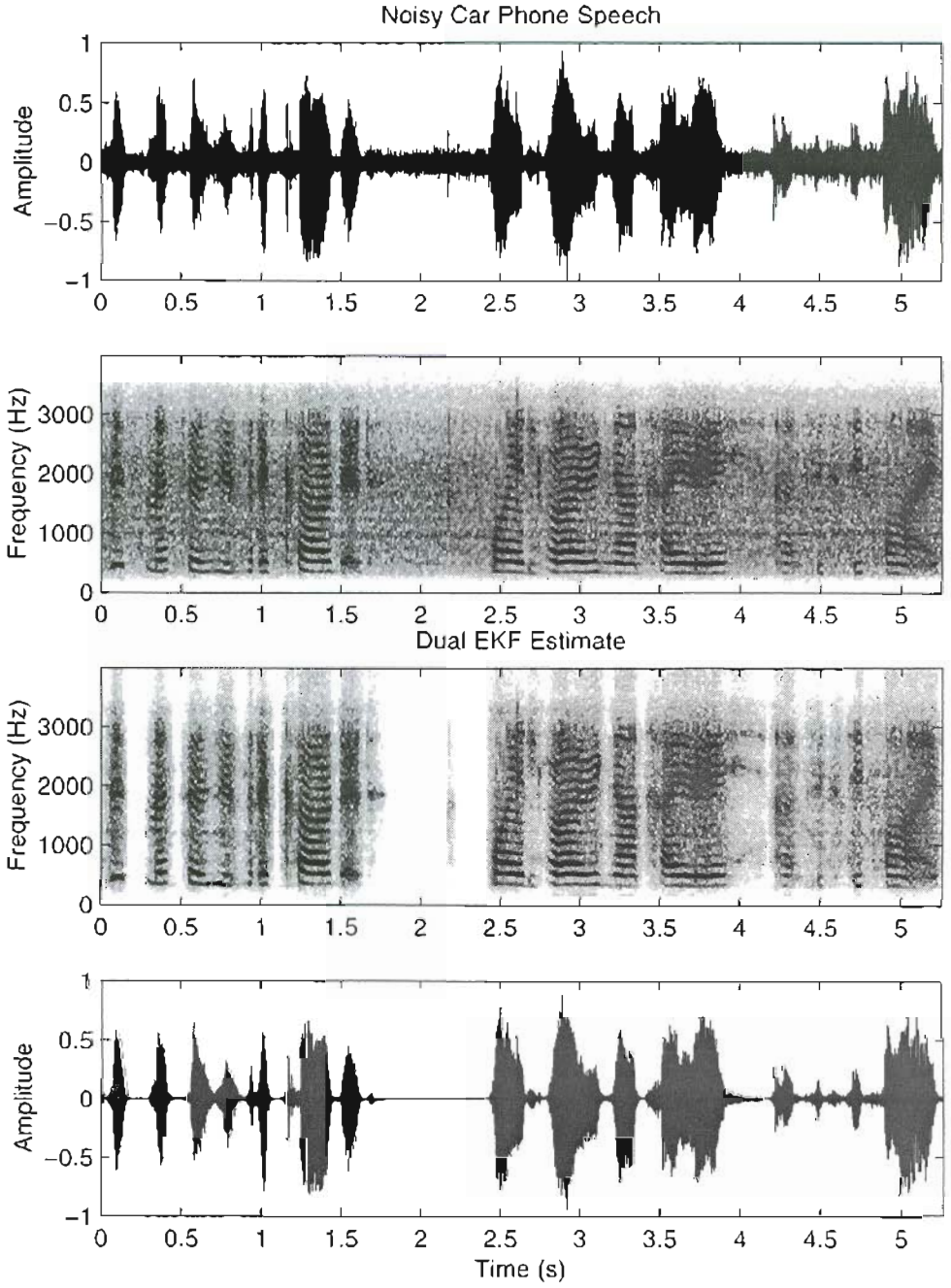


Figure 5.12: Enhancement of car phone speech. The noisy waveform appears in the top plot, followed by its spectrogram. The third and fourth plots contain the spectrogram and waveform, respectively, of the dual EKF result.

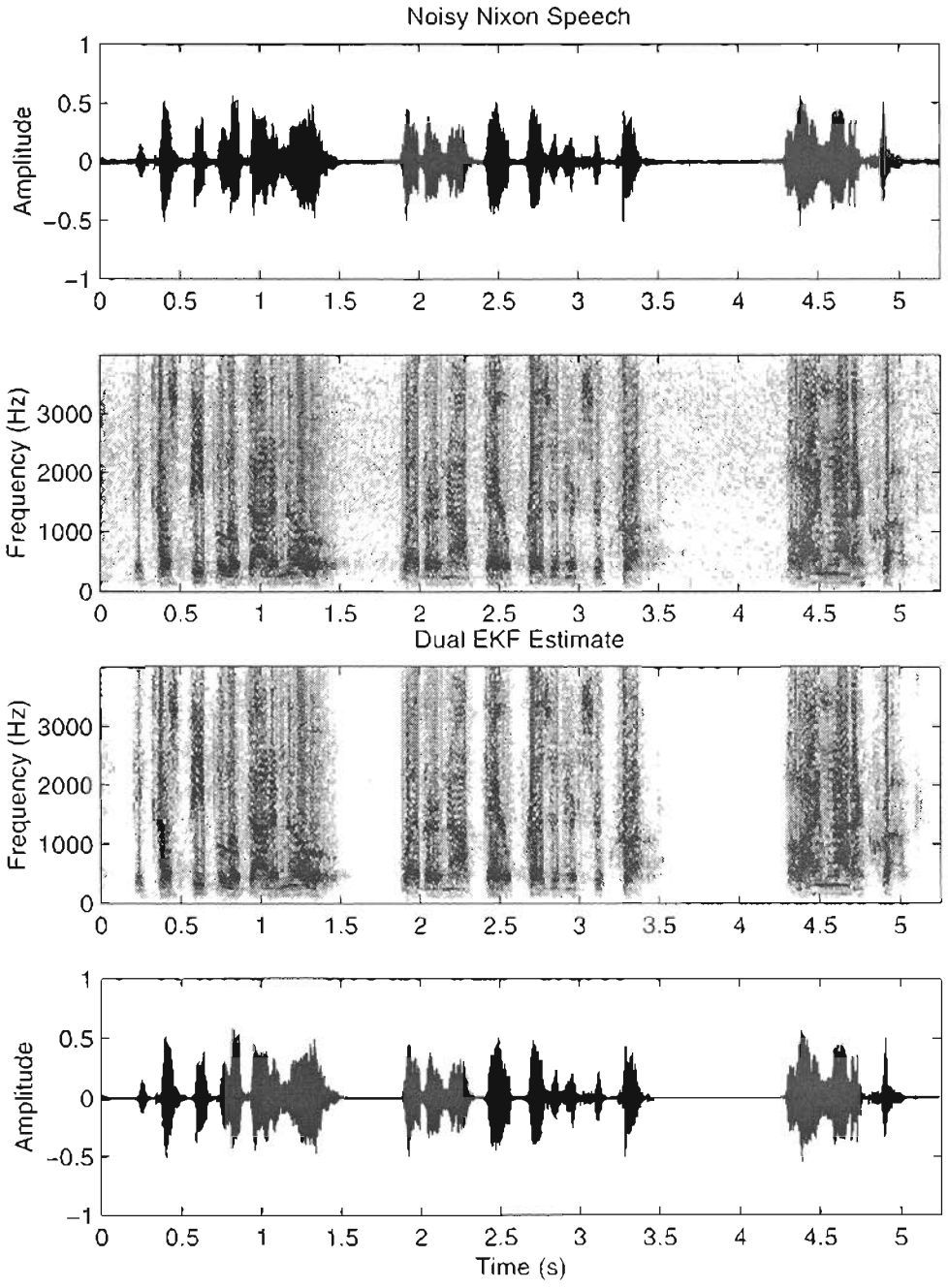


Figure 5.13: Enhancement of Richard Nixon's "I'm not a crook" speech. The noisy waveform appears in the top plot, followed by its spectrogram. The third and fourth plots contain the spectrogram and waveform, respectively, of the dual EKF result.

phone example, but the reduction of noise is apparent in both the spectrogram and waveform of the enhanced speech, shown in the bottom half of Figure 5.13.

5.5.8 Seminar Recording

A last example comes from a recording made during a lecture in the Portland Area Semiconductor Seminar Series at the Oregon Graduate Institute. The seminars are routinely videotaped and stored in an archive. However, during one particular lecture, 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. As suggested by Figure 5.14, the SNR is extremely low, making for an unusually difficult audio enhancement problem.

5.6 Discussion of Results

While the speech enhancement results in the previous section are very promising for both ASR and human-listener applications, much additional work remains to improve the application of the dual EKF to speech processing. A voice activity detector could be used to re-estimate the noise model from nonspeech segments of the waveform, thereby improving performance in the presence of slowly varying measurement noise correlations. Perhaps the most promising area of future research involves the use of perceptually motivated cost functions for the signal and weight estimation filters. Additional gains can possibly be made by simply band-pass filtering the speech into critical bands, and estimating the waveform in each band separately before recombining.

In any case, the results shown on economic, geophysical, and speech data demonstrate the potential of the dual EKF approach, and its applicability to a wide variety of real-world signal processing problems. The next chapter summarizes the general conclusions that can be drawn from this research, and describes directions for further refinement of the dual Kalman filtering approach.

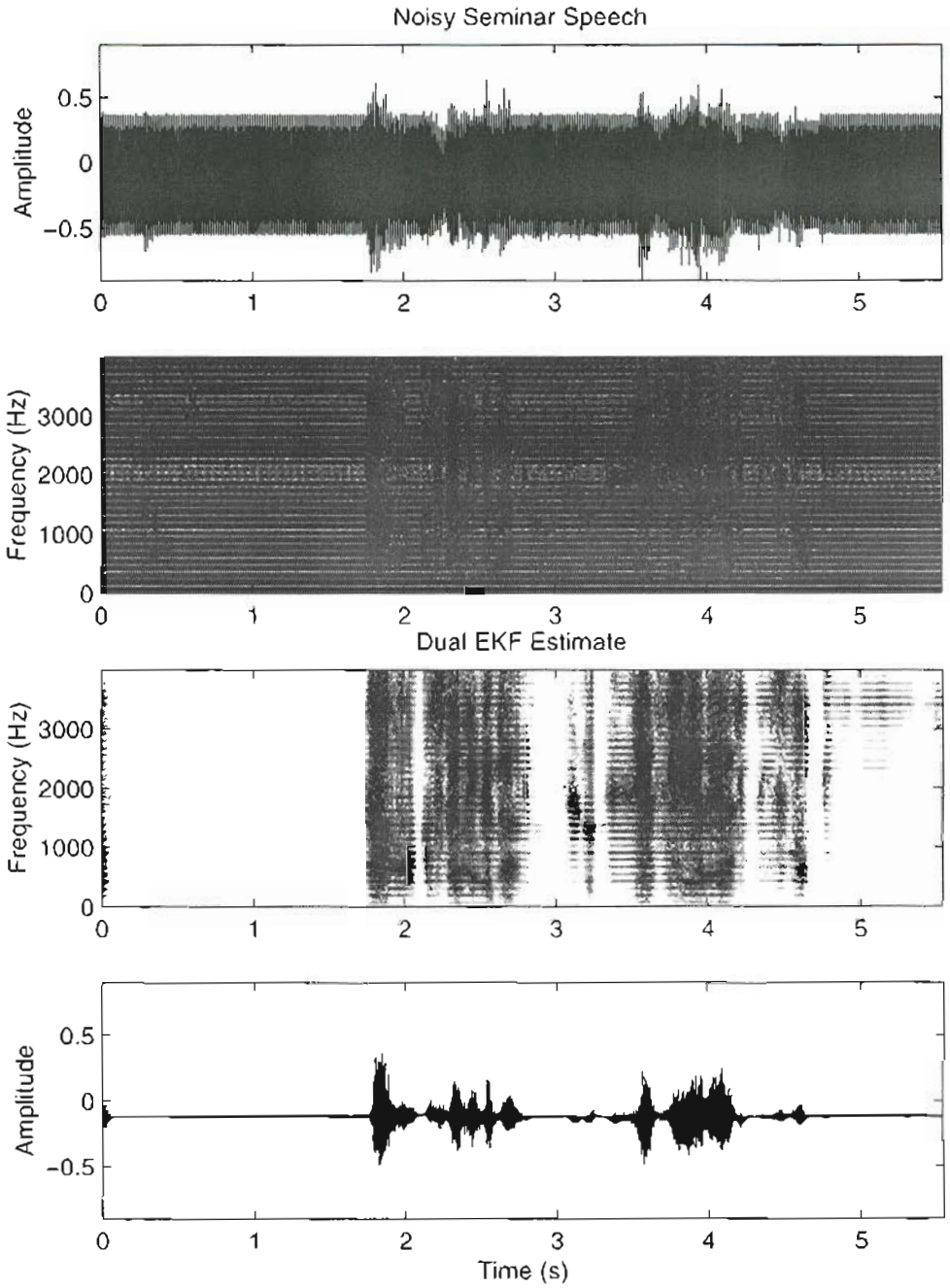


Figure 5.14: Enhancement of high-noise seminar recording. The noisy waveform appears in the top plot, followed by its spectrogram. The third and fourth plots contain the spectrogram and waveform, respectively, of the dual EKF result

Chapter 6

Conclusions and Future Work

6.1 General Summary

This thesis approaches dual estimation from 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 time-series. 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.

This approach offers some 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 [61, 47] is quite different from the joint EKF approach [38, 12], which minimizes a joint estimation cost[1]. 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 relative utility of the various cost functions is evaluated through the development of the dual extended Kalman filter. The dual EKF provides a common algorithmic platform for implementing a broad variety of methods, and allows for the direct comparison of the different cost functions used in the literature. Extensive empirical comparisons are performed, with the outcomes provided in Chapter 4.

The dual EKF is an effective sequential dual estimation method, which is applicable to both linear and nonlinear time-series models, and which can be used in the presence of white or colored measurement noise. The algorithm is comprehensive in that it provides sequential estimation of

[1] This fact is overlooked in [47], which emphasizes the similarity of these two algorithms.

noise variance parameters within the same theoretical framework used to estimate the model and signal.

Furthermore, the generality of the dual EKF is demonstrated in a range of application domains, including speech enhancement, economic forecasting, and analysis of geophysical data. These results illustrate the potential of the dual EKF for processing many different types of signals. In addition, the flexibility of the approach allows for the future development and use of application-specific cost functions and pre-processing schemes; these and other possible directions for future research are described in the next section.

6.2 Possible Extensions

The dual EKF allows a relatively small amount of prior information – in the form of the model structure of the dynamics and a model of the noise – to be used in solving the dual estimation problem. Many of the research directions suggested below would allow for other types of prior information to be included. Other ideas involve an attempt to reduce the amount of required prior information even further: either by learning the noise model (as in monaural signal separation), or by adapting the complexity of the model structure. Alternatives to the EKF, and the importance of developing specific applications are also discussed.

Model Improvements. As mentioned in Chapter 1, the simple nonlinear autoregressive model structure used in this thesis can be generalized to include exogenous control inputs to the function $f(\cdot)$, and to allow for multiple observations ($\dim(y_k) > 1$). These adjustments would be fairly straightforward in the state-space framework, and would greatly increase the range of applications; the algorithm could be used for system identification in control settings, and for predicting economic time-series using information contained in other series.

More general forms of distortion can also be considered by allowing for the observation to be a nonlinear function of x_k and n_k, rather than a linear addition. Convolutional noise and other forms of channel distortion can be addressed by such a model.

Nonlinear Noise Models. This thesis assumes a *linear* AR model for colored measurement noise. In some cases, the dynamics of the noise would be better modeled with a nonlinear autoregression, like the model used for the dynamics of signal. This modification would be very straightforward, and would be useful when the noise and signal are of similar complexity.

Monaural Signal Separation. An additional step towards equal treatment of the signal and noise is to regard the model parameters of the noise as unknown. Any distinction between

signal and noise is thereby removed, and the problem is reframed as that of separating two signals from a single (monaural) source. The problem, referred to as *monaural blind signal separation*, is a very challenging area of research. However, some preliminary work in [88] demonstrates the potential of the dual EKF in this setting.

Model Structure Selection and Regularization Besides the noise model, another key piece of information used by the dual EKF is the complexity of the signal dynamics, $f(\cdot)$. This thesis assumes a predetermined model structure for each signal, with a specific parameterization. However, in some contexts a suitable model structure will not be known *a priori*. In this case, an adjustment to the cost function to introduce *regularization* [30] could provide some control over the model complexity. Other approaches to model selection, such as *pruning* [30], can be used to select to appropriate number of parameters in the model.

Filtering in Other Transform Domains. The definition of the state vector $\mathbf{x}_k$ in Equation 3.11 as the lagged values of the signal, x_k, is only one of an infinite number of possibilities. Other representations of the signal, including polynomial and wavelet transforms, can also be considered. These alternative state-space definitions might be chosen to allow prior information about the signal to be included in the model, or to facilitate the use of application-specific cost functions and constraints.

Nonstationary Signal Modeling. Windowing nonstationary signals into short overlapping segments, as is done in this thesis for speech data, introduces the additional difficulties of data scarcity and over-training. A more desirable approach to filtering signals with time-varying dynamics might be developed by using domain-switching models, or by estimating a continuous model of the dynamics exhibited by $\mathbf{w}_k$. In other words, the changes in the dynamics of the signal are themselves modeled by a fixed function, whose parameters must be estimated along with x_k and $\mathbf{w}_k$. This approach assumes that the dynamics of $\mathbf{w}_k$ remain in a bounded region of the parameter space, but would offer the advantage of making all the noisy data in the past available for estimation of the signal.

Alternatives to the EKF. Appendix D provides an analysis of the approximations made by the extended Kalman filter, and the inaccuracies that result when the model is highly nonlinear. Alternative filters have been derived which offer the potential of better accuracy than the EKF, and which could be substituted for the signal, weight, or variance filters of the dual EKF algorithm. The use of unscented Kalman filters (UKF) [35] in this manner was investigated recently using the $J^{pe}(\mathbf{w})$ cost, with promising results [90]. Whereas the UKF still

adheres to a Gaussian assumption on the state, a sequential Monte Carlo approach known as particle filtering [15] avoids the Gaussian assumption altogether, although at increased computational expense. These alternative filtering methods could be used within a sequential dual estimation approach similar to the dual EKF, but with improved convergence properties for highly nonlinear signals.

Speech Enhancement. Some suggestions for improving the speech enhancement results of the dual EKF are made at the end of the previous chapter. In addition, several of the ideas mentioned above have particular relevance to the speech domain. First, the extension of the model to handle channel distortion could improve results in the telecommunication domain, where noise is not purely additive. Second, because background noise often includes other speaking voices, the problem of monaural blind signal separation is important for developing robust speech applications. Third, the development of perceptually-motivated cost functions using our knowledge of the human auditory system could be facilitated by estimating the speech in alternative transform domains. Finally, the ability to track dynamic regimes within the speech waveform would obviate the need for windowing speech, thereby improving performance and computation time.

Application Development. The results shown in the previous chapter are somewhat preliminary in the sense that the dual EKF was applied to these data sets with little or no application-specific modification. Numerous issues arise in the context of a particular application; the adaptation or alteration of the algorithm to accommodate these issues would certainly produce results superior to those shown here, and represents an important direction for future research.

The power of the dual EKF approach comes from its theoretical foundation, and its ability to be used with many different cost functions and application domains. This flexibility makes the dual EKF an excellent starting point for a number of possible research directions. Although certainly not exhaustive, the above list contains ideas regarded by the author as the most promising in terms of their potential benefit to the research community. Some of these proposals are quite straight-forward; others represent a considerable amount of work. All of them would increase the impact of the dual EKF paradigm on a variety of fields.

Bibliography

[1] Hirotugu Akaike. Maximum likelihood identification of Gaussian autoregressive moving average models. *Biometrika*, 60(2):255–65, 1973.

[2] Brian D. O. Anderson and John B. Moore. *Optimal Filtering.* Prentice-Hall, 1979.

[3] A. Blake, B. North, and M. Isaard. Learning multi-class dynamics. In *Advances in Neural Information Processing Systems 11*, pages 389–95. MIT Press, 1999.

[4] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics Speech and Signal Processing*, ASSP-27:113–20, 1979.

[5] Thomas Briegel and Volker 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*, pages 403–9. MIT Press, 1999.

[6] William L. Brogan. *Modern Control Theory.* Prentice Hall, 1991.

[7] A. E. Bryson, Jr. and L. J. Henrikson. Estimation using sampled data containing sequentially correlated noise. *American Institute of Aeronautics and Astronautics (AIAA) Journal of Spacecraft and Rockets*, 5(6):662–5, 1968.

[8] Chi-Tsong Chen. *Linear System Theory and Design.* Saunders, Harcourt Brace College Publishers, 1984.

[9] Carnegie Mellon University, Department of Statistics: *StatLib – Datasets Archive.* Available on the Internet at http://lib.stat.cmu.edu/datasets/, Last Updated: March 10, 2000. Accessed: September 1, 2000.

[10] Jerome T. Connor, R. Douglas Martin, and Les E. Atlas. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2):240–54, 1994.

[11] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[12] Henry Cox. On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on Automatic Control*, AC-9:5–12, 1964.

[13] Center for Spoken Language Understanding: *Speech Enhancement Assessment Resource* (SpEAR). Available on the Internet at http://cslu.ece.ogi.edu/nsel/data/index.html, Last Updated: June 1, 2000. Accessed: September 1, 2000.

[14] J. F. G. de Freitas, M. Niranjan, and A. H. Gee. The EM algorithm and neural networks for nonlinear state space estimation. Technical Report TR-313, Cambridge University Engineering Dept., 1998.

[15] 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, 1998.

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

[17] Pieter Eykhoff. *System Identification: Parameter and State Estimation.* John Wiley & Sons, 1974.

[18] Lee A. Feldkamp. Informal email correspondance with Eric A. Wan, 2000.

[19] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamic Systems.* Addison-Wesley, 1990.

[20] FRED: *Federal Reserve Economic Data.* Available on the Internet at http://www.stls.frb.org/fred/, Last Updated: July, 2000. Accessed: September 1, 2000.

[21] Sharon Gannot, David Burshtein, and Ehud Weinstein. Iterative-batch and sequential algorithms for single microphone speech enhancement. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pages 1215–8. IEEE, 1998.

[22] Zoubin Ghahramani. Learning dynamic Bayesian networks. In C.L. Giles and M. Gori, editors, *Adaptive Processing of Sequences and Data Structures*, pages 168–97. Springer-Verlag, 1998.

[23] Zoubin Ghahramani and Sam T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems 11*, pages 431–7. MIT Press, 1999.

[24] Jerry D. Gibson, Boneung Koo, and Steven D. Gray. Filtering of colored noise for speech enhancement and coding. *IEEE Transactions on Signal Processing*, 39(8):1732–41, 1991.

[25] Gene H. Golub and Charles F. van Loan. An analysis of the total least squares problem. *SIAM Journal of Numerical Analysis*, 17(6):883–93, 1980.

[26] Narendra K. Gupta and Raman K. Mehra. Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Transactions on Automatic Control*, AC-19(6):774–83, 1974.

[27] S. Hammel, C. K. R. T. Jones, and J. V. Moloney. Global dynamical systems and bifurcations of vector fields. *Journal of the Optical Society of America*, B 2(552), 1985.

[28] John H. L. Hansen and Bryan L. Pellom. An effective quality evaluation protocol for speech enhancement algorithms. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP-98*, pages 2819–22. Australian Speech Science and Technology Association, 1998.

[29] Simon Haykin. *Adaptive Filter Theory.* Prentice-Hall, 3^{rd} edition, 1996.

[30] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice-Hall, 2^{nd} edition, 1999.

[31] Simon Haykin and Jose Principe. Making sense of a complex world. *Signal Processing Magazine*, 15(3):66–80, 1998.

[32] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[33] John H. L. Hansen John R. Deller, John G. Praokis. *Discrete-Time Processing of Speech Signals.* Macmillan Publishing Company, 1993.

[34] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

[35] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, pages 182–93. SPIE, 1997.

[36] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83D:95–108, 1960.

[37] Boneung Koo, Jerry D. Gibson, and Steven D. Gray. Filtering of colored noise for speech enhancement and coding. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, pages 349–52. IEEE, 1989.

[38] Richard E. Kopp and Richard J. Orford. Linear regression applied to system identification for adaptive control systems. *American Institute of Aeronautics and Astronautics (AIAA) Journal*, 1:2300–06, 1963.

[39] Vikram Krishnamurthy, Leigh Johnston, and Andrew Logothetis. Optimal MAP estimation of bilinear systems via the EM algorithm. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pages 2373–6. IEEE, 1998.

[40] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks: prediction and system modelling. Technical Report LA-UR–87-2662, Los Alamos National Laboratory, 1987.

[41] Byung-Gook Lee, Ki Yong Lee, and Souguil Ann. An EM-based approach for parameter enhancement with an application to speech signals. *Signal Processing*, 46:1–14, 1995.

[42] Ki Yong Lee, Byung-Gook Lee, Iickho Song, and Jisang Yoo. Recursive speech enhancement using the EM algorithm with initial conditions trained by HMM's. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pages 621–4. IEEE, 1996.

[43] Frank L. Lewis. *Optimal Estimation.* John Wiley & Sons, 1986.

[44] Jae S. Lim and Alan V. Oppenheim. All-pole modeling of degraded speech. *IEEE Transactions on Acoustics Speech and Signal Processing*, 26(3):197–210, 1978.

[45] Lennart Ljung. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear sytems. *IEEE Transactions on Automatic Control*, AC-24(1):36–50, 1979.

[46] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.

[47] Lennart Ljung and Torsten Söderström. *Theory and Practice of Recursive Identification*. MIT Press, 1983.

[48] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 2^{nd} edition, 1984.

[49] Michael C. Mackey and Leon Glass. Oscillations and chaos in physiological control systems. *Science*, 197(4300):287–9, 1977.

[50] R. Douglas Martin. Robust methods for time series. Technical Report 20, Department of Statistics, University of Washington, 1982.

[51] Michael B. Matthews and G. S. Moschytz. Neural-network nonlinear adaptive filtering using the extended Kalman filter algorithm. In *International Neural Network Conference, INNC-90*, volume 1, pages 115–9. Kluwer Academic, 1990.

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

[53] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 2. Academic Press, 1982.

[54] Raman K. Mehra. Identification of stochastic linear dynamic systems using Kalman filter representation. *American Institute of Aeronautics and Astronatics (AIAA) Journal*, 9:28–31, 1971.

[55] John Moody. Economic forecasting: challenges and neural network solutions. In *International Symposium on Artificial Neural Networks*, 1995. Keynote address.

[56] John Moody, Uzi Levin, and Steve Rehfuss. Predicting the U.S. index of industrial production. *Neural Network World*, 3(6):791–4, 1993.

[57] Brian C. J. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, 4^{th} edition, 1997.

[58] Bruce R. Musicus and Jae S. Lim. Maximum likelihood parameter estimation of noisy data. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pages 224–7. IEEE, 1979.

[59] Alex T. Nelson and Eric A. Wan. Neural speech enhancement using dual extended Kalman filtering. In *Proceedings of the International Conference on Neural Networks, ICNN'97*, volume 4, pages 2171–5. IEEE, 1997.

[60] Alex T. Nelson and Eric A. Wan. A two-observation Kalman framework for maximum-likelihood modeling of noisy time series. In *Proceedings of International Joint Conference on Neural Networks, IJCNN'98*. IEEE, 1998. [CDROM].

[61] Lawrence W. Nelson and Edwin Stear. The simultaneous on-line estimation of parameters and states in linear systems. *IEEE Transactions on Automatic Control*, AC-21(2):94–8, 1976.

[62] Maciej Niedźwiecki and Krzysztof Cisowski. Adaptive scheme of elimination of broadband noise and impulsive disturbances from AR and ARMA signals. *IEEE Transactions on Signal Processing*, 44(3):528–37, 1996.

[63] K. K. Paliwal and A. Basu. A speech enhancement method based on Kalman filtering. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pages 177–80. IEEE, 1987.

[64] Donald B. Percival and Andrew T. Walden. *Spectral Analysis for Physical Applications: Multitaper and Conventional Techniques*. Cambridge University Press, 1993.

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

[66] Gintaras V. Puskorius and Lee A. Feldkamp. Neural control of nonlinear dynamic systems with Kalman filter trained recurrent networks. *IEEE Transactions on Neural Networks*, 5(2):279–97, 1994.

[67] Gintaras V. Puskorius and Lee A. Feldkamp. Extensions and enhancements of decoupled extended Kalman filter training. In *Proceedings of the International Conference on Neural Networks, ICNN'97*, volume 3, pages 1879–83. IEEE, 1997.

[68] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *American Institute of Aeronautics and Astronautics (AIAA) Journal*, 3(8):1445–50, 1965.

[69] Rice University: *Signal Processing Information Base* (SPIB). Available on the Internet at http://spib.ece.rice.edu/signal.html, Last Updated: September 19, 1995. Accessed: September 15, 1999.

[70] John A. Rice. *Mathematical Statistics and Data Analysis*. Wadsworth and Brooks/Cole, 1988.

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

[72] D. E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In Rumelhart, McClelland, et al., editors, *Parallel Distributed Processing*, volume 1, chapter 8, pages 318–62. MIT Press, 1986.

[73] Tim Sauer, James A. Yorke, and Martin Casdagli. Embedology. *Journal of Statistical Physics*, 65(3/4):579–616, 1991.

[74] Nicol N. Schraudolph. Online local gain adaptation for multi-layer perceptrons. Technical Report IDSIA-09-98, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), 1998.

[75] G. Seber and C. Wild. *Nonlinear Regression*, chapter 10: Errors-in-Variables Models, pages 491–527. John Wiley & Sons, 1989.

[76] 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(4):253–64, 1982.

[77] Sharad Singhal and Lance Wu. Training multilayer perceptrons with the extended Kalman filter. In *Advances in Neural Information Processing Systems 1*, pages 133–40. Morgan Kauffman, 1989.

[78] Harold W. Sorenson, editor. *Kalman Filtering: Theory and Application.* IEEE Press, 1985.

[79] Robert F. Stengel. *Optimal Control and Estimation.* Dover Publications, 1994.

[80] Stephen C. Stubberud and Mark Owen. Artificial neural network feedback loop with on-line training. In *International Symposium on Intelligent Control*, pages 514–9. IEEE, 1996.

[81] John Sum, Lai wan Chan, Chi sing Leung, and Gilber H. Young. Extended Kalman filter-based pruning method for recurrent neural networks. *Neural Computation*, 10(6):1481–1505, 1998.

[82] Floris Takens. Detecting strange attractors in turbulence. In D. Rand and L.S. Young, editors, *Dynamical systems and turbulence*, pages 366–81. Springer-Verlag, 1981. Lecture Notes in Mathematics, volume 898.

[83] Volker Tresp and Reimar Hofmann. Missing and noisy data in nonlinear time-series prediction. In B. Wilson et al., editors, *Neural Networks for Signal Processing V.* IEEE Signal Processing Society, 1995.

[84] Eric A. Wan. *Finite Impulse Response Neural Networks with Applications in Time Series Prediction.* PhD thesis, Stanford University, 1993.

[85] Eric A. Wan. Modeling nonlinear dynamics with neural networks: examples in time series prediction. In *Proceedings of the Fifth Workshop on Neural Networks: Academic/Industrial/NASA/Defense, WNN93/FNN93*, pages 327–32. Simulation Councils, 1993.

[86] Eric A. Wan. Combining fossil and sunspot data: committee predictions. In *Proceedings of the International Conference on Neural Networks, ICNN'97*, volume 4, pages 2176–80. IEEE, 1997.

[87] Eric A. Wan and Alex T. Nelson. Dual Kalman filtering methods for nonlinear prediction, estimation, and smoothing. In *Advances in Neural Information Processing Systems 9*, pages 793–99. MIT Press, 1997.

[88] Eric A. Wan and Alex T. Nelson. Neural dual extended Kalman filtering: Applications in speech enhancement and monaural blind signal separation. In *Neural Networks for Signal Processing VII*, pages 466–75. IEEE, 1997.

[89] Eric A. Wan and Alex T. Nelson. Removal of noise from speech using the dual EKF algorithm. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, pages 381–4. IEEE, 1998.

[90] Eric A. Wan, Rudolph van der Merwe, and Alex T. Nelson. Dual estimation and the unscented transformation. In *Advances in Neural Information Processing Systems 12*, pages 666–72. MIT Press, 2000.

[91] Andreas S. Weigend, Benardo A. Huberman, and David E. Rumelhart. Predicting the future: a connectionist approach. *International Journal of Neural Systems*, 1:193–209, 1990.

[92] Andreas S. Weigend, Hans Georg Zimmermann, and Ralph Neuneier. Clearning. Technical Report CU-CS-772-95, University of Colorado Dept. of Computer Science, 1995.

[93] Ehud Weinstein, Alan V. Oppenheim, Meir Feder, and John R. Buck. Iterative and sequential algorithms for multisensor signal enhancement. *IEEE Transactions on Signal Processing*, 42(4):846–59, 1994.

[94] Paul J. Werbos. *Handbook of Intelligent Control*, chapter 10, pages 283–356. Van Nostrand Reinhold, 1992.

[95] R. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–80, 1989.

[96] Ronald J. Williams. Training recurrent networks using the extended Kalman filter. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN'92*, volume 4, pages 241–6. IEEE, 1992.

[97] G. U. Yule. On a method of investigating periodicity in disturbed series with special reference to Wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London, A*, 226:267–98, 1927.

Appendix A

Gaussian Conditional Densities

In this appendix, the functional form for several of the conditional densities used in this thesis are derived under a Gaussian assumption on the process and measurement noises. Namely, in Section 2.3, expressions for the terms on the right hand side of Equation 2.8:

$$\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \mathbf{w}} = \rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{\mathbf{x}_1^N | \mathbf{w}} \tag{2.8}$$

were displayed without derivation. In Section 2.4, an expression for the marginal likelihood $\rho_{\mathbf{y}_1^N | \mathbf{w}}$ was stated. Derivations for these expressions are provided below. For convenience, the AR model of Equation 1.1 is rewritten here as:

$$\begin{aligned} x_k &= f\left(x_{k-1}, \ldots x_{k-M}, \mathbf{w}\right) + v_k \\ y_k &= x_k + n_k, \qquad\qquad \forall k \in \{1 \ldots N\}. \end{aligned} \tag{1.1}$$

A.1 Joint Likelihood $\rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}}$

This conditional density appears as the first term in Equation 2.8, and can be thought of as a joint likelihood function for the signal and weights. By employing the definition of conditional densities, we can write:

$$\rho_{\mathbf{y}_1^N | \mathbf{x}_1^N \mathbf{w}} = \rho_{y_N | \mathbf{y}_1^{N-1} \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{\mathbf{y}_1^{N-1} | \mathbf{x}_1^N \mathbf{w}} \tag{A.1}$$

$$= \rho_{y_N | \mathbf{y}_1^{N-1} \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{y_{N-1} | \mathbf{y}_1^{N-2} \mathbf{x}_1^N \mathbf{w}} \cdot \rho_{\mathbf{y}_1^{N-2} | \mathbf{x}_1^N \mathbf{w}} \tag{A.2}$$

$$\cdots \tag{A.3}$$

$$= \rho_{y_1 | \mathbf{x}_1^N \mathbf{w}} \cdot \prod_{k=2}^{N} \rho_{y_k | \mathbf{y}_1^{k-1} \mathbf{x}_1^N \mathbf{w}} \tag{A.4}$$

which, because $y_k = x_k + n_k$, reduces to:

$$= \prod_{k=1}^{N} \rho_{y_k | x_k}. \tag{A.5}$$

If n_k is zero-mean white Gaussian noise, then $\rho_{\mathbf{y}_k|\mathbf{x}_k} \sim \mathcal{N}(x_k, \sigma_n^2)$, and each term in the product is

$$\begin{aligned} \rho_{y_k|x_k} &= \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(y_k - E[y_k|x_k])^2}{2\sigma_n^2}\right), \\ &= \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(y_k - x_k)^2}{2\sigma_n^2}\right), \end{aligned} \tag{A.6}$$

and

$$\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N\mathbf{w}} = \frac{1}{\sqrt{(2\pi)^N(\sigma_n^2)^N}} \exp\left(-\sum_{k=1}^{N} \frac{(y_k - x_k)^2}{2\sigma_n^2}\right). \tag{A.7}$$

This is the first term on the right hand side of Equation 2.9 on page 23.

A.2 Conditional Density $\rho_{\mathbf{x}_1^N|\mathbf{w}}$

This conditional density appears as the second term in Equation 2.8. It can be expanded as:

$$\rho_{\mathbf{x}_1^N|\mathbf{w}} = \rho_{x_N|\mathbf{x}_1^{N-1}\mathbf{w}} \cdot \rho_{\mathbf{x}_1^{N-1}|\mathbf{w}} \tag{A.8}$$

$$= \rho_{x_N|\mathbf{x}_1^{N-1}\mathbf{w}} \cdot \rho_{x_{N-1}|\mathbf{x}_1^{N-2}\mathbf{w}} \cdot \rho_{\mathbf{x}_1^{N-2}|\mathbf{w}} \tag{A.9}$$

$$\cdots \tag{A.10}$$

$$= \prod_{k=1}^{N} \rho_{x_k|\mathbf{x}_1^{k-1}\mathbf{w}} \tag{A.11}$$

which, because $x_k = f(x_{k-1}, \ldots, x_{k-M}, \mathbf{w}) + v_k$, reduces to:

$$= \prod_{k=1}^{N} \rho_{x_k|\mathbf{x}_{k-M}^{k-1}\mathbf{w}}. \tag{A.12}$$

If v_k is zero-mean white Gaussian noise, then each term in the product is

$$\begin{aligned} \rho_{x_k|\mathbf{x}_{k-M}^{k-1}\mathbf{w}} &= \frac{1}{\sqrt{2\pi\sigma_v^2}} \exp\left(-\frac{(x_k - x_k^-)^2}{2\sigma_v^2}\right), \\ \text{where} \quad x_k^- &\triangleq E[x_k|\{x_t\}_{k-M}^{k-1}, \mathbf{w}] \\ &= f(x_{k-1}, \ldots, x_{k-M}, \mathbf{w}) \\ &= f(\mathbf{x}_{k-1}, \mathbf{w}). \end{aligned} \tag{A.13}$$

Hence,

$$\rho_{\mathbf{x}_1^N|\mathbf{w}} = \frac{1}{\sqrt{(2\pi)^N(\sigma_v^2)^N}} \exp\left(-\sum_{k=1}^{N} \frac{(x_k - x_k^-)^2}{2\sigma_v^2}\right). \tag{A.14}$$

This is the second term on the right hand side of Equation 2.9 on page 23.

A.3 Marginal Likelihood $\rho_{\mathbf{y}_1^N|\mathbf{w}}$

This conditional density appears as the first term in Equation 2.48 on page 35, and is a marginal likelihood function for the weights. By employing the definition of conditional densities, we can write:

$$\rho_{\mathbf{y}_1^N|\mathbf{w}} = \rho_{y_N|\mathbf{y}_1^{N-1}\mathbf{w}} \cdot \rho_{\mathbf{y}_1^{N-1}|\mathbf{w}} \tag{A.15}$$

$$= \rho_{y_N|\mathbf{y}_1^{N-1}\mathbf{w}} \cdot \rho_{y_{N-1}|\mathbf{y}_1^{N-2}\mathbf{w}} \cdot \rho_{\mathbf{y}_1^{N-2}|\mathbf{w}} \tag{A.16}$$

$$\cdots \tag{A.17}$$

$$= \prod_{k=1}^{N} \rho_{y_k|\mathbf{y}_1^{k-1}\mathbf{w}}. \tag{A.18}$$

When the dynamics $f(\cdot)$ are linear, and both v_k and n_k are zero-mean white Gaussian processes, then each term in the product is:

$$\rho_{y_k|\mathbf{y}_1^{k-1}\mathbf{w}} = \frac{1}{\sqrt{2\pi\sigma_{\varepsilon_k}^2}} \exp\left(-\frac{(y_k - \overline{y_{k|k-1}})^2}{2\sigma_{\varepsilon_k}^2}\right), \quad \text{where} \quad \overline{y_{k|k-1}} \triangleq E[y_k|\{y_t\}_1^{k-1}, \mathbf{w}], \tag{A.19}$$

and

$$\rho_{\mathbf{y}_1^N|\mathbf{w}} = \prod_{k=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{\varepsilon_k}^2}} \exp\left(-\frac{(y_k - \overline{y_{k|k-1}})^2}{2\sigma_{\varepsilon_k}^2}\right). \tag{A.20}$$

This is the expression given in Equation 2.49 on page 35. If $f(\cdot)$ is nonlinear, the densities $\rho_{y_k|\mathbf{y}_1^{k-1}\mathbf{w}}$ will lose their Gaussian form, and Equation 2.49 represents an approximation.

Appendix B

Second Marginal Expansion

This appendix investigates an alternative marginal cost function. Although no practical algorithm results from this exercise, the development is nonetheless interesting. In Section 2.4, the joint density is expanded into two terms as:

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

where the first term is maximized with respect to the signal, and the second term with respect to the weights.

An alternative to the above expansion is given by:

$$\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N} = \rho_{\mathbf{w} | \mathbf{y}_1^N \mathbf{x}_1^N} \cdot \rho_{\mathbf{x}_1^N | \mathbf{y}_1^N}. \tag{B.1}$$

This suggests an alternative estimation scheme, in which $\{\hat{x}_k\}_1^N$ is found by maximizing the second term, $\rho_{\mathbf{x}_1^N | \mathbf{y}_1^N}$, and $\hat{\mathbf{w}}$ is found by maximizing the first term, $\rho_{\mathbf{w} | \mathbf{y}_1^N \mathbf{x}_1^N}$.

Similar to the comment made in Section 2.4 about the first expanded form, note that $\mathbf{w}$ can be estimated from the first term alone, but to maximize $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$ with respect to $\{x_k\}_1^N$, both terms need to be maximized. While algorithms based on the first expanded form have appeared in the literature, the same is not true of the second form. This is primarily because of the difficulty in maximizing $\rho_{\mathbf{x}_1^N | \mathbf{y}_1^N}$, as is shown below.

B.1 Model-Free Signal Estimation

To begin, consider the estimation of $\{x_k\}_1^N$ *via* the second term. Applying Bayes rule, we see that:

$$\rho_{\mathbf{x}_1^N | \mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N | \mathbf{x}_1^N} \cdot \rho_{\mathbf{x}_1^N}}{\rho_{\mathbf{y}_1^N}}. \tag{B.2}$$

While the prior $\rho_{\mathbf{y}_1^N}$ can be ignored (it is independent of $\{x_k\}_1^N$), the same is not true of $\rho_{\mathbf{x}_1^N}$. In this case, the prior $\rho_{\mathbf{x}_1^N}$ is an important part of the density because it contains our knowledge that $\{x_k\}_1^N$ was generated by an autoregressive process.

To see the importance of this prior, consider maximizing the likelihood $\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N}$ alone. The corresponding cost function is:

$$J_{\mathbf{y}_1^N|\mathbf{x}_1^N}(x,\mathbf{w})(x) = \sum_{k=1}^{N}\left(\frac{(y_k - x_k)^2}{\sigma_n^2}\right), \tag{B.3}$$

which does not produce an interesting result. The maximum-likelihood estimates in this case would be $\{\hat{x}_k\}_1^N = \{y_k\}_1^N$.

The problem is that the time series $\{x_k\}_1^N$ has not been constrained to be generated by an autoregressive process. To make this restriction more explicit in the prior, we can rewrite it as $\rho_{\mathbf{x}_1^N|\mathcal{M}}$, where $\mathcal{M}$ represents the model *structure* of the autoregressive model, independent of a specific choice of parameters $\mathbf{w}$.

The prior can then be written as:

$$\rho_{\mathbf{x}_1^N|\mathcal{M}} = \rho_{\mathbf{x}_1^N|\mathcal{M}} \cdot \frac{\rho_{\mathbf{x}_1^N\mathbf{w}|\mathcal{M}}}{\rho_{\mathbf{x}_1^N\mathbf{w}|\mathcal{M}}} \tag{B.4}$$

$$= \frac{\rho_{\mathbf{x}_1^N\mathbf{w}|\mathcal{M}}}{\rho_{\mathbf{w}|\mathbf{x}_1^N\mathcal{M}}}. \tag{B.5}$$

This expression can be simplified by making the model structure $\mathcal{M}$ implicit in the parameters $\mathbf{w}$. That is,

$$\rho_{\mathbf{x}_1^N|\mathcal{M}} = \frac{\rho_{\mathbf{x}_1^N\mathbf{w}}}{\rho_{\mathbf{w}|\mathbf{x}_1^N}} = \frac{\rho_{\mathbf{x}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{w}}}{\rho_{\mathbf{w}|\mathbf{x}_1^N}}. \tag{B.6}$$

The density $\rho_{\mathbf{x}_1^N|\mathbf{y}_1^N}$ can now be written as:

$$\rho_{\mathbf{x}_1^N|\mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N}}{\rho_{\mathbf{y}_1^N}} \cdot \frac{\rho_{\mathbf{x}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{w}}}{\rho_{\mathbf{w}|\mathbf{x}_1^N}} \tag{B.7}$$

$$= \frac{\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{w}}}{\rho_{\mathbf{y}_1^N} \cdot \rho_{\mathbf{w}|\mathbf{x}_1^N}}. \tag{B.8}$$

Where we have used the fact that $\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N} = \rho_{\mathbf{y}_1^N|\mathbf{x}_1^N\mathbf{w}}$. Because $\rho_{\mathbf{w}}$ and $\rho_{\mathbf{y}_1^N}$ are independent of $\{x_k\}_1^N$, we can find $\{\hat{x}_k\}_1^N$ by maximizing the function:

$$\frac{\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}}}{\rho_{\mathbf{w}|\mathbf{x}_1^N}}, \tag{B.9}$$

or, equivalently, its log:

$$\log \rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}} - \log \rho_{\mathbf{w}|\mathbf{x}_1^N} \tag{B.10}$$

with respect to $\{x_k\}_1^N$. This produces a new cost function, given by:

$$\begin{aligned} J_{\mathbf{x}_1^N|\mathbf{y}_1^N}(x,\mathbf{w}) &= J^j(\mathbf{x}_1^N,\mathbf{w}) - J_{\mathbf{w}|\mathbf{x}_1^N}(x,\mathbf{w}), \\ \text{where} \qquad J_{\mathbf{w}|\mathbf{x}_1^N}(x,\mathbf{w}) &= \left(\mathbf{w}' - \hat{\mathbf{w}}(\{x_k\}_1^N)\right)^T P_{\mathbf{w}}^{-1}\left(\mathbf{w} - \hat{\mathbf{w}}(\{x_k\}_1^N)\right). \end{aligned} \tag{B.11}$$

The second term represents a penalty for estimates $\{\hat{x}_k\}_1^N$ that agree too well with the assumed value of $\mathbf{w}$. This effectively removes any bias on the solution which might result from the specific choice of $\mathbf{w}$.

Making use of this additional term in the cost function can prove difficult, however. In particular, since $\hat{\mathbf{w}}$ will typically be found by a nonlinear optimization procedure, there is no closed-form expression for $\hat{\mathbf{w}}$ as a function of $\{x_k\}_1^N$. Without this expression, the derivative of the weights with respect to the time-series cannot be computed, and the cost function cannot be minimized.

B.2 Signal-Based Weight Estimation

Assuming that a signal estimate $\{\hat{x}_k\}_1^N$ is found, however, weight estimates $\hat{\mathbf{w}}$ can now be found by maximizing the first term in the expansion of the joint density in Equation B.1, $\rho_{\mathbf{w}|\mathbf{y}_1^N\mathbf{x}_1^N}$. This can be written as:

$$\rho_{\mathbf{w}|\mathbf{y}_1^N\mathbf{x}_1^N} = \frac{\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}} \cdot \rho_{\mathbf{w}}}{\rho_{\mathbf{y}_1^N\mathbf{x}_1^N}}. \tag{B.12}$$

Since $\rho_{\mathbf{y}_1^N\mathbf{x}_1^N}$ is independent of $\mathbf{w}$, the numerator alone can be maximized to estimate the weights. Furthermore, the term $\rho_{\mathbf{w}}$ can be dropped if we assume that no prior information is available on the distribution of the weights. This leaves $\rho_{\mathbf{y}_1^N\mathbf{x}_1^N|\mathbf{w}}$ as the likelihood function for the weights. As described in Section 2.3, this can be maximized by minimizing one of the cost functions given in Equations 2.13, 2.14 on page 25, or for error coupling, Equations 2.19, 2.20 on page 28.

However, if $\{\hat{x}_k\}_1^N$ is actually obtained by maximizing $\rho_{\mathbf{x}_1^N|\mathbf{y}_1^N}$, then it will be independent of $\mathbf{w}$. In this case, the versions of the costs which reflect this independence (namely Equations 2.14 and 2.20) are the most appropriate.

Appendix C

Kalman Filtering

The Kalman filter [36] generates optimal state estimates for linear systems. In this appendix, the Kalman filter is derived from the MAP perspective, both in the context of signal estimation, and in the context of weight estimation.

C.1 Signal Estimation

Recall the linear state-space representation for a noisy time-series $\{y_k\}_1^N$, given in Equations 3.11 and 3.12 on page 47:

$$\mathbf{x}_k = \mathbf{A} \cdot \mathbf{x}_{k-1} + \mathbf{B} \cdot v_k$$
$$y_k = \mathbf{C} \cdot \mathbf{x}_k + n_k.$$

Section 3.2 showed how sequential estimation of the signal $\{x_k\}_1^N$ requires recursive estimation of the state $\mathbf{x}_k$. This involves the two steps illustrated in Figure 3.2: (1) the generation of the posterior statistics from the prior statistics, and (2) generation of the prior statistics from the posterior statistics at the previous time step.

C.1.1 Posterior State Estimation

The posterior mean of the state is defined as:

$$\hat{\mathbf{x}}_k = E[\mathbf{x}_k | \{y_t\}_1^k, \mathbf{w}] \tag{C.1}$$

which is equivalent to the MAP estimate:

$$\hat{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k} \rho_{\mathbf{x}_k|\mathbf{y}_1^k,\mathbf{w}} \tag{C.2}$$

when the statistics are Gaussian. The solution to the MAP formulation of the problem is shown below.

The posterior density for the state can be expanded by as

$$\rho_{\mathbf{x}_k|\mathbf{y}_1^k,\mathbf{w}} = \frac{\rho_{y_k|\mathbf{y}_1^{k-1}\mathbf{x}_k\mathbf{w}} \cdot \rho_{\mathbf{x}_k|\mathbf{y}_1^{k-1}\mathbf{w}} \cdot \rho_{\mathbf{y}_1^{k-1}|\mathbf{w}}}{\rho_{\mathbf{y}_1^k|\mathbf{w}}}. \tag{C.3}$$

Note that $\rho_{y_k|\mathbf{y}_1^{k-1}\mathbf{x}_k\mathbf{w}} = \rho_{y_k|\mathbf{x}_k}$. Also, $\rho_{\mathbf{y}_1^k|\mathbf{w}}$ can be dropped because it is functionally independent of $\hat{\mathbf{x}}$; hence, the MAP state estimate is found as:

$$\hat{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k}\left(\rho_{y_k|\mathbf{x}_k} \cdot \rho_{\mathbf{x}_k|\mathbf{y}_1^{k-1}\mathbf{w}}\right). \tag{C.4}$$

Under the Gaussian assumption, the two terms can be written out explicitly as:

$$\rho_{y_k|\mathbf{x}_k} = \frac{1}{\sqrt{\log 2\pi\sigma_n^2}} \cdot \exp\left\{-\frac{1}{2}(y_k - \mathbf{C}\mathbf{x}_k)(\sigma_n^2)^{-1}(y_k - \mathbf{C}\mathbf{x}_k)^T\right\}$$
$$\rho_{\mathbf{x}_k|\mathbf{y}_1^{k-1}\mathbf{w}} = \frac{1}{\sqrt{(\log 2\pi)^M|\mathbf{P}_k^-|}} \cdot \exp\left\{-\frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T(\mathbf{P}_k^-)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)\right\},$$

where $\hat{\mathbf{x}}_k^- \triangleq E[\mathbf{x}_k|\{y_t\}_1^{k-1}, \mathbf{w}]$ and $\mathbf{P}_k^- \triangleq E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^2|\{y_t\}_1^{k-1}, \mathbf{w}]$ represent the prior mean and covariance of the state. Therefore, taking the negative log of $\left(\rho_{y_k|\mathbf{x}_k} \cdot \rho_{\mathbf{x}_k|\mathbf{y}_1^{k-1}\mathbf{w}}\right)$ yields

$$\begin{aligned} \alpha \quad &+ \quad \frac{1}{2}(y_k - \mathbf{C}\mathbf{x}_k)^T\sigma_n^{-2}(y_k - \mathbf{C}\mathbf{x}_k) \\ &+ \quad \frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T(\mathbf{P}_k^-)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-), \end{aligned} \tag{C.5}$$

where α is a constant to account for the normalizing terms in the Gaussian density functions. Hence, $\hat{\mathbf{x}}_k$ can be found by minimizing the expression in Equation C.5. This is done by taking the derivative with respect to the unknown $\mathbf{x}_k$ and setting it to zero:

$$\frac{\partial \ln(\rho_{\mathbf{x}_k|\mathbf{y}_1^k,\mathbf{w}})}{\partial \mathbf{x}_k} = (\mathbf{P}_k^-)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{C}^T\sigma_n^{-2}(y_k - \mathbf{C}\mathbf{x}_k) \tag{C.6}$$

$$\Rightarrow \quad 0 = (\mathbf{P}_k^-)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{C}^T\sigma_n^{-2}[y_k - \mathbf{C}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{C}\hat{\mathbf{x}}_k^-] \tag{C.7}$$

Collecting $(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)$ terms on the left hand side gives:

$$(\mathbf{P}_k^-)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{C}^T\sigma_n^{-2}\mathbf{C}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) = \mathbf{C}^T\sigma_n^{-2}[y_k - \mathbf{C}\hat{\mathbf{x}}_k^-] \tag{C.8}$$

$$\text{or} \qquad \left((\mathbf{P}_k^-)^{-1} + \mathbf{C}^T\sigma_n^{-2}\mathbf{C}\right)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) = \mathbf{C}^T\sigma_n^{-2}[y_k - \mathbf{C}\hat{\mathbf{x}}_k^-]. \tag{C.9}$$

and solving for $\mathbf{x}_k$ yields:

$$\mathbf{x}_k = \hat{\mathbf{x}}_k^- + \left((\mathbf{P}_k^-)^{-1} + \mathbf{C}^T\sigma_n^{-2}\mathbf{C}\right)^{-1}\mathbf{C}^T\sigma_n^{-2}[y_k - \mathbf{C}\hat{\mathbf{x}}_k^-]. \tag{C.10}$$

Letting $\hat{\mathbf{x}}_k$ take the value of the solution, this can be rewritten in the more familiar form as:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-), \tag{C.11}$$
$$\text{where} \quad \mathbf{K}_k \triangleq \left((\mathbf{P}_k^-)^{-1} + \mathbf{C}^T\sigma_n^{-2}\mathbf{C}\right)^{-1}\mathbf{C}^T\sigma_n^{-2}$$

$$(A^{-1} + BDC)^{-1} = A - AB(CAB + D^{-1})^{-1}CA$$

Formula C.1: The matrix inversion lemma.

is commonly referred to as the *Kalman gain.*

Note that computing the gain $\mathbf{K}_k$ involves inverting an $M \times M$ dimensional matrix, where M is the length of the state vector $\mathbf{x}_k$. This can be a relatively expensive procedure for large state vectors. Alternatively, the *matrix inversion lemma* (see Formula C.1) allows $\mathbf{K}_k$ to be written in a form that involves inverting a matrix with the same dimension as the measurement, y_k (in this case a scalar).

Applying the matrix inversion lemma to:

$$\mathbf{K}_k = \left((\mathbf{P}_k^-)^{-1} + \mathbf{C}^T\sigma_n^{-2}\mathbf{C}\right)^{-1}\mathbf{C}^T\sigma_n^{-2} \qquad \text{gives} \tag{C.12}$$

$$\mathbf{K}_k = \left(\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1}\mathbf{C}\mathbf{P}_k^-\right)\mathbf{C}^T\sigma_n^{-2}, \tag{C.13}$$

which can be simplified by the following algebraic steps:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T\sigma_n^{-2} - \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1}\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T\sigma_n^{-2} \tag{C.14}$$

$$= \mathbf{P}_k^-\mathbf{C}^T[\sigma_n^{-2} - (\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1}\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T\sigma_n^{-2}] \tag{C.15}$$

$$= \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1}[(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)\sigma_n^{-2} - \mathbf{C}\mathbf{P}_k^-\mathbf{C}^T\sigma_n^{-2}] \tag{C.16}$$

$$= \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1}[(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T\sigma_n^{-2} + 1) - \mathbf{C}\mathbf{P}_k^-\mathbf{C}^T\sigma_n^{-2}] \tag{C.17}$$

leaving the commonly-used form:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1}. \tag{C.18}$$

For one-dimensional time-series data, $\mathbf{C}$ has dimension $1 \times M$, so the above equation involves inverting only a scalar.

C.1.2 Posterior Covariance Estimation

Recall that to continue sequential estimation of the state, the posterior error covariance $\mathbf{P}_k$ is also required. This can be found by using the definition:

$$\mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T], \tag{C.19}$$

and substituting the definition of $\hat{\mathbf{x}}_k$ in Equation C.11 to give:

$$\mathbf{P}_k = E[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)\right)^T]. \tag{C.20}$$

Multiplying out the quadratic produces:

$$\begin{aligned}\mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \\ -E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)^T]\mathbf{K}_k^T \\ -\mathbf{K}_k E[(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \\ +\mathbf{K}_k E[(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)^T]\mathbf{K}_k^T\end{aligned} \tag{C.21}$$

While the first term on the right hand side of Equation C.21 evaluates immediately to $\mathbf{P}_k^-$, evaluation of the second, third, and fourth terms in this last expression involves rewriting $(y_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)$ as:

$$\begin{aligned}(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-) &= (\mathbf{C}\mathbf{x}_k + n_k) - \mathbf{C}\hat{\mathbf{x}}_k^- \\ &= \mathbf{C}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + n_k,\end{aligned} \tag{C.22}$$

so that the second term in Equation C.21 contains

$$\begin{aligned}E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-)^T] &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T\mathbf{C}^T] + E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)n_k] \\ &= \mathbf{P}_k^-\mathbf{C}^T,\end{aligned} \tag{C.23}$$

where the cross-term vanished because the measurement noise n_k is assumed to be white, and therefore uncorrelated with $(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)$. The third term in Equation C.21 is simply the transpose of the second. The fourth term contains:

$$\begin{aligned}E[(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-)(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-)^T] =& \mathbf{C}E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T]\mathbf{C}^T + \mathbf{C}E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)n_k] \\ &+ E[n_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T]\mathbf{C}^T + E[n_k n_k]\end{aligned}$$

where the cross-terms are again dropped to give:

$$= \mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2, \tag{C.24}$$

Substituting the terms C.23 and C.24 into Equation C.21 yields:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{C}^T\mathbf{K}_k^T - \mathbf{K}_k\mathbf{C}\mathbf{P}_k^- + \mathbf{K}_k(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)\mathbf{K}_k^T \tag{C.25}$$

which, using $\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \sigma_n^2)^{-1}$, gives:

$$\begin{aligned}\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{C}^T\mathbf{K}_k^T - \mathbf{K}_k\mathbf{C}\mathbf{P}_k^- + \mathbf{P}_k^-\mathbf{C}^T\mathbf{K}_k^T \\ &= \mathbf{P}_k^- - \mathbf{K}_k\mathbf{C}\mathbf{P}_k^- \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^-.\end{aligned} \tag{C.26}$$

This provides the posterior error covariance $\mathbf{P}_k$ as a linear function of the prior covariance $\mathbf{P}_k^-$.

C.1.3 Prior Estimation

Now that equations have been obtained for $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$, it remains to be shown how $\hat{\mathbf{x}}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$ are generated for the next time step. Using the state-space equations makes this fairly straightforward. The prior state estimate is:

$$\begin{aligned}\hat{\mathbf{x}}_{k+1}^- &= E[\mathbf{x}_{k+1}|\{y_t\}_1^k, \mathbf{w}] \\ &= E[\mathbf{A}\mathbf{x}_k + \mathbf{B}v_{k-1}|\{y_t\}_1^k, \mathbf{w}] \\ &= \mathbf{A}E[\mathbf{x}_k|\{y_t\}_1^k, \mathbf{w}] + \mathbf{B}E[v_{k-1}|\{y_t\}_1^k, \mathbf{w}] \\ &= \mathbf{A}\hat{\mathbf{x}}_k,\end{aligned} \tag{C.27}$$

where the conditional expectation of v_{k-1} is zero under the assumption that the process noise is white.

The prior covariance is obtained readily as:

$$\begin{aligned}\mathbf{P}_{k+1}^- &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)^T|\{y_t\}_1^k, \mathbf{w}] \\ &= E[(\mathbf{A}\mathbf{x}_k + \mathbf{B}v_{k-1} - \mathbf{A}\hat{\mathbf{x}}_k)(\mathbf{A}\mathbf{x}_k + \mathbf{B}v_{k-1} - \mathbf{A}\hat{\mathbf{x}}_k)^T|\{y_t\}_1^k, \mathbf{w}] \\ &= E[(\mathbf{A}(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \mathbf{B}v_{k-1})(\mathbf{A}(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \mathbf{B}v_{k-1})^T|\{y_t\}_1^k, \mathbf{w}] \\ &= \mathbf{A}E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T|\{y_t\}_1^k, \mathbf{w}]\mathbf{A}^T + \mathbf{B}E[(v_{k-1})^2|\{y_t\}_1^k, \mathbf{w}]\mathbf{B}^T \\ &= \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{B}\sigma_v^2\mathbf{B}^T\end{aligned} \tag{C.28}$$

These equations for generating the prior mean and covariance from the posteriors are often referred to as the *time-update* equations of the Kalman filter. The equations for generating posteriors from the priors are referred to as the *measurement-update* equations. Both sets of Kalman filter equations are summarized in Formula 3.1.

C.2 Weight Estimation

The Kalman filter can also be used to produce optimal sequential estimates of the weights when the clean signal x_k is known. The following development exactly parallels that just presented for state estimation, with the addition of a description of recursive least squares at the end of the section.

The linear state-space representation:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{u}_k \tag{C.29}$$

$$x_k = \mathbf{w}_k^T\mathbf{x}_{k-1} + v_k \tag{C.30}$$

is used to characterize the weights as a (stationary) random walk.

The MAP estimate of the weights is defined in Equation 3.45 on page 55 as:

$$\hat{\mathbf{w}}_k = \arg\max_{\mathbf{w}} \rho_{\mathbf{w}|\mathbf{x}_1^k},$$

and the corresponding sequential MAP cost is derived as:

$$J_k(\mathbf{w}) = \frac{(x_k - x_k^-)^2}{\sigma_v^2} + (\mathbf{w} - \hat{\mathbf{w}}_k^-)^T (\mathbf{Q}_k^-)^{-1} (\mathbf{w} - \hat{\mathbf{w}}_k^-),$$

where the signal prediction is written in the linear case as $x_k^- = \mathbf{x}_{k-1}^T \mathbf{w}_k$. The prior weight estimate and error covariance are given in Equation 3.54 and 3.55 on page 57 as:

$$\begin{aligned} \hat{\mathbf{w}}_k^- &= \hat{\mathbf{w}}_{k-1} \qquad \text{where} \quad \hat{\mathbf{w}}_{k-1} \triangleq E[\mathbf{w}_{k-1} | \{x_t\}_1^{k-1}], \quad \text{and} \\ \mathbf{Q}_k^- &= \mathbf{Q}_{k-1} + \mathbf{U}_k, \qquad \text{where} \quad \mathbf{Q}_{k-1} \triangleq E[(\mathbf{w} - \hat{\mathbf{w}}_{k-1})(\cdot)^T | \{x_t\}_1^{k-1}]. \end{aligned}$$

Hence, the prior weight estimate $\hat{\mathbf{w}}_k^-$ and covariance $\mathbf{Q}_k^-$ are directly dependent on the posterior estimate $\hat{\mathbf{w}}_{k-1}$ and covariance $\mathbf{Q}_{k-1}$ from the previous time step. These equations constitute the time-update of the Kalman weight filter. The measurement-update involves computing the posterior mean and covariance from the priors. Assuming Gaussian statistics, this can be done by finding the MAP estimate.

C.2.1 Posterior Weight Estimation

The MAP estimate of the weights is found by taking the derivative of $J_k(\mathbf{w})$ with respect to $\mathbf{w}$, and setting it to zero:

$$\frac{\partial \ln(\rho_{\mathbf{w}_k|\mathbf{x}_1^k})}{\partial \mathbf{w}} = (\mathbf{Q}_k^-)^{-1}(\mathbf{w} - \hat{\mathbf{w}}_k^-) - \frac{1}{\sigma_v^2}\mathbf{x}_{k-1}(x_k - \mathbf{x}_{k-1}^T \mathbf{w}) \tag{C.31}$$

$$\Rightarrow \quad 0 = (\mathbf{Q}_k^-)^{-1}(\mathbf{w} - \hat{\mathbf{w}}_k^-) - \sigma_v^{-2}\mathbf{x}_{k-1}[x_k - \mathbf{x}_{k-1}^T(\mathbf{w} - \hat{\mathbf{w}}_k^-) - \mathbf{x}_{k-1}^T \hat{\mathbf{w}}_k^-] \tag{C.32}$$

Collecting $(\mathbf{w} - \hat{\mathbf{w}}_k^-)$ terms on the left hand side gives:

$$(\mathbf{Q}_k^-)^{-1}(\mathbf{w} - \hat{\mathbf{w}}_k^-) + \sigma_v^{-2}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T(\mathbf{w} - \hat{\mathbf{w}}_k^-) = \sigma_v^{-2}\mathbf{x}_{k-1}[x_k - \mathbf{x}_{k-1}^T \hat{\mathbf{w}}_k^-] \tag{C.33}$$

$$\text{or} \qquad \left((\mathbf{Q}_k^-)^{-1} + \sigma_v^{-2}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T\right)(\mathbf{w} - \hat{\mathbf{w}}_k^-) = \sigma_v^{-2}\mathbf{x}_{k-1}[x_k - \mathbf{x}_{k-1}^T \hat{\mathbf{w}}_k^-], \tag{C.34}$$

and solving for $\mathbf{w}$ yields:

$$\mathbf{w} = \hat{\mathbf{w}}_k^- + \left((\mathbf{Q}_k^-)^{-1} + \mathbf{x}_{k-1}\sigma_v^{-2}\mathbf{x}_{k-1}^T\right)^{-1}\mathbf{x}_{k-1}\sigma_v^{-2}[x_k - \mathbf{x}_{k-1}^T \hat{\mathbf{w}}_k^-]. \tag{C.35}$$

Letting $\hat{\mathbf{w}}_k$ take the value of the solution, this can be rewritten in the more familiar form as:

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}}(x_k - \mathbf{x}_{k-1}^T \hat{\mathbf{w}}_k^-), \tag{C.36}$$

$$\text{where} \quad \mathbf{K}_k^{\mathbf{w}} \triangleq \left((\mathbf{Q}_k^-)^{-1} + \mathbf{x}_{k-1}\sigma_v^{-2}\mathbf{x}_{k-1}^T\right)^{-1}\mathbf{x}_{k-1}\sigma_v^{-2}$$

is the Kalman gain for weight estimation. Using the matrix inversion lemma as in Equations C.12-C.18 allows $\mathbf{K}_k^{\mathbf{w}}$ to be written in the alternate form:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^- \mathbf{x}_{k-1}(\mathbf{x}_{k-1}^T \mathbf{Q}_k^- \mathbf{x}_{k-1} + \sigma_v^2)^{-1}. \tag{C.37}$$

C.2.2 Posterior Covariance of Weights

The posterior covariance of the weights is

$$\mathbf{Q}_k = E[(\mathbf{w} - \hat{\mathbf{w}}_k)(\mathbf{w} - \hat{\mathbf{w}}_k)^T | \{x_t\}_1^k]. \tag{C.38}$$

Substituting the definition of $\hat{\mathbf{w}}_k$ in Equation C.36 gives:

$$\mathbf{Q}_k = E[\big(\mathbf{w} - \hat{\mathbf{w}}_k^- - \mathbf{K}_k^{\mathbf{w}}(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)\big)\big(\mathbf{w} - \hat{\mathbf{w}}_k^- - \mathbf{K}_k^{\mathbf{w}}(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)\big)^T]. \tag{C.39}$$

Multiplying out the quadratic produces:

$$\begin{aligned}
\mathbf{Q}_k = & E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T] \\
& - E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)^T](\mathbf{K}_k^{\mathbf{w}})^T \\
& - \mathbf{K}_k^{\mathbf{w}} E[(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T] \\
& + \mathbf{K}_k^{\mathbf{w}} E[(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)^T](\mathbf{K}_k^{\mathbf{w}})^T
\end{aligned} \tag{C.40}$$

While the first term on the right hand side of Equation C.40 evaluates immediately to $\mathbf{Q}_k^-$, evaluation of the second, third, and fourth terms in this last expression requires rewriting $(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)$ as:

$$\begin{aligned}
(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-) &= (\hat{\mathbf{x}}_{k-1}^T \mathbf{w} + v_k) - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^- \\
&= \hat{\mathbf{x}}_{k-1}^T (\mathbf{w} - \hat{\mathbf{w}}_k^-) + v_k.
\end{aligned} \tag{C.41}$$

Hence, the second term in Equation C.40 contains

$$\begin{aligned}
E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)^T] &= E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T \hat{\mathbf{x}}_{k-1}] + E[(\mathbf{w} - \hat{\mathbf{w}}_k^-) v_k] \\
&= \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1},
\end{aligned} \tag{C.42}$$

where the cross-term vanished because the process noise v_k is white, and is therefore uncorrelated with $(\mathbf{w} - \hat{\mathbf{w}}_k^-)$. The third term in Equation C.40 is simply the transpose of the second. The fourth term contains:

$$\begin{aligned}
E[(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)(x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_k^-)^T] = & \hat{\mathbf{x}}_{k-1}^T E[(\mathbf{w} - \hat{\mathbf{w}}_k^-)(\mathbf{w} - \hat{\mathbf{w}}_k^-)^T] \hat{\mathbf{x}}_{k-1} \\
& + \hat{\mathbf{x}}_{k-1}^T E[(\mathbf{w} - \hat{\mathbf{w}}_k^-) v_k] \\
& + E[v_k (\mathbf{w} - \hat{\mathbf{w}}_k^-)^T] \hat{\mathbf{x}}_{k-1} + E[v_k v_k]
\end{aligned}$$

where the cross-terms are again dropped to give:

$$= \hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} + \sigma_v^2, \tag{C.43}$$

Substituting the terms C.42 and C.43 into Equation C.40 yields:

$$\mathbf{Q}_k = \mathbf{Q}_k^- - \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} (\mathbf{K}_k^{\mathbf{w}})^T - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- + \mathbf{K}_k^{\mathbf{w}} (\hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} + \sigma_v^2)(\mathbf{K}_k^{\mathbf{w}})^T \tag{C.44}$$

which, using $\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} (\hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} + \sigma_v^2)^{-1}$, gives:

$$\begin{aligned} \mathbf{Q}_k &= \mathbf{Q}_k^- - \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} (\mathbf{K}_k^{\mathbf{w}})^T - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- + \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} (\mathbf{K}_k^{\mathbf{w}})^T \\ &= \mathbf{Q}_k^- - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- \\ &= (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T) \mathbf{Q}_k^-. \end{aligned} \tag{C.45}$$

This provides the posterior error covariance $\mathbf{Q}_k$ as a linear function of the prior covariance $\mathbf{Q}_k^-$.

The covariance can also be written another way by substituting the definition of $\mathbf{K}_k^{\mathbf{w}}$ to give:

$$\mathbf{Q}_k = (\mathbf{I} - \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} (\hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} + \sigma_v^2)^{-1} \hat{\mathbf{x}}_{k-1}^T) \mathbf{Q}_k^- \tag{C.46}$$

$$= \mathbf{Q}_k^- - \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} (\hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- \hat{\mathbf{x}}_{k-1} + \sigma_v^2)^{-1} \hat{\mathbf{x}}_{k-1}^T \mathbf{Q}_k^- \tag{C.47}$$

$$= ((\mathbf{Q}_k^-)^{-1} + \hat{\mathbf{x}}_{k-1} \sigma_v^{-2} \hat{\mathbf{x}}_{k-1}^T)^{-1}, \tag{C.48}$$

where the last step follows directly from the matrix inversion lemma. With this equation in place, an alternative expression can now be obtained for the Kalman gain in Equation C.36:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k \hat{\mathbf{x}}_{k-1} \sigma_v^{-2}. \tag{C.49}$$

This expression is not generally used in the Kalman weight filter, but it is useful for showing the relationship between Kalman weight filtering and the modified Gauss-Newton optimization technique.

C.2.3 Recursive Least Squares

As stated in Section 3.3, RLS can be viewed as a special case of the Kalman weight filter by constraining the covariance of $\mathbf{u}_k$ such that

$$\mathbf{Q}_k^- = \lambda^{-1} \mathbf{Q}_{k-1}. \tag{C.50}$$

By defining:

$$\mathbf{\Sigma}_k \triangleq \left(\frac{1}{\sigma_v^2} \mathbf{Q}_k\right)^{-1} = \left(\frac{\lambda}{\sigma_v^2} \mathbf{Q}_{k+1}^-\right)^{-1}, \tag{C.51}$$

the weight measurement update Equations 3.60-3.62 from Formula 3.5:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{Q}_k^- \mathbf{x}_{k-1} (\mathbf{x}_{k-1}^T \mathbf{Q}_k^- \mathbf{x}_{k-1} + \sigma_v^2)^{-1} \quad ((3.60))$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} (x_k - \mathbf{x}_{k-1}^T \hat{\mathbf{w}}_k^-) \quad ((3.61))$$

$$\mathbf{Q}_k = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \mathbf{x}_{k-1}^T) \mathbf{Q}_k^- \quad ((3.62))$$

can be replaced by:

$$\mathbf{K}_k^{\mathbf{w}} = \boldsymbol{\Sigma}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} (\hat{\mathbf{x}}_{k-1}^T \boldsymbol{\Sigma}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} + \lambda)^{-1} \quad (C.52)$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} + \mathbf{K}_k^{\mathbf{w}} (x_k - \hat{\mathbf{x}}_{k-1}^T \hat{\mathbf{w}}_{k-1}) \quad (C.53)$$

$$\boldsymbol{\Sigma}_k^{-1} = \lambda^{-1} (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T) \boldsymbol{\Sigma}_{k-1}^{-1}, \quad (C.54)$$

which is the RLS algorithm in its more efficient form. An equivalent form is obtained by substituting Equation C.52 into Equation C.54 to get:

$$\boldsymbol{\Sigma}_k^{-1} = \lambda^{-1} \left(\boldsymbol{\Sigma}_{k-1}^{-1} - \boldsymbol{\Sigma}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} (\hat{\mathbf{x}}_{k-1}^T \boldsymbol{\Sigma}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} + \lambda)^{-1} \hat{\mathbf{x}}_{k-1}^T \boldsymbol{\Sigma}_{k-1}^{-1} \right) \quad (C.55)$$

which, by the matrix inversion lemma (Formula C.1), is:

$$= \lambda^{-1} (\boldsymbol{\Sigma}_{k-1} + \hat{\mathbf{x}}_{k-1} \lambda^{-1} \hat{\mathbf{x}}_{k-1}^T)^{-1} \quad (C.56)$$

implying that

$$\boldsymbol{\Sigma}_k = \lambda (\boldsymbol{\Sigma}_{k-1} + \hat{\mathbf{x}}_{k-1} \lambda^{-1} \hat{\mathbf{x}}_{k-1}^T) \quad (C.57)$$

$$= \lambda \boldsymbol{\Sigma}_{k-1} + \hat{\mathbf{x}}_{k-1} \hat{\mathbf{x}}_{k-1}^T. \quad (C.58)$$

This is the traditional form of the RLS update of the data covariance matrix. Rewriting Equation C.49 on the previous page as $\mathbf{K}_k^{\mathbf{w}} = \boldsymbol{\Sigma}_k^{-1} \hat{\mathbf{x}}_{k-1}$ allows the weight update in Equation C.53 to be written as

$$\hat{\mathbf{w}}_k = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T) \hat{\mathbf{w}}_{k-1} + \mathbf{K}_k^{\mathbf{w}} x_k \quad (C.59)$$

$$= (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T) \hat{\mathbf{w}}_{k-1} + \boldsymbol{\Sigma}_k^{-1} \mathbf{x}_{k-1} x_k \quad (C.60)$$

$$= \lambda^{-1} (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \hat{\mathbf{x}}_{k-1}^T) \boldsymbol{\Sigma}_{k-1}^{-1} \lambda \boldsymbol{\Sigma}_{k-1} \hat{\mathbf{w}}_{k-1} + \boldsymbol{\Sigma}_k^{-1} \mathbf{x}_{k-1} x_k \quad (C.61)$$

which, by Equation C.54 , is:

$$\hat{\mathbf{w}}_k = \boldsymbol{\Sigma}_k^{-1} (\lambda \boldsymbol{\Sigma}_{k-1} \hat{\mathbf{w}}_{k-1} + \mathbf{x}_{k-1} x_k) \quad (C.62)$$

$$= \boldsymbol{\Sigma}_k^{-1} (\lambda \boldsymbol{\beta}_{k-1} + \mathbf{x}_{k-1} x_k), \quad \text{where} \quad \boldsymbol{\beta}_k \triangleq \boldsymbol{\Sigma}_k \hat{\mathbf{w}}_k. \quad (C.63)$$

However, it immediately follows that:

$$\boldsymbol{\beta}_k = \boldsymbol{\Sigma}_k \boldsymbol{\Sigma}_k^{-1} (\lambda \boldsymbol{\Sigma}_{k-1} \hat{\mathbf{w}}_{k-1} + \mathbf{x}_{k-1} x_k) \tag{C.64}$$

$$= \lambda \boldsymbol{\Sigma}_{k-1} \hat{\mathbf{w}}_{k-1} + \mathbf{x}_{k-1} x_k \tag{C.65}$$

$$= \lambda \boldsymbol{\beta}_{k-1} + \mathbf{x}_{k-1} x_k. \tag{C.66}$$

Hence, the weights are given by:

$$\hat{\mathbf{w}}_k = \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\beta}_k, \tag{C.67}$$

and Equations 3.60-3.62 are equivalent to the RLS equations:

$$\boldsymbol{\Sigma}_k = \lambda \boldsymbol{\Sigma}_{k-1} + \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T \tag{C.68}$$

$$\boldsymbol{\beta}_k = \lambda \boldsymbol{\beta}_{k-1} + \mathbf{x}_{k-1} x_k \tag{C.69}$$

$$\hat{\mathbf{w}}_k = \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\beta}_k \tag{C.70}$$

as promised.

Appendix D

The EKF Approximation

The preceding appendix shows derivations of the Kalman signal and weight filters under the assumption of a linear state space system. In this appendix we consider the ramifications of applying Kalman filtering techniques to nonlinear systems. Because the nonlinearity of the signal filters used in this thesis is limited to the time-update, the measurement-update of the EKF is not addressed herein. The exact nature of the approximation made by the extended Kalman filter (EKF) time-update is considered first; this is followed by an analysis of the potential severity of this approximation.

D.1 Approximating the Expectation

Generating the prior mean $\hat{\mathbf{x}}_{k+1}^{-}$ and covariance $\mathbf{P}_{k+1}^{-}$ requires evaluating:

$$\hat{\mathbf{x}}_{k+1}^{-} = E\big[\mathbf{x}_{k+1}\big|\{y_t\}_1^k, \mathbf{w}\big] \tag{D.1a}$$

$$= E\big[\mathbf{F}(\mathbf{x}_k, \mathbf{w}) + \mathbf{B}v_{k+1}\big|\{y_t\}_1^k, \mathbf{w}\big] \tag{D.1b}$$

$$= E\big[\mathbf{F}(\mathbf{x}_k, \mathbf{w})\big|\{y_t\}_1^k, \mathbf{w}\big], \tag{D.1}$$

where the conditional expectation of v_{k+1} is zero under the assumption that the process noise is white. When the model is nonlinear, evaluation of this expectation is non-trivial. Recalling the structure of the vector function $\mathbf{F}(\cdot)$, gives:

$$\hat{\mathbf{x}}_{k+1}^{-} = \begin{bmatrix} E\big[f(\mathbf{x}_k, \mathbf{w})\big|\{y_t\}_1^k, \mathbf{w}\big] \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \hat{\mathbf{x}}_k \end{bmatrix}, \tag{D.2}$$

where $\hat{\mathbf{x}}_k = E\big[\mathbf{x}_k \big| \{y_t\}_1^k, \mathbf{w}\big]$. The EKF approximates the expectation of $f(\mathbf{x}_k, \mathbf{w})$ using a Taylor series expansion about $\hat{\mathbf{x}}_k$:

$$f(\mathbf{x}_k, \mathbf{w}) = f(\hat{\mathbf{x}}_k, \mathbf{w}) + \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{x}}^T (\mathbf{x}_k - \hat{\mathbf{x}}_k) + \frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \frac{\partial^2 f(\hat{\mathbf{x}}_k, \mathbf{w})}{(\partial \mathbf{x})^2}(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \cdots \tag{D.3}$$

and keeping only the first two terms:

$$f(\mathbf{x}_k, \mathbf{w}) \approx f(\hat{\mathbf{x}}_k, \mathbf{w}) + \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{x}}^T (\mathbf{x}_k - \hat{\mathbf{x}}_k). \tag{D.4}$$

The conditional expectation of the second term is zero, so expectation of the truncated Taylor series gives:

$$\hat{\mathbf{x}}_{k+1}^- \approx \mathbf{F}(\hat{\mathbf{x}}_k, \mathbf{w}). \tag{D.5}$$

Approximate evaluation of $\mathbf{P}_{k+1}^-$ requires writing the truncated Taylor series of the entire vector function:

$$\mathbf{F}(\mathbf{x}_k) \approx \mathbf{F}(\hat{\mathbf{x}}_k) + \mathbf{A}_k \cdot (\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{D.6}$$

where $\mathbf{A}_k$ is defined as:

$$\mathbf{A}_k \triangleq \frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{w})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_k} = \begin{bmatrix} \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{x}}^T \\ 1 \quad 0 \quad 0 \quad 0 \\ 0 \quad \ddots \quad 0 \quad \vdots \\ 0 \quad 0 \quad 1 \quad 0 \end{bmatrix}. \tag{D.7}$$

The prior covariance $\mathbf{P}_{k+1}^-$ is defined as:

$$\mathbf{P}_{k+1}^- = E\big[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)^T \big| \{y_t\}_1^k, \mathbf{w}\big] \tag{D.8a}$$

$$= E\big[(\mathbf{F}(\mathbf{x}_k) + \mathbf{B}v_{k+1} - \mathbf{F}(\hat{\mathbf{x}}_k))(\mathbf{F}(\mathbf{x}_k) + \mathbf{B}v_{k+1} - \mathbf{F}(\hat{\mathbf{x}}_k))^T \big| \{y_t\}_1^k, \mathbf{w}\big]. \tag{D.8}$$

Inserting the first order Taylor series approximation gives:

$$\mathbf{P}_{k+1}^- \approx E\Big[\big(\mathbf{F}(\hat{\mathbf{x}}_k) + \mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k) - \mathbf{F}(\hat{\mathbf{x}}_k) + \mathbf{B}v_{k+1}\big)(\,\cdot\,)^T \Big| \{y_t\}_1^k, \mathbf{w}\Big] \tag{D.9a}$$

$$\approx E\Big[\big(\mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \mathbf{B}v_{k+1}\big)\big(\mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \mathbf{B}v_{k+1}\big)^T \Big| \{y_t\}_1^k, \mathbf{w}\Big] \tag{D.9b}$$

$$\approx \mathbf{A}_k E\big[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \big| \{y_t\}_1^k, \mathbf{w}\big]\mathbf{A}_k^T + \mathbf{B}E[(v_{k+1})^2 | \{y_t\}_1^k, \mathbf{w}]\mathbf{B}^T \tag{D.9c}$$

$$\approx \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{B}\sigma_v^2 \mathbf{B}^T. \tag{D.9}$$

Equations D.5 and D.9 form the time-update equations for the extended Kalman filter. The measurement-update is the same as in the linear case, so the EKF is obtained merely by replacing

the KF time-update equations (C.27 and C.28) with the following:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}(\hat{\mathbf{x}}_{k-1}, \mathbf{w}) \tag{D.10}$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{B}\sigma_v^2\mathbf{B}^T, \tag{D.11}$$

as given in Formula 3.2 on page 51.

D.2 Severity of the EKF Approximation

This section investigates how close the EKF time-update approximations are to the true mean and covariance. If, for a particular application, the errors in the approximations are on the same order as the approximations themselves, then the EKF is of little practical use.

This concern can be addressed by considering the portion of the Taylor series that was disregarded during the truncation (*i.e.*, the higher order terms). In particular, defining the remainder term as[1]:

$$rem = \sum_{i=2}^{\infty} \frac{1}{i!} \frac{\partial^i f(\hat{\mathbf{x}}_k)}{(\partial \mathbf{x})^i} (\mathbf{x}_k - \hat{\mathbf{x}}_k)^i \tag{D.12}$$

allows the Taylor series can be rewritten as

$$f(\mathbf{x}_k, \mathbf{w}) = f(\hat{\mathbf{x}}_k, \mathbf{w}) + \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{x}}^T (\mathbf{x}_k - \hat{\mathbf{x}}_k) + rem. \tag{D.13}$$

Hence, the error in the EKF estimate of the mean $\hat{\mathbf{x}}_{k+1}^-$ has magnitude $E[rem|\{y_t\}_1^k, \mathbf{w}]$ (when $\hat{\mathbf{x}}_{k+1}^-$ is a vector, the error is restricted to the first element). Ideally, this value could be determined by seeing what the infinite series:

$$E\big[rem\big|\{y_t\}_1^k, \mathbf{w}\big] = \sum_{i=2}^{\infty} \frac{1}{i!} \frac{\partial^i f(\hat{\mathbf{x}}_k)}{(\partial \mathbf{x})^i} E\big[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^i\big|\{y_t\}_1^k, \mathbf{w}\big] \tag{D.14}$$

converges to. Of course, computing the central moments $E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^i|\{y_t\}_1^k, \mathbf{w}]$ of $\mathbf{x}_k$ requires knowledge of the current conditional distribution, and is not generally tractable. However, assuming that $\mathbf{x}_k$ is a Gaussian random variable (so far, we are still treating the scalar case), the required moments are easily computed. This Gaussianity assumption might only be valid for $k = 0$ (with Gaussian prior, $\mathbf{x}_0$), but is reasonable since the EKF propagates only the mean and covariance of the distribution anyway. If we accept that the Gaussian assumption is central to the EKF,

[1] Of course $\frac{\partial^i f}{(\partial \mathbf{x})^i}$ is a multidimension tensor in general, so this notation is incorrect unless $\mathbf{x}_k$ is a scalar ($M = 1$). For the sake of simplicity, then, consider only the scalar case for the time being.

the remaining question is: how well does EKF compute the mean and variance of a propagated Gaussian random variable?

Unfortunately, the central moments of a Gaussian distribution increase without bound with the order, i. If σ_{x_k} is the (conditional) standard deviation of $\mathbf{x}_k$, then [32]:

$$E\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^i \middle| \{y_t\}_1^k, \mathbf{w}\right] = \begin{cases} 0 & \text{all odd i} \geq 1, \\ 1 \cdot 3 \cdot 5 \cdots (i-1)\sigma_{x_k}^i & \text{all even i} \geq 2 \end{cases} \tag{D.15}$$

The rapid growth of the moments causes the sum in Equation D.14 to diverge, even for fairly pedestrian choices of $f(\cdot)$. For example, consider the simple function $f(x_k) = \tanh(x_k)$, which is a reasonable choice since the neural networks considered in this thesis incorporate this nonlinearity. The higher derivatives of $\tanh(x_k)$ also grow arbitrarily large, but when scaled by the inverse of the factorial, give a convergent family of functions, as shown in Figure D.1. However, a plot of the

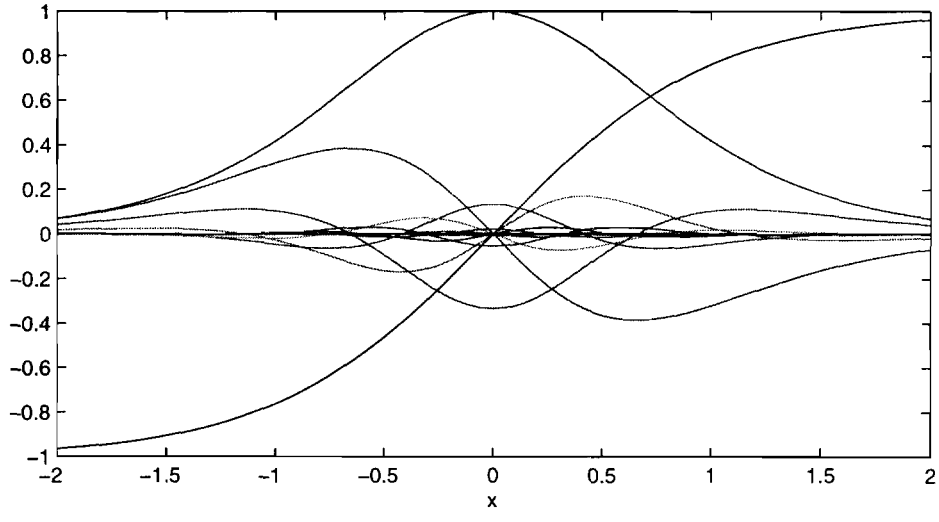


Figure D.1: The scaled derivatives of $f(x) = \tanh(x)$ appearing in the Taylor series: $\frac{\partial^i f(\hat{\mathbf{x}}_k)}{(\partial \mathbf{x})^i}$.

maximum value of each of these scaled derivatives as a function of the order i can be compared with a plot of the Gaussian central moments (see Figure D.2) to see that the series will always diverge after some value of i. Smaller values of $\sigma_{x_k}^2$ only increase the value of i after which the terms in Equation D.12 begin increasing without bound.

The problem of diverging moments can be circumvented by writing what is called the "Taylor series with a remainder term:"

$$f(\mathbf{x}_k, \mathbf{w}) = f(\hat{\mathbf{x}}_k, \mathbf{w}) + \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})}{\partial \mathbf{x}}^T (\mathbf{x}_k - \hat{\mathbf{x}}_k) + \frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \frac{\partial^2 f(\alpha \mathbf{x}_k + (1-\alpha)\hat{\mathbf{x}}_k, \mathbf{w})}{(\partial \mathbf{x})^2}(\mathbf{x}_k - \hat{\mathbf{x}}_k), \tag{D.16}$$

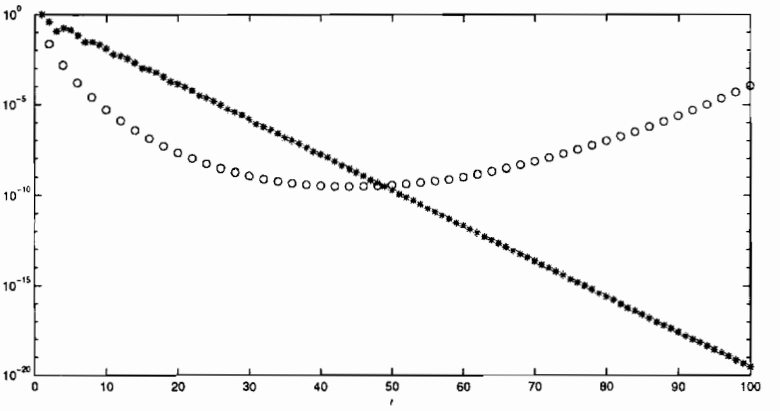


Figure D.2: The even Gaussian moments for $\sigma^2_{x_k} = .15$ (circles), compared with the scaled derivatives of tanh (*). For any value of $\sigma^2_{x_k}$, the moments will eventually grow faster than the scaled derivatives shrink.

where α is unspecified. The infinite sum has been replaced by a finite sum by redefining the remainder as:

$$rem = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{G}(\mathbf{x}_k, \mathbf{w})(\mathbf{x}_k - \hat{\mathbf{x}}_k), \tag{D.17}$$

where

$$\mathbf{G}(\mathbf{x}_k, \mathbf{w}) = \frac{1}{2} \frac{\partial^2 f(\alpha \mathbf{x}_k + (1-\alpha)\hat{\mathbf{x}}_k, \mathbf{w})}{(\partial \mathbf{x})^2}. \tag{D.18}$$

D.2.1 Error in the Mean

Note that we are now returning to the more general case where $\mathbf{x}_k$ is a vector. Generally, the expectation of the remainder cannot be computed in this form (and we have just seen the difficulties encountered when using the standard Taylor series expansion). However, an upper bound can be computed for $E[rem]$, as follows. First, define $\mathbf{d} \triangleq (\mathbf{x}_k - \hat{\mathbf{x}}_k)$, and let $d^{(j)}$ and $G^{(j,m)}$ denote individual elements of $\mathbf{d}$ and $\mathbf{G}$, respectively. Then the remainder term is:

$$rem = \sum_{j,m} G^{(j,m)} d^{(j)} d^{(m)} \tag{D.19a}$$

$$= \sum_{j,m} rem_{j,m}, \qquad \text{where} \quad rem_{j,m} = G^{(j,m)} d^{(j)} d^{(m)}. \tag{D.19}$$

Therefore (suppressing the conditioning arguments of the expectation from the notation):

$$E[rem] = \sum_{j,m} E[rem_{j,m}] \tag{D.20}$$

Taking each term separately,

$$E[rem_{j,m}] = E[G^{(j,m)} d^{(j)} d^{(m)}] \tag{D.21a}$$

$$= E[G^{(j,m)}] \cdot E[d^{(j)} d^{(m)}] + \varsigma \cdot \sigma_G \cdot \sigma_{dd}, \tag{D.21b}$$

where ς is the correlation coefficient, and σ_G and σ_{dd} are the standard deviations of $G^{(j,m)}$ and $d^{(j)} d^{(m)}$, respectively. Note that $E[d^{(j)} d^{(m)}] = \mathbf{P}^{(j,m)}$, so:

$$E[rem_{j,m}] = E[G^{(j,m)}] \cdot \mathbf{P}^{(j,m)} + \varsigma \cdot \sigma_G \cdot \sqrt{E\left[\left(d^{(j)} d^{(m)} - E[d^{(j)} d^{(m)}]\right)^2\right]} \tag{D.21c}$$

$$= E[G^{(j,m)}] \cdot \mathbf{P}^{(j,m)} + \varsigma \cdot \sigma_G \cdot \sqrt{E\left[(d^{(j)})^2 (d^{(m)})^2\right] - E\left[d^{(j)} d^{(m)}\right]^2} \tag{D.21d}$$

$$= E[G^{(j,m)}] \cdot \mathbf{P}^{(j,m)} + \varsigma \cdot \sigma_G \cdot \sqrt{E\left[(d^{(j)})^2 (d^{(m)})^2\right] - (\mathbf{P}^{(j,m)})^2} \tag{D.21}$$

Generally, the terms $E[G^{(j,m)}]$ and σ_G are not computable, and the correlation coefficient is unknown. However, the bounds:

$$E[G^{(j,m)}] < \max |G^{(j,m)}|, \qquad \sigma_G < 2 \max |G^{(j,m)}|, \quad \text{and} \quad \varsigma \leq 1 \tag{D.22}$$

can be employed to bound the expected remainder as:

$$E[rem] \leq \sum_{j,m} \max |G^{(j,m)}| \cdot \left(\mathbf{P}^{(j,m)} + 2\sqrt{E\left[(d^{(j)})^2 (d^{(m)})^2\right] - (\mathbf{P}^{(j,m)})^2}\right). \tag{D.23}$$

This gives an upper bound on the error in the EKF approximation to $\hat{\mathbf{x}}_{k+1}^-$, in terms of the second derivatives of $f(\cdot)$, and a combination of the second and fourth central moments of $\mathbf{x}_k$, assuming $\mathbf{x}_k$ is Gaussian. The inequality is somewhat easier to interpret when $\mathbf{x}_k$ is a scalar with conditional variance $\sigma_{x_k}^2$. In this case:

$$E[rem] \leq \max \left| \frac{1}{2} \frac{\partial^2 f(\hat{\mathbf{x}}_k, \mathbf{w})}{(\partial \mathbf{x})^2} \right| \cdot \left(\sigma_{x_k}^2 + 2\sqrt{3\sigma^4[x_k] - \sigma^4[x_k]}\right) \tag{D.24a}$$

$$E[rem] \leq \frac{1}{2} \max \left| \frac{\partial^2 f(\hat{\mathbf{x}}_k, \mathbf{w})}{(\partial \mathbf{x})^2} \right| \cdot (1 + 2\sqrt{2}) \sigma_{x_k}^2 \tag{D.24}$$

For the particular choice $f(\cdot) = \tanh(\cdot)$, the bound can be computed to be around $1.47 \cdot \sigma_{x_k}^2$. The value of $f(\hat{\mathbf{x}}_k)$ is between -1 and 1, so $\sigma_{x_k}^2 = .01$ will produce an error of at least 1.5%.

D.2.2 Error in the Covariance

A bound on the error in the EKF estimate of the covariance $\mathbf{P}_{k+1}^-$ can also be determined by analysis of the Taylor series remainder. Because in the time-series case the nonlinearity appears only in the first element of $\mathbf{F}(\mathbf{x}_k, \mathbf{w})$, the error is restricted to the top left corner of the covariance:

$(\mathbf{P}_{k+1}^{-})^{(0,0)}$. Substituting the Taylor series with remainder into Equation D.8 (instead of the truncated series) gives:

$$\mathbf{P}_{k+1}^{-} = E\Big[\big(\mathbf{F}(\hat{\mathbf{x}}_k) + \mathbf{B}rem + \mathbf{B}v_{k+1} + \mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k) - \mathbf{F}(\hat{\mathbf{x}}_k) - \mathbf{B}E[rem]\big)\,(\,\cdot\,)^T\Big] \tag{D.25a}$$

$$= E\Big[\big(\mathbf{B}(rem - E[rem]) + \mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \mathbf{B}v_{k+1}\big)\,(\,\cdot\,)^T\Big] \tag{D.25b}$$

$$\begin{aligned} &= \mathbf{A}_k E\big[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\big]\mathbf{A}_k^T + \mathbf{B}E\big[(rem - E[rem])^2\big]\mathbf{B}^T \\ &\quad + \mathbf{B}E\big[(v_{k+1})^2\big]\mathbf{B}^T \end{aligned} \tag{D.25c}$$

$$= \mathbf{A}_k\mathbf{P}_k\mathbf{A}_k^T + \mathbf{B}\sigma_v^2\mathbf{B}^T + \mathbf{B}\big(E[(rem)^2] - E[rem]^2\big)\mathbf{B}^T, \tag{D.25}$$

where we continue to suppress the conditioning arguments of the expectations for more compact notation. Hence, the error in the EKF covariance update is $\mathbf{B}\big(E[(rem)^2] - (E[rem])^2\big)\mathbf{B}^T$. As before, the expectations cannot be computed in general, so an upper bound on the error is sought. The maximum of a difference between two positive numbers is greater than the difference of the two maxima, so the desired bound on the error must be found as the maximum of the first term alone.

Starting with the component-wise definition of rem given in Equation D.19, an expression for rem^2 is obtained:

$$rem^2 = \sum_{j,m,n,o} G^{(j,m)}G^{(n,o)}d^{(j)}d^{(m)}d^{(n)}d^{(o)} \tag{D.26a}$$

$$= \sum_{j,m} rem^2_{j,m,n,o}, \qquad \text{where} \quad rem^2_{j,m,n,o} = G^{(j,m)}d^{(j)}d^{(m)}d^{(n)}d^{(o)}G^{(n,o)}. \tag{D.26}$$

Therefore:

$$E[rem^2] = \sum_{j,m,n,o} E[rem^2_{j,m,n,o}] \tag{D.27}$$

Following the approach used for the mean, an upper bound for the individual terms is found to yield the overall bound on the error in the top left element of the covariance:

$$\begin{aligned} E[rem^2] \leq \sum_{j,m,n,o} &\max\big[(G^{(j,m)})^2\big] \\ &\Big(E[d^{(j)}d^{(m)}d^{(n)}d^{(o)}] + 2\sqrt{E\big[(d^{(j)}d^{(m)}d^{(n)}d^{(o)})^2\big] - E[d^{(j)}d^{(m)}d^{(n)}d^{(o)}]^2}\Big), \end{aligned} \tag{D.28}$$

which is a function of the second derivatives of $f(\cdot)$, and the fourth and eighth central moments of $\mathbf{x}_k$. Like the bound on the error in the mean, this expression simplifies considerably in the scalar

case:

$$E[rem^2] \leq \frac{1}{4}\max\left[\left(\frac{\partial^2 f(\hat{\mathbf{x}}_k, \mathbf{w})}{(\partial \mathbf{x})^2}\right)^2\right] \cdot (3\sigma_{x_k}^4 + 2\sqrt{105\sigma_{x_k}^8 - 9\sigma_{x_k}^8}) \tag{D.29a}$$

$$\leq \frac{1}{4}\max\left[\left(\frac{\partial^2 f(\hat{\mathbf{x}}_k, \mathbf{w})}{(\partial \mathbf{x})^2}\right)^2\right] \cdot (3 + 8\sqrt{6})\sigma_{x_k}^4. \tag{D.29}$$

This gives a bound of around $(3.35 \cdot \sigma_{x_k}^4)$ for $f(\cdot) = \tanh(\cdot)$. For $\sigma_{x_k}^2 = .01$ the bound on the error is around 0.03%.

D.2.3 Conclusions

The EKF uses an approximate (and therefore suboptimal) method of calculating the mean and covariance of a Gaussian random variable passed through a nonlinear map. The errors in the mean and covariance are likely to accumulate to some degree over multiple time-steps, and the state will become increasingly non-Gaussian. However, the measurement update of the EKF has the effect of reducing the error in the mean, and shrinking the covariance, so the errors will not generally increase indefinitely. By exploring the errors made at each time-step, this appendix provides a first step towards the broader problem of understanding the effects of the EKF approximation as a recursive function of time. The following general conclusions can be drawn:

1. The error in the mean will be large if the state covariance is on the same scale as the nonlinearity. Hence, problems are more likely to arise when the state covariance is large; *i.e.*, when the measurement noise and process noise have high variance.

2. The error in the covariance will also depend on the scale of the nonlinearity with respect to the statistics of the state. However, Equation D.25 indicates that an adjustment to σ_v^2 can potentially be used to compensate for this error. Therefore, estimating σ_v^2 along with the state can sometimes produce better state estimates than when the true variance σ_v^2 is used.

The bounds in this appendix can help in determining when signal estimates might be significantly improved with a more expensive algorithm, such as a higher-order Kalman filter (incorporating additional Taylor series terms) [53], unscented Kalman filter [35], or particle filter [15].

Appendix E

Observed-Error Derivatives

The observed-error form of the weight filter, discussed in Section 3.3.2 on page 62, is based on the idea of approximating the gradient and Hessian of a cost function by choosing an appropriate form for the measurement equation in the state-space representation of the weights. This allows a Kalman weight filter to be used as an efficient, sequential modified-Newton algorithm.

A variety of cost functions can be minimized by altering the form of the observed-error vector, $\mathfrak{e}_k$; the appropriate form for the prediction error cost is shown on page 63. This appendix shows how the choices for the observed-error and its first derivative (appearing in Section 3.5 on page 70) approximate the gradient and Hessian of the other four costs discussed in this thesis.

E.1 Joint Cost (Direct Substitution)

Minimizing the joint cost function with the observed-error form of the dual EKF requires defining the instantaneous cost as:

$$J_k = \log(2\pi\sigma_n^2) + \frac{(y_k - \hat{x}_k)^2}{\sigma_n^2} + \log(2\pi\sigma_v^2) + \frac{(\hat{x}_k - \hat{x}_k^-)^2}{\sigma_v^2} \tag{E.1}$$

$$= \log(2\pi\sigma_n^2) + \frac{e_k^2}{\sigma_n^2} + \log(2\pi\sigma_v^2) + \frac{\tilde{\hat{x}}_k^2}{\sigma_v^2} \tag{E.2}$$

where $e_k \triangleq (y_k - \hat{x}_k)$ and $\tilde{\hat{x}}_k \triangleq (\hat{x}_k - \hat{x}_k^-)$.

E.1.1 Weight Estimation

The gradient of J_k with respect to the weights is given by:

$$\nabla_{\mathbf{w}} J_k = \frac{2e_k}{\sigma_n^2}\nabla_{\mathbf{w}} e_k + \frac{2\tilde{\hat{x}}_k}{\sigma_v^2}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k \tag{E.3}$$

and the Hessian is:

$$\nabla_{\mathbf{w}}^2 J_k = \frac{2}{\sigma_n^2}\nabla_{\mathbf{w}} e_k \nabla_{\mathbf{w}} e_k^T + \frac{2}{\sigma_v^2}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k \nabla_{\mathbf{w}}\tilde{\hat{x}}_k^T \quad + o(2) \tag{E.4}$$

where $o(2)$ represents the terms with second-order derivatives with respect to $\mathbf{w}$. Such terms will necessarily be neglected by a first-order approximation to the Hessian. As suggested on page 73, the gradient and Hessian can be approximated by defining the observed-error measurement as:

$$\mathbf{e}_k = \begin{bmatrix} \sigma_n^{-1} e_k \\ \sigma_v^{-1} \tilde{\hat{x}}_k \end{bmatrix} \quad \text{with negative gradient} \quad \mathbf{H}_{o,k} = -\begin{bmatrix} \sigma_n^{-1} \nabla_{\mathbf{w}}^T e_k \\ \sigma_v^{-1} \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k \end{bmatrix} \tag{E.5}$$

so that $\mathbf{e}_k^T \mathbf{e}_k = J_k$, as required. Furthermore, letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$,

$$\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{e}_k = -2 \cdot [\sigma_n^{-1} \nabla_{\mathbf{w}} e_k \quad \sigma_v^{-1} \nabla_{\mathbf{w}} \tilde{\hat{x}}_k] \cdot \begin{bmatrix} \sigma_n^{-1} e_k \\ \sigma_v^{-1} \tilde{\hat{x}}_k \end{bmatrix} \tag{E.6}$$

$$= -\nabla_{\mathbf{w}} J_k \tag{E.7}$$

gives the negative instantaneous gradient. A first-order approximation to the instantaneous Hessian is given by:

$$\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k} = 2 \cdot [\sigma_n^{-1} \nabla_{\mathbf{w}} e_k \quad \sigma_v^{-1} \nabla_{\mathbf{w}} \tilde{\hat{x}}_k] \cdot \begin{bmatrix} \sigma_n^{-1} \nabla_{\mathbf{w}}^T e_k \\ \sigma_v^{-1} \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k \end{bmatrix} \tag{E.8}$$

$$= 2\sigma_n^{-2} \nabla_{\mathbf{w}} e_k \nabla_{\mathbf{w}} e_k^T + 2\sigma_v^{-2} \nabla_{\mathbf{w}} \tilde{\hat{x}} \nabla_{\mathbf{w}} \tilde{\hat{x}}^T \tag{E.9}$$

$$\approx \nabla_{\mathbf{w}}^2 J_k. \tag{E.10}$$

E.1.2 Variance Estimation

The gradient of J_k with respect to the variance of either the measurement or process noise is given by:

$$\begin{aligned} \frac{\partial J_k}{\partial \sigma^2} = & \frac{1}{\sigma_n^2} \frac{\partial \sigma_n^2}{\partial \sigma^2} + \Big[\frac{2e_k}{\sigma_n^2} \frac{\partial e_k}{\partial \sigma^2} - \frac{e_k^2}{(\sigma_n^2)^2} \frac{\partial \sigma_n^2}{\partial \sigma^2} \Big] \\ & + \frac{1}{\sigma_v^2} \frac{\partial \sigma_v^2}{\partial \sigma^2} + \Big[\frac{2\tilde{\hat{x}}_k}{\sigma_v^2} \frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} - \frac{\tilde{\hat{x}}_k^2}{(\sigma_v^2)^2} \frac{\partial \sigma_v^2}{\partial \sigma^2} \Big] \end{aligned} \tag{E.11}$$

and the second derivative is:

$$\begin{aligned} \frac{\partial^2 J_k}{(\partial \sigma^2)^2} = & -\frac{1}{(\sigma_n^2)^2} \Big(\frac{\partial \sigma_n^2}{\partial \sigma^2} \Big)^2 + \Big[\frac{2}{\sigma_n^2} \Big(\frac{\partial e_k}{\partial \sigma^2} \Big)^2 - \frac{4e_k}{(\sigma_n^2)^2} \Big(\frac{\partial e_k}{\partial \sigma^2} \frac{\partial \sigma_n^2}{\partial \sigma^2} \Big) + \frac{2e_k^2}{(\sigma_n^2)^3} \Big(\frac{\partial \sigma_n^2}{\partial \sigma^2} \Big)^2 \Big] \\ & - \frac{1}{(\sigma_v^2)^2} \Big(\frac{\partial \sigma_v^2}{\partial \sigma^2} \Big)^2 + \Big[\frac{2}{\sigma_v^2} \Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} \Big)^2 - \frac{4\tilde{\hat{x}}_k}{(\sigma_v^2)^2} \Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} \frac{\partial \sigma_v^2}{\partial \sigma^2} \Big) + \frac{2\tilde{\hat{x}}_k^2}{(\sigma_v^2)^3} \Big(\frac{\partial \sigma_v^2}{\partial \sigma^2} \Big)^2 \Big] + o(2) \end{aligned}$$

or, equivalently:

$$\begin{aligned} \frac{\partial^2 J_k}{(\partial \sigma^2)^2} = & \Big(\frac{2e_k^2}{(\sigma_n^2)^3} - \frac{1}{(\sigma_n^2)^2} \Big) \Big(\frac{\partial \sigma_n^2}{\partial \sigma^2} \Big)^2 + \frac{2}{\sigma_n^2} \Big(\frac{\partial e_k}{\partial \sigma^2} \Big)^2 - \frac{4e_k}{(\sigma_n^2)^2} \Big(\frac{\partial e_k}{\partial \sigma^2} \frac{\partial \sigma_n^2}{\partial \sigma^2} \Big) \\ & \Big(\frac{2\tilde{\hat{x}}_k^2}{(\sigma_v^2)^3} - \frac{1}{(\sigma_v^2)^2} \Big) \Big(\frac{\partial \sigma_v^2}{\partial \sigma^2} \Big)^2 + \frac{2}{\sigma_v^2} \Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} \Big)^2 - \frac{4\tilde{\hat{x}}_k}{(\sigma_v^2)^2} \Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} \frac{\partial \sigma_v^2}{\partial \sigma^2} \Big) + o(2). \end{aligned} \tag{E.12}$$

The observed-error vector is defined as:

$$\breve{\boldsymbol{\epsilon}}_k \triangleq \begin{bmatrix} (\ell_n)^{\frac{1}{2}} \\ \sigma_n^{-1} e_k \\ (\ell_v)^{\frac{1}{2}} \\ \sigma_v^{-1} \tilde{\hat{x}}_k \end{bmatrix}, \quad \text{so that} \quad \breve{\mathbf{H}}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_n)^{-\frac{1}{2}}}{\sigma_n^2}\frac{\partial \sigma_n^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_n}\frac{\partial e_k}{\partial \sigma^2} + \frac{e_k}{2(\sigma_n^2)^{(3/2)}}\frac{\partial \sigma_n^2}{\partial \sigma^2} \\ -\frac{1}{2}\frac{(\ell_v)^{-\frac{1}{2}}}{\sigma_v^2}\frac{\partial \sigma_v^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_v}\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} + \frac{\tilde{\hat{x}}_k}{2(\sigma_v^2)^{(3/2)}}\frac{\partial \sigma_v^2}{\partial \sigma^2} \end{bmatrix}. \tag{E.13}$$

Letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$, the exact negative gradient is given by $\breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\boldsymbol{\epsilon}}_k$. However, the first-order component of the second derivative is merely approximated by:

$$\begin{aligned} \breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{H}}_{o,k} =& \frac{\ell_n^{-1}}{2(\sigma_n^2)^2}\Big(\frac{\partial \sigma_n^2}{\partial \sigma^2}\Big)^2 + \frac{2}{\sigma_n^2}\Big(\frac{\partial e_k}{\partial \sigma^2}\Big)^2 - \frac{2e_k}{(\sigma_n^2)^2}\Big(\frac{\partial e_k}{\partial \sigma^2}\frac{\partial \sigma_n^2}{\partial \sigma^2}\Big) + \frac{e_k^2}{2(\sigma_n^2)^3}\Big(\frac{\partial \sigma_n^2}{\partial \sigma^2}\Big)^2 \\ &+ \frac{\ell_v^{-1}}{2(\sigma_v^2)^2}\Big(\frac{\partial \sigma_v^2}{\partial \sigma^2}\Big)^2 + \frac{2}{\sigma_v^2}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big)^2 - \frac{2\tilde{\hat{x}}_k}{(\sigma_v^2)^2}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\frac{\partial \sigma_v^2}{\partial \sigma^2}\Big) + \frac{\tilde{\hat{x}}_k^2}{2(\sigma_v^2)^3}\Big(\frac{\partial \sigma_v^2}{\partial \sigma^2}\Big)^2, \end{aligned}$$

or, gathering like terms:

$$\begin{aligned} \breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{H}}_{o,k} =& \Big(\frac{\ell_n^{-1}}{2(\sigma_n^2)^2} + \frac{e_k^2}{2(\sigma_n^2)^3}\Big)\Big(\frac{\partial \sigma_n^2}{\partial \sigma^2}\Big)^2 + \frac{2}{\sigma_n^2}\Big(\frac{\partial e_k}{\partial \sigma^2}\Big)^2 - \frac{2e_k}{(\sigma_n^2)^2}\Big(\frac{\partial e_k}{\partial \sigma^2}\frac{\partial \sigma_n^2}{\partial \sigma^2}\Big) \\ &+ \Big(\frac{\ell_v^{-1}}{2(\sigma_v^2)^2} + \frac{\tilde{\hat{x}}_k^2}{2(\sigma_v^2)^3}\Big)\Big(\frac{\partial \sigma_v^2}{\partial \sigma^2}\Big)^2 + \frac{2}{\sigma_v^2}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big)^2 - \frac{2\tilde{\hat{x}}_k}{(\sigma_v^2)^2}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\frac{\partial \sigma_v^2}{\partial \sigma^2}\Big). \end{aligned} \tag{E.14}$$

Comparing this expression to the one in Equation E.12 shows that the third and sixth coefficients are off by a factor of $\frac{1}{2}$. Although this cannot be remedied, the second derivative is most closely approximated by matching the first and fourth coefficients:

$$\Big(\frac{\ell_n^{-1}}{2(\sigma_n^2)^2} + \frac{e_k^2}{2(\sigma_n^2)^3}\Big) = \Big(\frac{2e_k^2}{(\sigma_n^2)^3} - \frac{1}{(\sigma_n^2)^2}\Big) \quad . \tag{E.15}$$

$$\text{and} \qquad \Big(\frac{\ell_v^{-1}}{2(\sigma_v^2)^2} + \frac{\tilde{\hat{x}}_k^2}{2(\sigma_v^2)^3}\Big) = \Big(\frac{2\tilde{\hat{x}}_k^2}{(\sigma_v^2)^3} - \frac{1}{(\sigma_v^2)^2}\Big). \tag{E.16}$$

These equalities are satisfied so long as ℓ_n and ℓ_v are redefined as the time-varying quantities:

$$\ell_{n,k} = \log(\alpha_k \cdot 2\pi\sigma_n^2) \qquad \ell_{v,k} = \log(\gamma_k \cdot 2\pi\sigma_v^2), \tag{E.17}$$

and α_k and γ_k are chosen to satisfied the conditions:

$$\ell_{n,k} = \frac{\sigma_n^2}{3e_k^2 - 2\sigma_n^2} \quad \text{and} \quad \ell_{v,k} = \frac{\sigma_v^2}{3\tilde{\hat{x}}_k^2 - 2\sigma_v^2} \tag{E.18}$$

for all time k. As described on page 76, this redefinition is equivalent to adding the offset $\log(\alpha_k) + \log(\gamma_k)$ to the cost J_k.

E.1.3 Colored Noise

When the measurement noise n_k is colored, the observed-error forms are the same as for the white noise case, except with the noise error, $\tilde{n}_k = (\hat{n}_k - \hat{n}_k^-)$, replacing $e_k = (y_k - \hat{x}_k)$, and colored noise innovations variance, $\sigma_{v_n}^2$, replacing σ_n^2.

E.2 Joint Cost (Error Coupled)

Sequential minimization of $J^{ec}(\mathbf{w})$ requires yet another form of the observed-error weight filter. Here the instantaneous error is:

$$J_k = \log(2\pi\sigma_{e_k}^2) + \frac{e_k^2}{\sigma_{e_k}^2} + \log(2\pi g_k) + \frac{(\tilde{\hat{x}}_k)^2}{g_k}, \tag{E.19}$$

E.2.1 Weight Estimation

The gradient of J_k with respect to the weights is:

$$\begin{aligned}\nabla_{\mathbf{w}} J_k = &\frac{1}{\sigma_{e_k}^2}\nabla_{\mathbf{w}}\sigma_{e_k}^2 - \frac{e_k^2}{(\sigma_{e_k}^2)^2}\nabla_{\mathbf{w}}\sigma_{e_k}^2 + \frac{2e_k}{\sigma_{e_k}^2}\nabla_{\mathbf{w}} e_k \\ &+ \frac{1}{g_k}\nabla_{\mathbf{w}} g_k - \frac{\tilde{\hat{x}}_k^2}{(g_k)^2}\nabla_{\mathbf{w}} g_k + \frac{2\tilde{\hat{x}}_k}{g_k}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k\end{aligned} \tag{E.20}$$

and the Hessian is given by the unwieldy expression:

$$\begin{aligned}\nabla_{\mathbf{w}}^2 J_k = &-\frac{1}{(\sigma_{e_k}^2)^2}\nabla_{\mathbf{w}}\sigma_{e_k}^2\nabla_{\mathbf{w}}^T\sigma_{e_k}^2 + \frac{2e_k^2}{(\sigma_{e_k}^2)^3}\nabla_{\mathbf{w}}\sigma_{e_k}^2\nabla_{\mathbf{w}}^T\sigma_{e_k}^2 - \frac{2e_k}{(\sigma_{e_k}^2)^2}\nabla_{\mathbf{w}}\sigma_{e_k}^2\nabla_{\mathbf{w}}^T e_k \\ &- \frac{2e_k}{(\sigma_{e_k}^2)^2}\nabla_{\mathbf{w}} e_k\nabla_{\mathbf{w}}^T\sigma_{e_k}^2 + \frac{2}{\sigma_{e_k}^2}\nabla_{\mathbf{w}} e_k\nabla_{\mathbf{w}} e_k^T \\ &- \frac{1}{(g_k)^2}\nabla_{\mathbf{w}} g_k\nabla_{\mathbf{w}}^T g_k + \frac{2\tilde{\hat{x}}_k^2}{(g_k)^3}\nabla_{\mathbf{w}} g_k\nabla_{\mathbf{w}}^T g_k - \frac{2\tilde{\hat{x}}_k}{(g_k)^2}\nabla_{\mathbf{w}} g_k\nabla_{\mathbf{w}}^T\tilde{\hat{x}}_k \\ &- \frac{2\tilde{\hat{x}}_k}{(g_k)^2}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k\nabla_{\mathbf{w}}^T g_k + \frac{2}{g_k}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k\nabla_{\mathbf{w}}\tilde{\hat{x}}_k^T \quad + o(2).\end{aligned} \tag{E.21}$$

Combining terms where possible gives:

$$\begin{aligned}\nabla_{\mathbf{w}}^2 J_k = &\Big(\frac{2e_k^2}{(\sigma_{e_k}^2)^3} - \frac{1}{(\sigma_{e_k}^2)^2}\Big)\nabla_{\mathbf{w}}\sigma_{e_k}^2\nabla_{\mathbf{w}}^T\sigma_{e_k}^2 \\ &- \frac{2e_k}{(\sigma_{e_k}^2)^2}\Big(\nabla_{\mathbf{w}}\sigma_{e_k}^2\nabla_{\mathbf{w}}^T e_k + \nabla_{\mathbf{w}} e_k\nabla_{\mathbf{w}}^T\sigma_{e_k}^2\Big) + \frac{2}{\sigma_{e_k}^2}\nabla_{\mathbf{w}} e_k\nabla_{\mathbf{w}} e_k^T \\ &+ \Big(\frac{2\tilde{\hat{x}}_k^2}{(g_k)^3} - \frac{1}{(g_k)^2}\Big)\nabla_{\mathbf{w}} g_k\nabla_{\mathbf{w}}^T g_k \\ &- \frac{2\tilde{\hat{x}}_k}{(g_k)^2}\Big(\nabla_{\mathbf{w}} g_k\nabla_{\mathbf{w}}^T\tilde{\hat{x}}_k + \nabla_{\mathbf{w}}\tilde{\hat{x}}_k\nabla_{\mathbf{w}}^T g_k\Big) + \frac{2}{g_k}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k\nabla_{\mathbf{w}}\tilde{\hat{x}}_k^T \quad + o(2).\end{aligned} \tag{E.22}$$

As before $o(2)$ is used to represent the terms containing second derivatives with respect to $\mathbf{w}$.

The gradient and Hessian of J_k are approximated by defining the observed-error term and its

negative gradient as:

$$\mathbf{e}_k = \begin{bmatrix} (\ell_{e,k})^{\frac{1}{2}} \\ \sigma_{e_k}^{-1} e_k \\ (\ell_{g,k})^{\frac{1}{2}} \\ g_k^{(-1/2)} \tilde{\hat{x}}_k \end{bmatrix}, \quad \text{and} \quad \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{2} \frac{(\ell_{e,k})^{-\frac{1}{2}}}{\sigma_{e_k}^2} \nabla_{\mathbf{w}}^T (\sigma_{e_k}^2) \\ -\frac{1}{\sigma_{e_k}} \nabla_{\mathbf{w}}^T e_k + \frac{e_k}{2(\sigma_{e_k}^2)^{(3/2)}} \nabla_{\mathbf{w}}^T \sigma_{e_k}^2 \\ -\frac{1}{2} \frac{(\ell_{g,k})^{-\frac{1}{2}}}{g_k} \nabla_{\mathbf{w}}^T (g_k) \\ -\frac{1}{g_k^{(1/2)}} \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k + \frac{\tilde{\hat{x}}_k}{2 g_k^{(3/2)}} \nabla_{\mathbf{w}}^T g_k \end{bmatrix}, \tag{E.23}$$

where $\ell_{e,k} = \log(2\pi\sigma_{e_k}^2)$ and $\ell_{g,k} = \log(2\pi g_k)$. This satisfies $\mathbf{e}_k^T \mathbf{e}_k = J_k$, and $\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{e}_k = -\nabla_{\mathbf{w}} J_k$ gives the negative gradient as expressed in Equation E.20.

The Hessian is approximated by:

$$\begin{aligned} \mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k} = & \frac{1}{2\ell_{e,k}(\sigma_{e_k}^2)^2} \nabla_{\mathbf{w}} \sigma_{e_k}^2 \nabla_{\mathbf{w}}^T \sigma_{e_k}^2 + \frac{2}{\sigma_{e_k}^2} \nabla_{\mathbf{w}} e_k \nabla_{\mathbf{w}}^T e_k \\ & - \frac{e_k}{(\sigma_{e_k}^2)^2} \Big(\nabla_{\mathbf{w}}^T \sigma_{e_k}^2 \nabla_{\mathbf{w}} e_k + \nabla_{\mathbf{w}} e_k \nabla_{\mathbf{w}}^T \sigma_{e_k}^2 \Big) + \frac{e_k^2}{2(\sigma_{e_k}^2)^3} \nabla_{\mathbf{w}} \sigma_{e_k}^2 \nabla_{\mathbf{w}}^T \sigma_{e_k}^2 \\ & + \frac{1}{2\ell_{e,k}(g_k)^2} \nabla_{\mathbf{w}} g_k \nabla_{\mathbf{w}}^T g_k + \frac{2}{g_k} \nabla_{\mathbf{w}} \tilde{\hat{x}}_k \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k \\ & - \frac{\tilde{\hat{x}}_k}{(g_k)^2} \Big(\nabla_{\mathbf{w}} g_k \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k + \nabla_{\mathbf{w}} \tilde{\hat{x}}_k \nabla_{\mathbf{w}}^T g_k \Big) + \frac{\tilde{\hat{x}}_k^2}{2(g_k)^3} \nabla_{\mathbf{w}} g_k \nabla_{\mathbf{w}}^T g_k, \end{aligned} \tag{E.24}$$

or by rearranging the terms:

$$\begin{aligned} \mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k} = & \Big(\frac{e_k^2}{2(\sigma_{e_k}^2)^3} + \frac{\ell_{e,k}^{-1}}{2(\sigma_{e_k}^2)^2} \Big) \nabla_{\mathbf{w}} \sigma_{e_k}^2 \nabla_{\mathbf{w}}^T \sigma_{e_k}^2 \\ & - \frac{e_k}{(\sigma_{e_k}^2)^2} \Big(\nabla_{\mathbf{w}} \sigma_{e_k}^2 \nabla_{\mathbf{w}}^T e_k + \nabla_{\mathbf{w}} e_k \nabla_{\mathbf{w}}^T \sigma_{e_k}^2 \Big) + \frac{2}{\sigma_{e_k}^2} \nabla_{\mathbf{w}} e_k \nabla_{\mathbf{w}}^T e_k \\ & + \Big(\frac{\tilde{\hat{x}}_k^2}{2(g_k)^3} + \frac{\ell_{g,k}^{-1}}{2(g_k)^2} \Big) \nabla_{\mathbf{w}} g_k \nabla_{\mathbf{w}}^T g_k \\ & - \frac{\tilde{\hat{x}}_k}{(g_k)^2} \Big(\nabla_{\mathbf{w}} g_k \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k + \nabla_{\mathbf{w}} \tilde{\hat{x}}_k \nabla_{\mathbf{w}}^T g_k \Big) + \frac{2}{g_k} \nabla_{\mathbf{w}} \tilde{\hat{x}}_k \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k. \end{aligned} \tag{E.25}$$

Comparing Equations E.22 and E.25 shows that $\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k}$ approximates the first-order part of the Hessian best when the coefficients of the first and fourth terms are matched (the coefficients of the second and fifth terms are unalterably off by a factor of $\frac{1}{2}$). This is accomplished by redefining $\ell_{e,k}$ and $\ell_{g,k}$ as the time-varying quantities:

$$\ell_{e,k} = \log(\alpha_k \cdot 2\pi\sigma_{e_k}^2) \qquad \ell_{g,k} = \log(\gamma_k \cdot 2\pi g_k) \quad , \tag{E.26}$$

where α_k and γ_k are chosen for each k such that:

$$\ell_{e,k} = \frac{\sigma_{e_k}^2}{3e_k^2 - 2\sigma_{e_k}^2} \qquad \ell_{g,k} = \frac{g}{3\tilde{\hat{x}}_k^2 - 2g} \quad , \tag{E.27}$$

as required.

E.2.2 Variance Estimation

The gradient of the error-coupled joint cost with respect to either of the noise variances is given by:

$$\begin{aligned}\frac{\partial J_k}{\partial \sigma^2} &= \frac{1}{\sigma^2_{e_k}}\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2} - \frac{e_k^2}{(\sigma^2_{e_k})^2}\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2} + \frac{2e_k}{\sigma^2_{e_k}}\frac{\partial e_k}{\partial \sigma^2} \\ &+ \frac{1}{g_k}\frac{\partial g_k}{\partial \sigma^2} - \frac{\tilde{\hat{x}}_k^2}{(g_k)^2}\frac{\partial g_k}{\partial \sigma^2} + \frac{2\tilde{\hat{x}}_k}{g_k}\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\end{aligned} \tag{E.28}$$

and the Hessian is given by the unwieldy expression:

$$\begin{aligned}\frac{\partial^2 J_k}{(\partial \sigma^2)^2} &= -\frac{1}{(\sigma^2_{e_k})^2}\Big(\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\Big)^2 + \frac{2e_k^2}{(\sigma^2_{e_k})^3}\Big(\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\Big)^2 - \frac{2e_k}{(\sigma^2_{e_k})^2}\Big(\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\frac{\partial e_k}{\partial \sigma^2}\Big) \\ &- \frac{2e_k}{(\sigma^2_{e_k})^2}\Big(\frac{\partial e_k}{\partial \sigma^2}\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\Big) + \frac{2}{\sigma^2_{e_k}}\Big(\frac{\partial e_k}{\partial \sigma^2}\Big)^2 \\ &- \frac{1}{(g_k)^2}\Big(\frac{\partial g_k}{\partial \sigma^2}\Big)^2 + \frac{2\tilde{\hat{x}}_k^2}{(g_k)^3}\Big(\frac{\partial g_k}{\partial \sigma^2}\Big)^2 - \frac{2\tilde{\hat{x}}_k}{(g_k)^2}\Big(\frac{\partial g_k}{\partial \sigma^2}\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big) \\ &- \frac{2\tilde{\hat{x}}_k}{(g_k)^2}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\frac{\partial g_k}{\partial \sigma^2}\Big) + \frac{2}{g_k}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big)^2 + o(2).\end{aligned} \tag{E.29}$$

Combining terms where possible gives:

$$\begin{aligned}\frac{\partial^2 J_k}{(\partial \sigma^2)^2} &= \Big(\frac{2e_k^2}{(\sigma^2_{e_k})^3} - \frac{1}{(\sigma^2_{e_k})^2}\Big)\Big(\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\Big)^2 - \frac{4e_k}{(\sigma^2_{e_k})^2}\Big(\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\frac{\partial e_k}{\partial \sigma^2}\Big) + \frac{2}{\sigma^2_{e_k}}\Big(\frac{\partial e_k}{\partial \sigma^2}\Big)^2 \\ &+ \Big(\frac{2\tilde{\hat{x}}_k^2}{(g_k)^3} - \frac{1}{(g_k)^2}\Big)\Big(\frac{\partial g_k}{\partial \sigma^2}\Big)^2 - \frac{4\tilde{\hat{x}}_k}{(g_k)^2}\Big(\frac{\partial g_k}{\partial \sigma^2}\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big) + \frac{2}{g_k}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big)^2 + o(2).\end{aligned} \tag{E.30}$$

As before, $o(2)$ is used to represent the terms containing second derivatives with respect to σ.

These derivatives are provided by the Kalman variance filter by defining the observed-error $\breve{\epsilon}_k$ the same as ϵ_k in Equation E.23, and computing $\breve{\mathbf{H}}_{o,k}$ as:

$$\breve{\mathbf{H}}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_{e,k})^{-\frac{1}{2}}}{\sigma^2_{e_k}}\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2} \\ -\frac{1}{\sigma_{e_k}}\frac{\partial e_k}{\partial \sigma^2} + \frac{e_k}{2(\sigma^2_{e_k})^{(3/2)}}\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2} \\ -\frac{1}{2}\frac{(\ell_{g,k})^{-\frac{1}{2}}}{g_k}\frac{\partial g_k}{\partial \sigma^2} \\ -\frac{1}{g_k^{(1/2)}}\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2} + \frac{\tilde{\hat{x}}_k}{2g_k^{(3/2)}}\frac{\partial g_k}{\partial \sigma^2} \end{bmatrix}. \tag{E.31}$$

Defining $\sigma_r^2 = \frac{1}{2}\mathbf{I}$ as before gives $\breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\epsilon}_k = -\frac{\partial J_k}{\partial \sigma^2}$. Similar to the weight estimation form, the second derivative is approximated by:

$$\begin{aligned}\breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{H}}_{o,k} &= \Big(\frac{e_k^2}{2(\sigma^2_{e_k})^3} + \frac{\ell_{e,k}^{-1}}{2(\sigma^2_{e_k})^2}\Big)\Big(\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\Big)^2 - \frac{2e_k}{(\sigma^2_{e_k})^2}\Big(\frac{\partial \sigma^2_{e_k}}{\partial \sigma^2}\frac{\partial e_k}{\partial \sigma^2}\Big) + \frac{2}{\sigma^2_{e_k}}\Big(\frac{\partial e_k}{\partial \sigma^2}\Big)^2 \\ &+ \Big(\frac{\tilde{\hat{x}}_k^2}{2(g_k)^3} + \frac{\ell_{g,k}^{-1}}{2(g_k)^2}\Big)\Big(\frac{\partial g_k}{\partial \sigma^2}\Big)^2 - \frac{2\tilde{\hat{x}}_k}{(g_k)^2}\Big(\frac{\partial g_k}{\partial \sigma^2}\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big) + \frac{2}{g_k}\Big(\frac{\partial \tilde{\hat{x}}_k}{\partial \sigma^2}\Big)^2.\end{aligned} \tag{E.32}$$

Again, the coefficients of the second and fourth terms differ by a factor of $\frac{1}{2}$ from the coefficients in the expression for the true second derivative in Equation E.30. However, the first and fourth coefficients can be made to match by forcing the identities:

$$\ell_{e,k} = \frac{\sigma_{e_k}^2}{3e_k^2 - 2\sigma_{e_k}^2} \qquad \ell_{g,k} = \frac{g}{3\tilde{\hat{x}}_k^2 - 2g} \quad , \tag{E.33}$$

as before.

E.2.3 Colored Noise

As with $J_c^j\mathbf{w}$, the colored noise forms of the observed-error for $J_c^{ec}\mathbf{w}$ are very similar to the white noise case. Again, $\tilde{\hat{n}}_k = (\hat{n}_k - \hat{n}_k^-)$, replaces e_k; the variance of e_k is replaced by the variance of $\tilde{\hat{n}}_k$: $g_{n,k}$.

E.3 Maximum-Likelihood Cost Function

The instantaneous cost for maximum-likelihood estimation is

$$J_k = \log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \hat{x}_k^-)^2}{\sigma_{\varepsilon_k}^2} \tag{E.34}$$

$$= \log(2\pi\sigma_{\varepsilon_k}^2) + \frac{\varepsilon_k^2}{\sigma_{\varepsilon_k}^2} \tag{E.35}$$

where $\varepsilon_k = (y_k - \hat{x}_k^-)$.

E.3.1 Weight Estimation

The gradient of J_k with respect to the weights is:

$$\nabla_{\mathbf{w}} J_k = \frac{1}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2 - \frac{\varepsilon_k^2}{(\sigma_{\varepsilon_k}^2)^2}\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2 + \frac{2\varepsilon_k}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}\varepsilon_k \tag{E.36}$$

The Hessian is given by:

$$\begin{aligned}\nabla_{\mathbf{w}}^2 J_k = &-\frac{1}{(\sigma_{\varepsilon_k}^2)^2}\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 + \frac{2\varepsilon_k^2}{(\sigma_{\varepsilon_k}^2)^3}\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 - \frac{2\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\varepsilon_k \\ &- \frac{2\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 + \frac{2}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}\varepsilon_k^T + o(2),\end{aligned} \tag{E.37}$$

$$\begin{aligned}\nabla_{\mathbf{w}}^2 J_k = &\left(\frac{2\varepsilon_k^2}{(\sigma_{\varepsilon_k}^2)^3} - \frac{1}{(\sigma_{\varepsilon_k}^2)^2}\right)\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 \\ &- \frac{2\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\left(\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\varepsilon_k + \nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2\right) + \frac{2}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}\varepsilon_k^T + o(2).\end{aligned} \tag{E.38}$$

These quantities are approximated by letting

$$\boldsymbol{\epsilon}_k \triangleq \begin{bmatrix} \sqrt{\ell_{\varepsilon,k}} \\ \sigma_{\varepsilon_k}^{-1}\varepsilon_k \end{bmatrix}, \quad \text{giving} \quad \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_{\varepsilon,k})^{-\frac{1}{2}}}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 \\ -\frac{1}{\sigma_{\epsilon_k}}\nabla_{\mathbf{w}}^T\varepsilon_k + \frac{\varepsilon_k}{2(\sigma_{\varepsilon_k}^2)^{(3/2)}}\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 \end{bmatrix} \tag{E.39}$$

and letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$. This gives $\check{\mathbf{H}}_{o,k}^T\sigma_r^{-2}\check{\boldsymbol{\epsilon}}_k = -\nabla_{\mathbf{w}}J_k$ as desired. Meanwhile,

$$\begin{aligned}\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{H}_{o,k} = &\frac{1}{2\ell_{\varepsilon,k}(\sigma_{\varepsilon_k}^2)^2}\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 + \frac{2}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}^T\varepsilon_k \\ &-\frac{\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\Big(\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}\varepsilon_k + \nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2\Big) + \frac{\varepsilon_k^2}{2(\sigma_{\varepsilon_k}^2)^3}\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2,\end{aligned} \tag{E.40}$$

or by rearranging the terms:

$$\begin{aligned}\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{H}_{o,k} = &\Big(\frac{\varepsilon_k^2}{2(\sigma_{\varepsilon_k}^2)^3} + \frac{\ell_{\varepsilon,k}^{-1}}{2(\sigma_{\varepsilon_k}^2)^2}\Big)\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2 \\ &-\frac{\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\Big(\nabla_{\mathbf{w}}\sigma_{\varepsilon_k}^2\nabla_{\mathbf{w}}^T\varepsilon_k + \nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}^T\sigma_{\varepsilon_k}^2\Big) + \frac{2}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}\varepsilon_k\nabla_{\mathbf{w}}^T\varepsilon_k.\end{aligned} \tag{E.41}$$

Comparing Equations E.38 and E.41 shows that $\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{H}_{o,k}$ approximates the first-order part of the Hessian best when the coefficients of the first and third terms are matched (the coefficients of the second term is unalterably off by a factor of $\frac{1}{2}$). This is accomplished by redefining $\ell_{\varepsilon,k}$ with the time-varying quantity:

$$\ell_{\varepsilon,k} = \log(\alpha_k \cdot 2\pi\sigma_{\varepsilon_k}^2) \tag{E.42}$$

where α_k is chosen for each k such that:

$$\ell_{\varepsilon,k} = \frac{\sigma_{\varepsilon_k}^2}{3\varepsilon_k^2 - 2\sigma_{\varepsilon_k}^2}. \tag{E.43}$$

E.3.2 Variance Estimation

The derivative of J_k with respect to the noise variances is:

$$\frac{\partial J_k}{\partial\sigma^2} = \frac{1}{\sigma_{\varepsilon_k}^2}\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2} - \frac{\varepsilon_k^2}{(\sigma_{\varepsilon_k}^2)^2}\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2} + \frac{2\varepsilon_k}{\sigma_{\varepsilon_k}^2}\frac{\partial\varepsilon_k}{\partial\sigma^2} \tag{E.44}$$

The Hessian is given by:

$$\begin{aligned}\frac{\partial^2 J_k}{(\partial\sigma^2)^2} = &-\frac{1}{(\sigma_{\varepsilon_k}^2)^2}\Big(\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2}\Big)^2 + \frac{2\varepsilon_k^2}{(\sigma_{\varepsilon_k}^2)^3}\Big(\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2}\Big)^2 - \frac{2\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\Big(\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2}\frac{\partial\varepsilon_k}{\partial\sigma^2}\Big) \\ &-\frac{2\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\frac{\partial\varepsilon_k}{\partial\sigma^2}\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2} + \frac{2}{\sigma_{\varepsilon_k}^2}\Big(\frac{\partial\varepsilon_k}{\partial\sigma^2}\Big)^2\end{aligned} \tag{E.45}$$

$$\frac{\partial^2 J_k}{(\partial\sigma^2)^2} = \Big(\frac{2\varepsilon_k^2}{(\sigma_{\varepsilon_k}^2)^3} - \frac{1}{(\sigma_{\varepsilon_k}^2)^2}\Big)\Big(\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2}\Big)^2 - \frac{4\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\Big(\frac{\partial\sigma_{\varepsilon_k}^2}{\partial\sigma^2}\frac{\partial\varepsilon_k}{\partial\sigma^2}\Big) + \frac{2}{\sigma_{\varepsilon_k}^2}\Big(\frac{\partial\varepsilon_k}{\partial\sigma^2}\Big)^2 \tag{E.46}$$

These quantities are approximated by letting

$$\mathbf{e}_k \triangleq \begin{bmatrix} \sqrt{\ell_{\varepsilon,k}} \\ \sigma_{\varepsilon_k}^{-1}\varepsilon_k \end{bmatrix}, \quad \text{giving} \quad \mathbf{H}_{o,k} = \begin{bmatrix} -\frac{1}{2}\frac{(\ell_{\varepsilon,k})^{-\frac{1}{2}}}{\sigma_{\varepsilon_k}^2}\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} \\ -\frac{1}{\sigma_{\varepsilon_k}}\frac{\partial \varepsilon_k}{\partial \sigma^2} + \frac{\varepsilon_k}{2(\sigma_{\varepsilon_k}^2)^{(3/2)}}\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} \end{bmatrix} \tag{E.47}$$

and letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$. This gives $\breve{\mathbf{H}}_{o,k}^T \sigma_r^{-2} \breve{\mathbf{e}}_k = -\frac{\partial J_k}{\partial \sigma^2}$ as desired. Meanwhile,

$$\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k} = \frac{1}{2\ell_{\varepsilon,k}(\sigma_{\varepsilon_k}^2)^2}\Big(\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2}\Big)^2 + \frac{2}{\sigma_{\varepsilon_k}^2}\Big(\frac{\partial \varepsilon_k}{\partial \sigma^2}\Big)^2 - \frac{2\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2}\frac{\partial \varepsilon_k}{\partial \sigma^2} + \frac{\varepsilon_k^2}{2(\sigma_{\varepsilon_k}^2)^3}\Big(\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2}\Big)^2 \tag{E.48}$$

or by rearranging the terms:

$$\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k} = \Big(\frac{\varepsilon_k^2}{2(\sigma_{\varepsilon_k}^2)^3} + \frac{\ell_{\varepsilon,k}^{-1}}{2(\sigma_{\varepsilon_k}^2)^2}\Big)\Big(\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2}\Big)^2 - \frac{2\varepsilon_k}{(\sigma_{\varepsilon_k}^2)^2}\frac{\partial \varepsilon_k}{\partial \sigma^2}\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} + \frac{2}{\sigma_{\varepsilon_k}^2}\Big(\frac{\partial \varepsilon_k}{\partial \sigma^2}\Big)^2 \tag{E.49}$$

Comparing Equations E.45 and E.49 shows that $\mathbf{H}_{o,k}^T \sigma_r^{-2} \mathbf{H}_{o,k}$ approximates the first-order part of the Hessian best when the coefficients of the first and third terms are matched (the coefficients of the second and fifth terms are unalterably off by a factor of $\frac{1}{2}$). As with weight estimation, this is accomplished with Equation E.43.

E.4 EM Cost Function

The sequential EM cost is derived in Appendix F to be:

$$\begin{aligned} J^{em} = N\log(2\pi^2\sigma_n^2) + N\log(2\pi^2\sigma_v^2) + \sum_{k=1}^{N}\Bigg(\frac{(y_k - \hat{x}_k)^2 + p_{k|k}}{\sigma_n^2} \\ + \frac{(\hat{x}_k - \hat{x}_{k|k}^-)^2 + p_{k|k} - 2p_{k|k}^\dagger + p_{k|k}^-}{\sigma_v^2}\Bigg), \end{aligned} \tag{E.50}$$

where only the predictions $\hat{x}_{k|k}^-$ and covariances $p_{k|k}^\dagger$ and $p_{k|k}^-$ can be considered functions of the weights.

E.4.1 Weight Estimation

The gradient of the instantaneous cost J_k with respect to the weights is:

$$\nabla_{\mathbf{w}} J_k = \frac{2\tilde{\hat{x}}_{k|k}\nabla_{\mathbf{w}}\tilde{\hat{x}}_{k|k} - 2\nabla_{\mathbf{w}}p_{k|k}^\dagger + \nabla_{\mathbf{w}}p_{k|k}^-}{\sigma_v^2}, \tag{E.51}$$

and the Hessian is given by:

$$\nabla_{\mathbf{w}}^2 J_k = \frac{2\nabla_{\mathbf{w}}\tilde{\hat{x}}_{k|k}\nabla_{\mathbf{w}}^T\tilde{\hat{x}}_{k|k} - 2\nabla_{\mathbf{w}}^2 p_{k|k}^\dagger + \nabla_{\mathbf{w}}^2 p_{k|k}^- + 2\tilde{\hat{x}}_{k|k}\nabla_{\mathbf{w}}^2\tilde{\hat{x}}_{k|k}}{\sigma_v^2}. \tag{E.52}$$

These quantities are approximated by letting

$$\mathbf{e}_k \triangleq \begin{bmatrix} \sigma_v^{-1}\tilde{\hat{x}}_k \\ \sigma_v^{-1}\sqrt{-2p^{\dagger}_{k|k}} \\ \sigma_v^{-1}\sqrt{p^{-}_{k|k}} \end{bmatrix}, \quad \text{giving} \quad \mathbf{H}_{o,k} = -\begin{bmatrix} \sigma_v^{-1}\nabla_{\mathbf{w}}\tilde{\hat{x}}_k \\ \frac{\sqrt{-2}}{2\sigma_v\sqrt{p^{\dagger}_{k|k}}}\nabla_{\mathbf{w}}p^{\dagger}_{k|k} \\ \frac{1}{2\sigma_v\sqrt{p^{-}_{k|k}}}\nabla_{\mathbf{w}}p^{-}_{k|k} \end{bmatrix} \tag{E.53}$$

Using $\sigma_r^2 = \frac{1}{2}\mathbf{I}$ gives $\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{e}_k = -\nabla_{\mathbf{w}}J_k$ as desired, and

$$\mathbf{H}_{o,k}^T\sigma_r^{-2}\mathbf{H}_{o,k} = \frac{1}{\sigma_v^2}\left(2\nabla_{\mathbf{w}}\tilde{\hat{x}}_{k|k}\nabla_{\mathbf{w}}^T\tilde{\hat{x}}_{k|k} - \frac{\nabla_{\mathbf{w}}p^{\dagger}_{k|k}\nabla_{\mathbf{w}}^T p^{\dagger}_{k|k}}{p^{\dagger}_{k|k}} + \frac{\nabla_{\mathbf{w}}p^{-}_{k|k}\nabla_{\mathbf{w}}^T p^{-}_{k|k}}{p^{-}_{k|k}}\right).$$

The first term is the first-order part of the Hessian in Equation E.52. The last two terms can be dropped by simply letting $p^{-}_{k|k} = p^{\dagger}_{k|k}$ approach ∞ (use a very large number) in the definitions of $\mathbf{e}_k$ and $\mathbf{H}_{o,k}$ above. The effect of this redefinition cancels out of the gradient computation, and gives the desired approximation to the Hessian.

E.4.2 Variance Estimation

The derivative of J_k with respect to either of the variances is:

$$\frac{\partial J_k}{\partial \sigma^2} = \frac{1}{\sigma^2} - \frac{num_k}{(\sigma^2)^2} \tag{E.54}$$

and the second derivative is:

$$\frac{\partial^2 J_k}{(\partial\sigma^2)^2} = -\frac{1}{(\sigma^2)^2} + \frac{2num_k}{(\sigma^2)^3}\sigma_v^2, \tag{E.55}$$

where num_k is the appropriate numerator term:

$$num_k \triangleq \begin{cases} (y_k - \hat{x}_k)^2 + p_{k|k} & \text{when } \sigma^2 = \sigma_n^2, \\ (\hat{x}_k - \hat{x}^{-}_{k|k})^2 + p_{k|k} - 2p^{\dagger}_{k|k} + p^{-}_{k|k} & \text{when } \sigma^2 = \sigma_v^2. \end{cases} \tag{E.56}$$

These quantities are approximated by letting

$$\breve{\mathbf{e}}_k \triangleq \begin{bmatrix} \sqrt{\ell_k} \\ \sigma^{-1}\sqrt{num_k} \end{bmatrix}, \quad \text{with derivative} \quad \breve{\mathbf{H}}_{o,k} = -\begin{bmatrix} \frac{1}{2\sigma^2\sqrt{\ell_k}} \\ -\frac{\sqrt{num_k}}{2(\sigma^2)^{3/2}} \end{bmatrix}. \tag{E.57}$$

Letting $\sigma_r^2 = \frac{1}{2}\mathbf{I}$ gives $\breve{\mathbf{H}}_{o,k}^T\sigma_r^{-2}\breve{\mathbf{e}}_k = -\nabla_{\mathbf{w}}J_k$ as desired. Meanwhile,

$$\breve{\mathbf{H}}_{o,k}^T\sigma_r^{-2}\breve{\mathbf{H}}_{o,k} = \frac{1}{2\ell_k(\sigma^2)^2} + \frac{num_k}{2(\sigma^2)^3}\Bigg) \tag{E.58}$$

Forcing $\ell_k = -2$ gives the exact second derivative, scaled by a factor of $\frac{1}{4}$. This scaling factor is easily fixed by using $\frac{1}{2}\breve{\mathbf{e}}_k$ and $2\mathbf{H}_{o,k}$ in the variance filter.

E.4.3 Colored Noise

For colored measurement noise, the sequential EM cost is:

$$J_c^{em} = N\log(2\pi^2\sigma_v^2) + N\log(2\pi^2\sigma_{v_n}^2) + \sum_{k=1}^{N}\left(\frac{(\hat{x}_k - \hat{x}_{k|k}^-)^2 + p_{k|k} - 2p_{k|k}^\dagger + p_{k|k}^-}{\sigma_v^2} + \frac{(\hat{n}_k - \hat{n}_{k|k}^-)^2 + p_{n,k|k} - 2p_{n,k|k}^\dagger + p_{n,k|k}^-}{\sigma_{v_n}^2}\right). \tag{E.59}$$

However, the derivatives for the weight and variance filters are computed exactly as in the white noise case, with the exception that the numerator term num_k is now defined as

$$num_k \triangleq (\hat{n}_k - \hat{n}_{k|k}^-)^2 + p_{n,k|k} - 2p_{n,k|k}^\dagger + p_{n,k|k}^- \tag{E.60}$$

when $\sigma^2 = \sigma_{v_n}^2$.

Appendix F

EM Cost Function

The expectation-maximization (EM) algorithm is useful in many different settings. This appendix derives the EM cost function in the context of the dual estimation problem. The off-line problem is considered here, wherein all data up to time N is available. The development for the linear white noise case closely follows that given by Shumway and Stoffer in [76], but is restricted to one-dimensional measurements, y_k.

F.1 Batch EM

The expectation-maximization cost function is given in Equation 2.55 on page 37 as:

$$J^{em} = E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}\left[N\log(4\pi^2\sigma_v^2\sigma_n^2) + \sum_{k=1}^{N}\left(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - x_k^-)^2}{\sigma_v^2}\right) \;\middle|\; \{y_t\}_1^N, \hat{\mathbf{w}}\right].$$

From here forward the conditioning arguments in the expectation are implied, but not shown. Moving the expectation inside the sum and expanding the quadratics gives:

$$\begin{aligned} J^{em} &= N\log(4\pi^2\sigma_v^2\sigma_n^2) + \sum_{k=1}^{N} E\left[\frac{y_k^2 - 2y_k x_k + x_k^2}{\sigma_n^2} + \frac{x_k^2 - 2x_k x_k^- + (x_k^-)^2}{\sigma_v^2}\right], \\ &= N\log(4\pi^2\sigma_v^2\sigma_n^2) + \sum_{k=1}^{N}\left(\frac{y_k^2 - 2y_k E[x_k] + E[x_k^2]}{\sigma_n^2} + \frac{E[x_k^2] - 2E[x_k x_k^-] + E[(x_k^-)^2]}{\sigma_v^2}\right). \end{aligned} \quad \text{(F.1)}$$

Furthermore, defining

$$\hat{x}_{k|N} \triangleq E[x_k|\{y_t\}_1^N, \hat{\mathbf{w}}], \quad \text{(F.2)}$$

$$\hat{x}_{k|N}^- \triangleq E[x_k^-|\{y_t\}_1^N, \hat{\mathbf{w}}] \approx f(\hat{x}_{k-1|N}, \ldots, \hat{x}_{k-M|N}, \mathbf{w}), \quad \text{(F.3)}$$

$$p_{k|N} \triangleq var[x_k] = E[x_k^2|\{y_t\}_1^N, \hat{\mathbf{w}}] - \hat{x}_{k|N}^2 \quad \text{(F.4)}$$

$$p_{k|N}^- \triangleq var[x_k^-|\{y_t\}_1^N, \hat{\mathbf{w}}] = E[(x_k^-)^2|\{y_t\}_1^N, \hat{\mathbf{w}}] - (\hat{x}_{k|N}^-)^2 \quad \text{(F.5)}$$

$$p_{k|N}^\dagger \triangleq var[x_k, x_k^-|\{y_t\}_1^N, \hat{\mathbf{w}}] = E[x_k x_k^-|\{y_t\}_1^N, \hat{\mathbf{w}}] - \hat{x}_{k|N}\hat{x}_{k|N}^- \quad \text{(F.6)}$$

allows the cost to be rewritten as:

$$J^{em} = N\log(4\pi^2\sigma_v^2\sigma_n^2) + \sum_{k=1}^{N}\Bigg(\frac{y_k^2 - 2y_k\hat{x}_{k|N} + \hat{x}_{k|N}^2 + p_{k|N}}{\sigma_n^2} + \frac{\hat{x}_{k|N}^2 + p_{k|N} - 2\hat{x}_{k|N}\hat{x}_{k|N}^- - 2p_{k|N}^\dagger + (\hat{x}_{k|N}^-)^2 + p_{k|N}^-}{\sigma_v^2}\Bigg) \tag{F.7}$$

$$= N\log(4\pi^2\sigma_v^2\sigma_n^2) + \sum_{k=1}^{N}\Bigg(\frac{(y_k - \hat{x}_{k|N})^2 + p_{k|N}}{\sigma_n^2} + \frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 + p_{k|N} - 2p_{k|N}^\dagger + p_{k|N}^-}{\sigma_v^2}\Bigg) \tag{F.8}$$

Note that $\hat{x}_{k|N} = \mathbf{C}\hat{\mathbf{x}}_{k|N}$, and $p_{k|N} = \mathbf{C}\mathbf{P}_{k|N}\mathbf{C}^T$.

F.1.1 Linear Case

Furthermore, if the model is linear, then $\hat{x}_{k|N}^- = \mathbf{w}^T\hat{\mathbf{x}}_{k-1|N}$, and:

$$p_{k|N}^\dagger = E[(x_k - \hat{x}_k)(x_k^- - \hat{x}_k^-)|\{y_t\}_1^N, \hat{\mathbf{w}}] \tag{F.9a}$$

$$= E[(x_k - \hat{x}_k)(\mathbf{w}^T\mathbf{x}_{k-1} - \mathbf{w}^T\hat{\mathbf{x}}_{k-1})|\{y_t\}_1^N, \hat{\mathbf{w}}] \tag{F.9b}$$

$$= \mathbf{C}E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^T|\{y_t\}_1^N, \hat{\mathbf{w}}]\cdot\mathbf{w}^T \tag{F.9c}$$

$$= \mathbf{C}(\mathbf{P}_{k|N}^\#)\mathbf{w}, \tag{F.9}$$

$$p_{k|N}^- = E[(x_k^- - \hat{x}_k^-)^2|\{y_t\}_1^N, \hat{\mathbf{w}}] \tag{F.10a}$$

$$= E[(\mathbf{w}^T\mathbf{x}_{k-1} - \mathbf{w}^T\hat{\mathbf{x}}_{k-1})^2|\{y_t\}_1^N, \hat{\mathbf{w}}] \tag{F.10b}$$

$$= \mathbf{w}^T\cdot E[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^2|\{y_t\}_1^N, \hat{\mathbf{w}}]\cdot\mathbf{w} \tag{F.10c}$$

$$= \mathbf{w}^T(\mathbf{P}_{k-1|N})\mathbf{w}. \tag{F.10}$$

The quantities $\hat{\mathbf{x}}_{k|N}$, $\hat{\mathbf{x}}_{k-1|N}$, $\mathbf{P}_{k|N}$, $\mathbf{P}_{k-1|N}$, $\mathbf{P}_{k|N}^\#$ can all be generated from the current weight estimates $\hat{\mathbf{w}}$ by a Kalman smoother [68], modified slightly as in [76] to produce $\mathbf{P}_{k|N}^\#$. These values are no longer interpreted as functions of the unknown parameters $\mathbf{w}$, σ_v^2, and σ_n^2. In fact, of all the terms in the cost function, only $\hat{x}_{k|N}^-$, $p_{k|N}^-$, and $p_{k|N}^\dagger$ are functions of $\mathbf{w}$. The weights can therefore be estimated by minimizing

$$J^{em}(\mathbf{w}) = \sum_{k=1}^{N}\Bigg(\frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 - 2p_{k|N}^\dagger + p_{k|N}^-}{\sigma_v^2}\Bigg). \tag{F.11}$$

Similarly, all the terms in the numerator in Equation 2.55 are dependent on the previous variance estimates, $\hat{\sigma}_n^2$ and $\hat{\sigma}_n^2$, rather than on the variance being estimated. Hence, σ_v^2 is estimated using the partial cost:

$$J^{em}(\sigma_v^2) = N\log(2\pi\sigma_v^2) + \sum_{k=1}^{N}\Bigg(\frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 + p_{k|N} - 2p_{k|N}^\dagger + p_{k|N}^-}{\sigma_v^2}\Bigg), \tag{F.12}$$

while the portion relevant to estimating σ_n^2 is:

$$J^{em}(\sigma_n^2) = N\log(2\pi\sigma_n^2) + \sum_{k=1}^{N}\left(\frac{(y_k - \hat{x}_{k|N})^2 + p_{k|N}}{\sigma_n^2}\right). \tag{F.13}$$

Closed form solutions for $\mathbf{w}$, σ_v^2, and σ_n^2 are derived as in [76] by setting the gradients of the costs to zero. For weight estimation, this gives:

$$0 = \sum_{k=1}^{N}\left(\frac{-2\hat{\mathbf{x}}_{k-1|N}(\hat{x}_{k|N} - \mathbf{w}^T\hat{\mathbf{x}}_{k-1|N}) - 2(\mathbf{P}_{k|N}^{\#})^T\mathbf{C}^T + 2(\mathbf{P}_{k-1|N})\mathbf{w}}{\sigma_v^2}\right) \tag{F.14}$$

$$= \sum_{k=1}^{N}\left(\hat{\mathbf{x}}_{k-1|N}\hat{\mathbf{x}}_{k-1|N}^T\mathbf{w} + (\mathbf{P}_{k-1|N})\mathbf{w}\right) - \sum_{k=1}^{N}\left(\hat{\mathbf{x}}_{k-1|N}\hat{x}_{k|N} + (\mathbf{P}_{k|N}^{\#})^T\mathbf{C}^T\right). \tag{F.15}$$

Solving for $\mathbf{w}$ gives:

$$\hat{\mathbf{w}} = \left\{\sum_{k=1}^{N}\left(\hat{\mathbf{x}}_{k-1|N}\hat{\mathbf{x}}_{k-1|N}^T + (\mathbf{P}_{k-1|N})\right)\right\}^{-1} \cdot \sum_{k=1}^{N}\left(\hat{\mathbf{x}}_{k-1|N}\hat{\mathbf{x}}_{k|N}^T + (\mathbf{P}_{k|N}^{\#})^T\right)\mathbf{C}^T. \tag{F.16}$$

Similarly, the process noise variance can be estimated by taking the derivative of the cost $J^{em}(\sigma_v^2)$ with respect to σ_v^2, and setting it to zero:

$$0 = \frac{N}{\sigma_v^2} - \sum_{k=1}^{N}\frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 + p_{k|N} - 2p_{k|N}^{\dagger} + p_{k|N}^-}{(\sigma_v^2)^2} \tag{F.17}$$

$$\Rightarrow \hat{\sigma_v^2} = \frac{1}{N}\sum_{k=1}^{N}\left((\hat{x}_{k|N} - \hat{x}_{k|N}^-)^2 + p_{k|N} - 2p_{k|N}^{\dagger} + p_{k|N}^-\right). \tag{F.18}$$

The measurement noise variance can be estimated by setting the derivative of its cost with respect to σ_n^2 to zero:

$$0 = \frac{N}{\sigma_n^2} - \sum_{k=1}^{N}\left(\frac{(y_k - \hat{x}_{k|N})^2 + p_{k|N}}{(\sigma_n^2)^2}\right) \tag{F.19}$$

$$\Rightarrow \hat{\sigma_n^2} = \frac{1}{N}\sum_{k=1}^{N}\left((y_k - \hat{x}_{k|N})^2 + p_{k|N}\right). \tag{F.20}$$

Equations F.16, F.18, and F.20 are known as the *M-step* of the algorithm. The new estimates of the parameters are then used in the Kalman smoother (or *E-step*), which is followed by another M-step, and so on.

F.1.2 Nonlinear Case

As mentioned in Chapter 3, nonlinear systems require a generalized M-step[1], and the E-step is often performed with an extended Kalman smoother (EKS). In this case, the prediction term

[1]The variance estimation can still be performed in closed form.

of Equation F.3 is approximated as $\hat{x}^-_{k|N} = f(\hat{\mathbf{x}}_{k|N}, \mathbf{w})$, and the variances $p^-_{k|N}$ and $p^\dagger_{k|N}$ are approximated as

$$p^-_{k|N} = E[(x^-_k - \hat{x}^-_k)^2 | \{y_t\}_1^k, \mathbf{w}] \tag{F.21a}$$

$$= E[(f(\mathbf{x}_{k-1}, \mathbf{w}) - f(\hat{\mathbf{x}}_{k-1}, \mathbf{w}))^2 | \{y_t\}_1^k, \mathbf{w}] \tag{F.21b}$$

$$\approx \nabla_{\mathbf{x}}^T f \cdot E[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^2 | \{y_t\}_1^k, \mathbf{w}] \cdot \nabla_{\mathbf{x}} f \tag{F.21c}$$

$$= \mathbf{C}\mathbf{A}_{k|N}(\mathbf{P}_{k-1|N})\mathbf{A}^T_{k|N}\mathbf{C}^T \tag{F.21}$$

where the covariance $\mathbf{P}_{k-1|N}$ is itself approximated by the EKS. The transition matrix $\mathbf{A}_{k|N}$ is found by linearizing $f(\cdot)$ at $\hat{\mathbf{x}}_{k-1|N}$. Similarly, the cross-variance $p^\dagger_{k|N}$ is approximated as:

$$p^\dagger_{k|N} = E[(x_k - \hat{x}_k)(x^-_k - \hat{x}^-_k) | \{y_t\}_1^k, \mathbf{w}] \tag{F.22a}$$

$$= E[(x_k - \hat{x}_k)(f(\mathbf{x}_{k-1}, \mathbf{w}) - f(\hat{\mathbf{x}}_{k-1}, \mathbf{w})) | \{y_t\}_1^k, \mathbf{w}] \tag{F.22b}$$

$$\approx \mathbf{C}E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^T | \{y_t\}_1^k, \mathbf{w}] \cdot \nabla_{\mathbf{x}} f \tag{F.22c}$$

$$= \mathbf{C}(\mathbf{P}^\#_k)\mathbf{A}^T_{k|N}\mathbf{C}^T, \tag{F.22}$$

where the covariance $\mathbf{P}^\#_k$ is approximated by a modified EKS.

F.2 Colored Noise EM Cost

As given in Equation 2.58 on page 37, the colored-noise EM cost is:

$$J_c^{em}(\mathbf{w}) = E_{\mathbf{x}_1^N \mathbf{n}_1^N | \mathbf{y}_1^N \mathbf{w}} \left[\sum_{k=1}^{N} \left(\log(2\pi\sigma_v^2) + \frac{(x_k - x^-_k)^2}{\sigma_v^2} + \log(2\pi\sigma^2_{v_n}) + \frac{(n_k - n^-_k)^2}{\sigma^2_{v_n}} \right) \Bigg| \{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n \right],$$

As before, the conditioning arguments are suppressed for brevity in the following development. Moving the expectation inside the sum and expanding the quadratic terms gives:

$$J_c^{em}(\mathbf{w}) = N \log(4\pi^2\sigma_v^2\sigma^2_{v_n}) + \sum_{k=1}^{N} E\left[\frac{x_k^2 - 2x_k x^-_k + (x^-_k)^2}{\sigma_v^2} + \frac{n_k^2 - 2n_k n^-_k + (n^-_k)^2}{\sigma^2_{v_n}} \right], \tag{F.23}$$

$$= N \log(4\pi^2\sigma_v^2\sigma^2_{v_n}) + \sum_{k=1}^{N} \left(\frac{E[x_k^2] - 2E[x_k x^-_k] + E[(x^-_k)^2]}{\sigma_v^2} + \frac{E[n_k^2] - 2E[n_k n^-_k] + E[(n^-_k)^2]}{\sigma^2_{v_n}} \right), \tag{F.24}$$

Using the definitions:

$$\begin{aligned}
\hat{n}_{k|N} &\triangleq E[n_k|\mathbf{w}, \mathbf{w}_n, \{y_t\}_1^N] \\
\hat{n}^-_{k|N} &\triangleq E[n^-_k|\hat{\mathbf{w}}, \mathbf{w}_n, \{y_t\}_1^N] \quad = \mathbf{w}_n^T \hat{\mathbf{n}}_{k-1} \\
p_{n,k|N} &\triangleq var[n_k|\hat{\mathbf{w}}, \mathbf{w}_n, \{y_t\}_1^N] \quad = E[n_k^2|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] - \hat{n}^2_{k|N} \\
p^-_{n,k|N} &\triangleq var[n^-_k|\hat{\mathbf{w}}, \mathbf{w}_n, \{y_t\}_1^N] \quad = E[(n^-_k)^2|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] - (\hat{n}^-_{k|N})^2 \\
p^\dagger_{n,k|N} &\triangleq var[n_k n^-_k|\hat{\mathbf{w}}, \mathbf{w}_n, \{y_t\}_1^N] \quad = E[n_k n^-_k|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] - \hat{n}_{k|N}\hat{n}^-_{k|N}
\end{aligned}$$

and the corresponding terms defined for the signal x_k in Equations F.2-F.6, allows the cost to be written as:

$$\begin{aligned}
J_c^{em}(\mathbf{w}, \sigma_v^2, \sigma_{v_n}^2) =& N \log(4\pi^2 \sigma_v^2 \sigma_{v_n}^2) \\
&+ \sum_{k=1}^{N} \Bigg(\frac{(\hat{x}_{k|N} - \hat{x}^-_{k|N})^2 + p_{k|N} - 2p^\dagger_{k|N} + p^-_{k|N}}{\sigma_v^2} \\
&+ \frac{(\hat{n}_{k|N} - \hat{n}^-_{k|N})^2 + p_{n,k|N} - 2p^\dagger_{n,k|N} + p^-_{n,k|N}}{\sigma_{v_n}^2} \Bigg).
\end{aligned} \tag{F.25}$$

For linear models, the closed form solutions for $\mathbf{w}$ and σ_v^2 that minimize this cost are the same as given in Equations F.16 and F.18; for nonlinear models, the generalized M-step for $\mathbf{w}$ is also unchanged. Meanwhile, the variance of the process noise driving the colored measurement noise is found as:

$$\hat{\sigma}^2_{v_n} = \frac{1}{N} \sum_{k=1}^{N} \Big((\hat{n}_{k|N} - \hat{n}^-_{k|N})^2 + p_{n,k|N} - 2p^\dagger_{n,k|N} + p^-_{n,k|N} \Big). \tag{F.26}$$

The noise model is assumed linear, so $\hat{n}^-_{k|N} = \mathbf{w}_n^T \hat{\mathbf{n}}_{k-1|N}$. Furthermore, note that $p_{n,k|N} = \mathbf{C}\mathbf{P}_{n,k|N}\mathbf{C}^T$, where $\mathbf{P}_{n,k|N} = cov[\mathbf{x}|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n]$ and:

$$\begin{aligned}
p^\dagger_{n,k|N} &= E[(n_k - \hat{n}_k)(n^-_k - \hat{n}^-_k)|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] && \text{(F.27a)} \\
&= E[(n_k - \hat{n}_k)(\mathbf{w}_n^T \mathbf{n}_{k-1} - \mathbf{w}_n^T \hat{\mathbf{n}}_{k-1})|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] && \text{(F.27b)} \\
&= \mathbf{C} E[(\mathbf{n}_k - \hat{\mathbf{n}}_k)(\mathbf{n}_{k-1} - \hat{\mathbf{n}}_{k-1})^T|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] \cdot \mathbf{w}_n^T && \text{(F.27c)} \\
&= \mathbf{C}(\mathbf{P}^{\#}_{n,k|N})\mathbf{w}_n, && \text{(F.27)}
\end{aligned}$$

$$\begin{aligned}
p^-_{k|N} &= E[(n^-_k - \hat{n}^-_k)^2|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] && \text{(F.28a)} \\
&= E[(\mathbf{w}_n^T \mathbf{n}_{k-1} - \mathbf{w}_n^T \hat{\mathbf{n}}_{k-1})^2|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] && \text{(F.28b)} \\
&= \mathbf{w}_n^T \cdot E[(\mathbf{n}_{k-1} - \hat{\mathbf{n}}_{k-1})^2|\{y_t\}_1^N, \hat{\mathbf{w}}, \mathbf{w}_n] \cdot \mathbf{w}_n && \text{(F.28c)} \\
&= \mathbf{w}_n^T (\mathbf{P}_{n,k-1|N}) \mathbf{w}_n. && \text{(F.28)}
\end{aligned}$$

The quantities $\hat{\mathbf{n}}_{k-1|N}$, $\mathbf{P}_{n,k|N}$, $\mathbf{P}_{n,k-1|N}$, $\mathbf{P}^{\#}_{n,k|N}$ can all be generated from the current weight estimates $\hat{\mathbf{w}}$ and $\mathbf{w}_n$ by a Kalman smoother, modified to accomodate colored noise, and to produce $\mathbf{P}^{\#}_{n,k|N}$.

Appendix G

Errors-in-Variables

Errors-in-variables (EIV) models are sometimes used to handle regression problems wherein the regressors are measured with error [75]. Because the autoregressive models used in this thesis for time-series analysis are a special kind of regression problem, a strong relationship exists between the EIV framework and the dual estimation methods developed in this thesis.

Consider the batch problem of estimating $\mathbf{w}$ and $\mathbf{x}_1^N$ given the vector of noisy observations $\mathbf{y}_1^N$. The EIV model makes a distinction between input and output variables, and between the deterministic and stochastic parts of the input-output relationship. In the context of noisy time-series, the input and output data are the same thing. The "input data" are

$$y_k = x_k + n_k, \tag{G.1}$$

and the "output data" are:

$$y_k = f(\mathbf{x}_{k-1}, \mathbf{w}) + v_k + n_k. \tag{G.2}$$

Hence, the input data contain errors $(y_k - x_k)$, with variance σ_n^2, and the output data contain errors $(y_k - f(\mathbf{x}_{k-1}, \mathbf{w}))$, with variance $\sigma_v^2 + \sigma_n^2$. Furthermore, the cross-covariance between the two errors is σ_n^2. These errors can be concatenated in the column vector:

$$\mathbf{E} \triangleq \begin{bmatrix} \mathbf{e} \\ \boldsymbol{\varepsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1^N - \mathbf{x}_1^N \\ \mathbf{y}_1^N - (\mathbf{x}^-)_1^N \end{bmatrix}, \tag{G.3}$$

where $(\mathbf{x}^-)_1^N$ is a column vector with elements: $x_k^- = f(\mathbf{x}_{k-1}, \mathbf{w})$. The maximum likelihood estimates of $\mathbf{x}_1^N$ and $\mathbf{w}$ are found by maximizing the log-likelihood of $\mathbf{E}$, or minimizing the cost function:

$$J(\mathbf{x}_1^k, \mathbf{w}) = \mathbf{E}^T \boldsymbol{\Sigma}^{-1} \mathbf{E}, \tag{G.4}$$

where $\boldsymbol{\Sigma}^{-1}$ is the covariance of $\mathbf{E}$. The EIV method consists of iteratively:

1. minimizing $J(\mathbf{x}_1^k, \mathbf{w})$ with respect to $\mathbf{x}_1^N$, with $\mathbf{w}$ fixed at the current estimate: $\hat{\mathbf{w}}$.

2. minimizing $J(\mathbf{x}_1^k, \mathbf{w})$ with respect to $\mathbf{w}$, with $\mathbf{x}_1^N$ fixed at the current estimate: $\hat{\mathbf{x}}_1^N$.

When the model is linear, these steps can be solved in closed form with a batch weighted least-squares type of solution.

The relationship of the EIV cost to the joint cost explored in Chapter 2 can be seen by expanding $\mathbf{\Sigma}$ in Equation G.4 as:

$$J(\mathbf{x}_1^N, \mathbf{w}) = \begin{bmatrix} \mathbf{e} \\ \boldsymbol{\varepsilon} \end{bmatrix}^T \cdot \begin{bmatrix} \sigma_n^2 \mathbf{I} & \sigma_n^2 \mathbf{I} \\ \sigma_n^2 \mathbf{I} & (\sigma_v^2 + \sigma_n^2)\mathbf{I} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{e} \\ \boldsymbol{\varepsilon} \end{bmatrix} \tag{G.5}$$

$$= \begin{bmatrix} \mathbf{e} \\ \boldsymbol{\varepsilon} \end{bmatrix}^T \cdot \begin{bmatrix} (\frac{1}{\sigma_n^2} + \frac{1}{\sigma_v^2})\mathbf{I} & -\frac{1}{\sigma_v^2}\mathbf{I} \\ -\frac{1}{\sigma_v^2}\mathbf{I} & \frac{1}{\sigma_v^2}\mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{e} \\ \boldsymbol{\varepsilon} \end{bmatrix} \tag{G.6}$$

Letting the components of $\mathbf{e}$ and $\boldsymbol{\varepsilon}$ be denoted by $e_k = (y_k - x_k)$ and $\varepsilon_t = (y_k - x_k^-)$, respectively, the cost can be written in component form as:

$$J(\mathbf{x}_1^N, \mathbf{w}) = \sum_{t=1}^{N} \left(\frac{e_k^2}{\sigma_n^2} + \frac{e_k^2}{\sigma_v^2} + \frac{(\varepsilon_k)^2}{\sigma_v^2} + \frac{-2\varepsilon_k e_k}{\sigma_v^2} \right) \tag{G.7}$$

$$= \sum_{t=1}^{N} \left(\frac{e_k^2}{\sigma_n^2} + \frac{(\varepsilon_k - e_k)^2}{\sigma_v^2} \right). \tag{G.8}$$

However, note that the term $(\varepsilon_k - e_k)$ is identical to $(x_k - x_k^-)$, so the EIV cost is identical to the joint cost given in Equation 2.11 on page 23:

$$J^j(\mathbf{x}_1^N, \mathbf{w}) = \sum_{t=1}^{N} \left(\frac{(y_k - x_k)^2}{\sigma_n^2} + \frac{(x_k - x_k^-)^2}{\sigma_v^2} \right). \tag{G.9}$$

Appendix H

Measurement Noise Variance Upper Bound

On page 105, a procedure is described for estimating σ_n^2 by starting at an upper bound, and decrementing the estimate. In this appendix, the upper bound for the measurement noise variance is derived. Recall the formulation of a noncausal mapping from $2M+1$ noisy data points $\mathbf{y}_{k-M}^{k+M}$ to an estimate of the x_k:

$$\hat{x}_k = \mathbf{w}^T \mathbf{y}_{k-M}^{k+M}. \tag{H.1}$$

Consider weight estimates of the form:

$$\hat{\mathbf{w}} = \mathbf{R}_{yy}^{-1}(\mathbf{r}_{yy} - \hat{\sigma}_n^2 \mathbf{e}_0), \tag{H.2}$$

where $\hat{\mathbf{w}} = \hat{\mathbf{w}}^*$ is the optimal weight vector when $\hat{\sigma}_n^2 = \sigma_n^2$. Then the variance of the estimate can be expressed in terms of $\hat{\sigma}_n^2$:

$$var(\hat{x}_k) = E[\hat{\mathbf{w}}^T \mathbf{y}_{k-M}^{k+M} \cdot \mathbf{y}_{k-M}^{k+M} \hat{\mathbf{w}}] \tag{H.3}$$

$$= \hat{\mathbf{w}}^T \hat{\mathbf{R}}_{yy} \hat{\mathbf{w}} \tag{H.4}$$

$$= (\mathbf{r}_{yy} - \hat{\sigma}_n^2 \mathbf{e}_0)\mathbf{R}_{yy}^{-1}\mathbf{R}_{yy}\mathbf{R}_{yy}^{-1}(\mathbf{r}_{yy} - \hat{\sigma}_n^2 \mathbf{e}_0), \tag{H.5}$$

$$= (\mathbf{r}_{yy} - \hat{\sigma}_n^2 \mathbf{e}_0)\mathbf{R}_{yy}^{-1}(\mathbf{r}_{yy} - \hat{\sigma}_n^2 \mathbf{e}_0). \tag{H.6}$$

To find $\hat{\sigma}_n^2$ such that $\hat{x}_k$ is the minimum variance estimate, set the derivative of the variance (with respect to $\hat{\sigma}_n^2$) to zero:

$$\frac{\partial var(\hat{x}_k)}{\partial \sigma_n^2} = 2(\mathbf{r}_{yy} - \hat{\sigma}_n^2 \mathbf{e}_0)\mathbf{R}_{yy}^{-1}(-\mathbf{e}_0) = 0 \tag{H.7}$$

$$\Rightarrow \hat{\sigma}_n^2 \mathbf{e}_0 \mathbf{R}_{yy}^{-1} \mathbf{e}_0 = \mathbf{r}_{yy}\mathbf{R}_{yy}^{-1}\mathbf{e}_0 \tag{H.8}$$

$$\Rightarrow \hat{\sigma}_n^2 (\mathbf{R}_{yy}^{-1})^{(0,0)} = \mathbf{e}_0^T \mathbf{e}_0 \tag{H.9}$$

$$\Rightarrow \hat{\sigma}_n^2 = \frac{1}{(\mathbf{R}_{yy}^{-1})^{(0,0)}} \tag{H.10}$$

To show that this is an upper bound on the true variance σ_n^2, consider the MSE of the optimal estimator:

$$\text{MSE} = E[(x_k - \hat{\mathbf{w}}^{*T}\mathbf{y}_{k-M}^{k+M})^2] \tag{H.11}$$

$$= \sigma_x^2 - 2\hat{\mathbf{w}}^{*T}\mathbf{r}_{yx} + \hat{\mathbf{w}}^{*T}\mathbf{R}_{yy}^{-1}\hat{\mathbf{w}}^{*} \tag{H.12}$$

$$= \sigma_x^2 - 2(\mathbf{r}_{yy} - \sigma_n^2\mathbf{e}_0)^T\mathbf{R}_{yy}^{-1}\mathbf{r}_{yx} + (\mathbf{r}_{yy} - \sigma_n^2\mathbf{e}_0)^T\mathbf{R}_{yy}^{-1}(\mathbf{r}_{yy} - \sigma_n^2\mathbf{e}_0) \tag{H.13}$$

$$= \sigma_x^2 - 2\mathbf{r}_{yy}^T\mathbf{R}_{yy}^{-1}\mathbf{r}_{yx} + 2\sigma_n^2\mathbf{e}_0^T\mathbf{R}_{yy}^{-1}\mathbf{r}_{yx} + \mathbf{r}_{yy}^T\mathbf{R}_{yy}^{-1}\mathbf{r}_{yy} - 2\mathbf{r}_{yy}^T\mathbf{R}_{yy}^{-1}\sigma_n^2\mathbf{e}_0 + (\sigma_n^2)^2\mathbf{e}_0^T\mathbf{R}_{yy}^{-1}\mathbf{e}_0, \tag{H.14}$$

where $\mathbf{r}_{yy}^T\mathbf{R}_{yy}^{-1} = \mathbf{e}_0^T$ gives:

$$= \sigma_x^2 - 2\mathbf{e}_0^T\mathbf{r}_{yx} + 2\sigma_n^2\mathbf{e}_0^T\mathbf{R}_{yy}^{-1}\mathbf{r}_{yx} + \mathbf{e}_0^T\mathbf{r}_{yy} - 2\mathbf{e}_0^T\sigma_n^2\mathbf{e}_0 + (\sigma_n^2)^2\mathbf{e}_0^T\mathbf{R}_{yy}^{-1}\mathbf{e}_0 \tag{H.15}$$

$$= \sigma_x^2 - 2\sigma_x^2 + 2\sigma_n^2\mathbf{e}_0^T\mathbf{R}_{yy}^{-1}(\mathbf{r}_{yy} - \sigma_n^2\mathbf{e}_0) + \sigma_y^2 - 2\sigma_n^2 + (\sigma_n^2)^2(\mathbf{R}_{yy}^{-1})^{(0,0)} \tag{H.16}$$

$$= \sigma_n^2 + 2\sigma_n^2 - 2(\sigma_n^2)^2(\mathbf{R}_{yy}^{-1})^{(0,0)} - 2\sigma_n^2 + (\sigma_n^2)^2(\mathbf{R}_{yy}^{-1})^{(0,0)} \tag{H.17}$$

$$= \sigma_n^2 - (\sigma_n^2)^2(\mathbf{R}_{yy}^{-1})^{(0,0)} \tag{H.18}$$

$$= \sigma_n^2(1 - \sigma_n^2(\mathbf{R}_{yy}^{-1})^{(0,0)}). \tag{H.19}$$

Now, of course the MSE must be non-negative, which means that:

$$\sigma_n^2(\mathbf{R}_{yy}^{-1})^{(0,0)} \leq 1 \tag{H.20}$$

$$\Rightarrow \sigma_n^2 \leq \frac{1}{(\mathbf{R}_{yy}^{-1})^{(0,0)}}, \tag{H.21}$$

giving the desired upper bound on σ_n^2. When $\mathbf{R}$ is not known, it can be replaced by an estimate, $\hat{\mathbf{R}}$.

Appendix I

T Test

Section 4.2.2 describes the problem of determining whether two algorithmic treatments, $(a \neq b) \in A$, produce losses $L_a(\mathbf{x}_1^N, \mathbf{w}, \mathbf{y}_1^N)$ and $L_b(\mathbf{x}_1^N, \mathbf{w}, \mathbf{y}_1^N)$ which are significantly different. The t test – a common method for determining statistical significance [70] – is described in this Appendix for that purpose.

Letting $\mu_a \triangleq E_{\mathbf{Y}}[L_a]$ be the expectation of the loss over all noise realizations, the treatment-specific loss can be rewritten as $L_a = \mu_a + \varepsilon_a$, where ε_a represents a zero-mean random disturbance due to the specific realization of the data.

Given R repetitions of the measurement noise, the loss of the a^{th} treatment on the r^{th} repetition of the data is denoted by $L_a^{[r]}$, where $r \in \{1, 2, \ldots, R\}$. Each $L_a^{[r]}$ can be thought of as a sample from the distribution on L_a. The sample mean of L_a can be computed as $\frac{1}{R}\sum_{r=1}^{R} L_a^{[r]}$, and used to evaluate a. When comparing two treatments, however, the significance of the difference in their sample means must be considered.

Note however, that because the same R repetitions of the data $(\mathbf{y}_1^N)^{[r]}$ (and initial parameters $\hat{\mathbf{w}}_0$, $\hat{\sigma}_{v,0}^2$, $\hat{\sigma}_{n,0}^2$) are used across all methods, the samples for any two treatments are not independent. In fact, taken for any two methods at a time, the results constitute *paired samples* [70], because the conditions contributing to, for example, $L_a^{[r]}$ and $L_b^{[r]}$, differ only in the treatment used.

On the other hand, defining the difference $d_{a,b} \triangleq (L_a - L_b)$ creates a random variable that is sampled independently, by $d_{a,b}^{[r]} \triangleq (L_a^{[r]} - L_b^{[r]})$. Assuming a Gaussian distribution on the difference, with mean $\mu_{a,b} = \mu_a - \mu_b$ and variance $\sigma_{a,b}^2$, the significance can be tested by determining how likely it is that the distribution of $d_{a,b}$ has zero mean. This can be done by way of a *t test* [70].

Computing the sample average $D_{a,b} = \frac{1}{R}\sum_r d_{a,b}^{[r]}$, and normalizing as:

$$m = \frac{D_{a,b} - \mu_{a,b}}{\sigma_{a,b} \cdot R^{-\frac{1}{2}}} \tag{I.1}$$

produces a random variable, $\mathbf{m}$, with normal distribution. Furthermore, the sample variance,

$s_D^2 = \frac{1}{R-1}\sum_r (d_{a,b}^{[r]} - D_{a,b})^2$, scaled as:

$$v = \frac{(R-1)s_D^2}{\sigma_{a,b}^2}, \tag{I.2}$$

produces a chi-squared random variable, $\mathbf{v}$, with $R-1$ degrees of freedom.

Given two independent random variables $\mathbf{m} \sim \mathcal{N}(0,1)$ and $\mathbf{v} \sim \chi_n^2$, the random variable $\mathbf{t} = \mathbf{m}/\sqrt{\mathbf{v}n^{-1}}$ is distributed with the *t distribution* with n degrees of freedom [70]. Therefore, the statistic

$$t_{a,b} = \frac{D_{a,b} - 0}{s_D \cdot R^{-\frac{1}{2}}} \tag{I.3}$$

will have the t distribution with $R-1$ degrees of freedom if and only if $\mu_{a,b} = 0$. This condition is referred to as the *null hypothesis*, H_0; it represents the case of no difference between algorithmic treatments a and b.

Therefore, the probability that $\mu_{a,b} = 0$ when $|t_{a,b}| \geq t_1$ for some positive value of t_1 , is given by:

$$\alpha \triangleq Pr(\text{“}H_0 \text{ is true”} \ \& \ |t_{a,b}| \geq t_1) = 1 - \int_{-t_1}^{t_1} \rho_{t_{R-1}} \cdot dT, \tag{I.4}$$

where $\rho_{t_{R-1}}$ is the pdf of the t distribution with $R-1$ degrees of freedom. The smallest possible value of α occurs when t_1 is chosen to equal $|t_{a,b}|$; this minimum value of α is known as the p-value of $t_{a,b}$. When the p-value is close to zero, the probability of $\mu_{a,b} = 0$ is low, indicating a significant difference between treatments a and b. A large p-value, on the other hand, indicates that there is not sufficient evidence for differentiating between a and b.

Biographical Note

Alex Tremain Nelson was born in Salt Lake City, Utah on June 1, 1971. His family moved to Berkeley, California, where he attended Berkeley High School, rowed on the varsity crew team, and attained the rank of Eagle Scout. In 1993, he earned an Sc.B. degree in Engineering from Brown University with honors, *magna cum laude*. He was awarded an M.S. degree in Electrical and Computer Engineering from the University of California at San Diego, in 1995. Other awards and honors include:

- First place presentation in Annual Student Paper Competition, Oregon Graduate Institute ECE Department, 2000.
- Recipient of University of California Regents Fellowship, 1993-94.
- Recipient of Technical Analysis Corporation President's Award, 1993.
- Member of Sigma Xi, Tau Beta Pi, and the IEEE Signal Processing Society.

His interests include neural, adaptive, and machine-learning approaches to signal processing, image processing, and control. In particular, he is interested in the extraction of information from complex data streams, such as audio, video, and biomedical signals, as well as the estimation of system parameters for automatic control applications.

Alex Nelson is an author of the following publications:

- Dual Estimation and the Unscented Transformation. E. Wan, R. van der Merwe and A. Nelson. In *Advances in Neural Information Processing Systems: NIPS 1999*. 2000.
- Networks for Speech Enhancement. E. Wan and A. Nelson. In *Handbook of Neural Networks for Speech Processing*, Edited by Shigeru Katagiri, Artech House, Boston, USA. 1999, (in press).
- Removal of Noise from Speech Using the Dual EKF Algorithm. E. Wan and A. Nelson. In *Proceedings of the IEEE International Conference on Acoustics and Signal Processing: ICASSP'98*, 1998.

- A Two-Observation Kalman Framework for Maximum-Likelihood Modeling of Noisy Time Series. A. Nelson and E. Wan. In *Proceedings of the IEEE International Joint Conference on Neural Networks: IJCNN'98*, 1998.

- Neural Dual Extended Kalman Filtering: Applications in Speech Enhancement and Monaural Blind Signal Separation. E. Wan and A. Nelson. In *Neural Networks for Signal Processing VII: Proceedings of the 1997 IEEE Workshop*, ed. Principe, Morgan, Giles, Wilson, 1997.

- Neural Speech Enhancement Using Dual Extended Kalman Filtering. A. Nelson and E. Wan, In *Proceedings of the IEEE International Conference on Neural Networks: ICNN'97*, 1997.

- Dual Kalman Filtering Methods for Nonlinear Prediction, Smoothing, and Estimation. E. Wan and A. Nelson In *Advances in Neural Information Processing Systems: Proceedings of the 1996 Conference, NIPS-9*, ed. Mozer, Jordan, and Petsche, 1997.