DESIGN OF FINITE MEMORY FILTERS

by

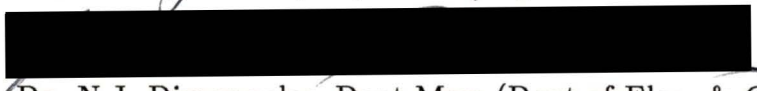
Hao Xu

B.Eng., Shanghai Second Polytechnic University, Shanghai, 1983

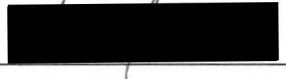
A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the Department
of
Electrical & Computer Engineering

We accept this thesis as conforming
to the required standard

Dr. P. Agathoklis, Supervisor(Dept of Elec. & Comp. Eng.)



Dr. N.J. Dimopoulos, Dept Mem.(Dept of Elec. & Comp. Eng.)



Dr. R. Illner, Outside Member(Department of Mathematics)

Dr. A.R. Sourour, External Examiner(Dept of Mathematics)

Supervisor: Dr. P. Agathoklis

Abstract

This thesis deals with the design of finite memory filters for both the one and the two dimensional cases.

Finite memory filters have many applications in real time control, estimation and identification, particularly in cases where the information about the system dynamics and the noise statistics is not precisely known. A recursive finite memory filter is developed to fit an nth order polynomial to equally spaced measurements. The parameters of the filter are determined by applying a weighted least-squares performance index, and therefore the stability of the recursive finite memory filter is guaranteed. The implementations of the nonrecursive and recursive finite memory filters are obtained and the performance of both filters is compared.

The performance of the finite memory filter is compared with the one of discrete Kalman filter. It is shown that the finite memory filter requires considerably less computations than the Kalman filter. The Kalman filter showed a better performance in the presence of white Gaussian noise while in the presence of non-gaussian noise both filter performed similarly.

A 2-D finite memory filter is developed by a second order bivariant polynomial to a set of 2-D measurements. A weighted least-squares performance index is used to ensure stability and the performance of the recursive and nonrecursive implementation is compared.

Examiners:

[redacted]

Dr. P. Agathoklis, Supervisor(Dept of Elec. & Comp. Eng.)

[redacted]

Dr. N.J. Dimopoulos, Dept Mem.(Dept of Elec. & Comp. Eng.)

[redacted]

Dr. R. Illner, Outside Member(Department of Mathematics)

[redacted]

Dr. A.R. Sourour, External Examiner(Dept of Mathematics)

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my appreciation to my supervisor Dr. Panajotis Agathoklis for his guidance during the course of this work and for his valuable advice in the preparation of this manuscript.

The assistance from faculty and staff in the Department of Electrical & Computer Engineering is also appreciated.

Financial support in the form of a University of Victoria Fellowship during 1986/1987 is gratefully acknowledged.

To My Parents

Chapter 1

Introduction

1.1 Review of Literature

A large class of estimation problems is concerned with finding an optimal estimation of some quantity (an unknown parameter, a random variable, or a random signal) based on measurements of a linear function of this quantity corrupted by additive noise. This estimation problem can be considered as a special case of the general approximation problem which can be roughly stated as the approximation of an unknown quantity from a combination of known quantities. This problem can be formulated either deterministically or probabilistically. Modern estimation methods use known relationships to compute the desired information from the measurements, taking into account measurement errors, prior knowledge of the information, the effects of disturbances and control actions on the system. Diverse measurements can be blended to form "best" estimates, and information which is unavailable for measurement can be approximated in an optimal fashion. Filtering is the process of extracting information from data which may only infer the desired information and which may contain errors.

The pioneer of a "theory" of estimation in which attempts are made to minimize various functions of the errors can be contributed to Galileo Galilei

in 1632 [1]. Then came a whole series of illustrious investigators, including Roger Cotes, Euler, Lagrange, Laplace, Bernoulli, and others [2].

In 1809, the German mathematician K. F. Gauss invented the technique of deterministic least-squares and employed it in a relatively simple orbit measurement problem [3]. He defined the least-squares estimation in the following manner: [3] " ...the most probable value of the unknown quantities will be that in which the sum of the squares of the difference between the actually observed and the computed values multiplied by numbers that measure the degree of precision is minimum." Since then, the various aspects of the least-squares method have been considered extensively in the literature [4]. The contribution by Fisher [5], Wiener [6] and Kolmogorov [7] are of particular importance.

Fisher's contribution [5] was the introduction of the maximum likelihood approach in connection with the probability density function. Contrary to minimizing the function of the difference between estimate and observation, Fisher proposed that the maximum of the probability density function is determined by maximizing the logarithm of this function. Therefore, the relationship between the least-squares method and probability theory was established. Utilizing random process theory, Wiener [6] proposed a procedure for the frequency domain design of statistically optimal filters. This technique addressed the continuous time problem in terms of correlation functions and the continuous filter impulse response and it was limited to statistically stationary processes and provided optimal estimates only in the steady-state. In the same time period, Kolmogorov [7] gave a comprehensive treatment of the prediction problem for discrete-time stationary processes.

With the stimulation of the increased usage of the digital computer in 1950's, the idea of generating least-squares estimation recursively was introduced [8]. This work is generally considered to have sparked the widespread

interest in the subject and the subsequent development of "Kalman filtering".

One important milestone in the area of linear filtering is Kalman's papers [9,10]. He provided an alternative way of formulating the least squares filtering problem using state-space methods. The two main features of the Kalman filters are (1) vector modeling of the random processes under consideration and (2) recursive processing of the noisy measurement data. Further, Kalman filter equations provide an extremely convenient procedure for digital computer implementation. Different from other filters, such as Wiener filter, Kalman filters are very general, can be applied to both stationary and nonstationary processes and can include the initial conditions of processes in estimation, prediction and filtering. The practicality of the Kalman approach to the estimation problem has become immensely popular in aerospace applications such as navigation and guidance.

The filtering theory developed by Kalman assumes that the system dynamics are completely known and precisely modeled in the filter and that the noise is Gaussian. However, there are many applications in control, guidance and marketing where the process parameters or the statistics of the noise are not precisely known. In these cases estimation algorithms, which require the dynamic model or the noise statistics, may diverge. Such applications include gun fire control [15] and radar signal detecting [16] where the noise has a Rayleigh distribution. A way to avoid divergence is to limit the memory of the filter, which curtails the growth of the actual estimation error covariance matrix. A brief survey on limited-memory filtering can be found in [11]. The development of recursive filters with limited memory, was addressed by several authors, most notably Schweppe [12] and Jazwinski [13]. Their efforts concentrated on a particular case of interest, namely, the case of discrete signals with state-space models having no driving noise. The more complete work accomplished by Bruckstein and Kailath [14] provided the solution to a

general linear state-space model with driving noise for both continuous and discrete time signals.

A special class of limited-memory filters are the polynomial finite memory filters. These filters have a finite impulse response and, hence, are suitable for such applications where the process and/or the noise characteristics are unknown or too complex. The recursive algorithms for finite memory filters of first, second and third order were first introduced by Nesline and Zarchan [15] for the gun fire control. However, these filters have the disadvantage of possibly becoming unstable in a finite wordlength implementation. This problem was resolved by Agathoklis and Hamza [17] by introducing a weighting factor β which gives a recursive finite memory filter with stability margin $|1 - \beta^{-1}|$. This technique was extended to 2-dimensional systems by Agathoklis and Foda [19] and the resulting 2-D finite memory filters were applied to image processing.

1.2 Outline of the Thesis

This thesis is organized as follows:

In Chapter 2, a brief review of the results of linear estimation is presented. The least-squares, the recursive least-squares and Kalman filtering techniques are introduced and briefly discussed. Two alternative forms of the Kalman filtering are given and their particular features are also briefly compared. Finally, the condition between least-squares estimation and Kalman filtering is discussed.

In Chapter 3, a generalized recursive algorithm is developed for an nth order finite memory filter. The parameters of the filter are determined by applying a weighted least-squares performance index, and therefore the stability of the finite memory filter is guaranteed. The implementations of the nonre-

cursive and recursive finite memory filters are obtained and the performance of both filters is discussed. This work has been published in [21].

In Chapter 4, Kalman filter and finite memory filter are applied to a simple problem. The parameters of a first order polynomial are estimated using Kalman and finite memory filters and the results are compared. The criteria used are (1) computation time (2) relative square error between the estimated value and the real value from t_0 to t_N. The computation requirements for both methods are obtained and the system responses to the ramp signal corrupted with noise are compared. Two cases are considered. In the first, the noise is Gaussian while in the second it has a Rayleigh distribution.

In Chapter 5, a second order 2-dimensional finite memory filter is designed. The original 2-D function is approximated at each step by a second order surface. The corresponding equations for the nonrecursive and recursive finite memory filters are derived using a performance index which includes a weighting factor. The weighting factor ensures the stability of the recursive implementation of the 2-D finite memory filter. Further, the noise performance and the frequency response of the filter are studied. This work has been published in [20].

In Chapter 6, a summary of the topics covered in this thesis and suggestions for future work are presented.

Chapter 2

Linear Estimation

2.1 Introduction

A large class of estimation problems is concerned with finding an optimal estimate of some quantity (an unknown parameter, a random variable, or a random signal) based on a measurements of a linear function of this quantity corrupted by an additive noise. One possible approach to this problem is deterministic or probabilistic least-squares estimation. The deterministic case is principally concerned with the approximation problems and the finite memory filter is an example of this approach. The probabilistic case deals mainly with estimation problems and the Kalman filter is related to solving this type of problems.

In this chapter, a brief review of some of the linear estimation technique is given. In Section 2.2, the deterministic least-squares estimation is presented while in Section 2.3 the Kalman filter approach is reviewed. Finally, the relationship between deterministic least-squares estimation and Kalman filter is discussed in Section 2.4.

2.2 Deterministic Least-Squares Estimation

2.2.1 Least-Squares Technique

Consider a set of measurements, $\mathbf{z}_i$, which are linearly dependent on the unknown quantities $\mathbf{x}_i$ by the following expression

$$\mathbf{z}_i = \mathbf{H}_i\mathbf{x} + \mathbf{v}_i \tag{2.1}$$

where $\mathbf{x}$ is an n-dimensional vector, $\mathbf{H}_i$ is a measurement matrix, $\mathbf{v}_i$ is a vector of measurement noise and the size of the vector $\mathbf{z}$ may be different at each measurement [22]. It is assumed that there are k measurements available, which can be symbolically represented by

$$z_k = H_k\mathbf{x} + v_k \tag{2.2}$$

z_k is the m-dimensional composite vector of all the observations which can be written in partitioned form as

$$z_k = \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \\ \vdots \\ \mathbf{z_k} \end{bmatrix} \tag{2.3}$$

H_k and v_k are a $m \times n$ matrix and a $m \times 1$ matrix, respectively, defined by

$$H_k = \begin{bmatrix} \mathbf{H_1} \\ \mathbf{H_2} \\ \vdots \\ \mathbf{H_k} \end{bmatrix} \tag{2.4}$$

$$v_k = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \vdots \\ \mathbf{v_k} \end{bmatrix} \tag{2.5}$$

The problem is, then, to select an estimate $\hat{\mathbf{x}}_k$ of $\mathbf{x}$ such that the quadratic measure

$$\mathbf{J} = (z_k - H_k\hat{\mathbf{x}}_k)^T R_k^{-1}(z_k - H_k\hat{\mathbf{x}}_k) \tag{2.6}$$

is minimized, where R_k^{-1} is $m \times m$ symmetric, positive definite weighting matrix. The resulting least-squares estimate, $\hat{x}_{k_{LS}}$, is found by setting

$$\frac{\partial \mathbf{J}(\hat{\mathbf{x}}_k)}{\partial \hat{\mathbf{x}}_k} \mid_{\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k_{LS}}} = 0 \tag{2.7}$$

which yields

$$\hat{x}_{k_{LS}} = (H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} z_k \tag{2.8}$$

Define

$$\mathbf{P}_k = (H_k^T R_k^{-1} H_k)^{-1} \tag{2.9}$$

then $\hat{\mathbf{x}}_{k_{LS}}$ can be expressed as

$$\hat{\mathbf{x}}_{k_{LS}} = \mathbf{P}_k H_k^T R_k^{-1} z_k \tag{2.10}$$

Note that these results have no direct probability interpretation since they were derived based on a deterministic argument only. Consequently, the least-squares estimates may be preferred to other estimates when there is no basis for assigning probability density functions for $\mathbf{x}$ and $\mathbf{z}$.

2.2.2 Recursive Least-Squares Estimation

Consider k measurements of the form

$$\mathbf{z}_i = \mathbf{H}_i\mathbf{x} + \mathbf{v}_i \tag{2.11}$$

and suppose that an additional measurement of $\mathbf{x}$ has been obtained

$$\mathbf{z}_{k+1} = \mathbf{H}_{k+1}\mathbf{x} + \mathbf{v}_{k+1} \tag{2.12}$$

This new observation can be adjoined to the previous ones to give

$$z_{k+1} = H_{k+1}\mathbf{x} + v_{k+1} \tag{2.13}$$

where

$$z_{k+1} = \begin{bmatrix} z_k \\ \mathbf{z_{k+1}} \end{bmatrix} \tag{2.14}$$

$$H_{k+1} = \begin{bmatrix} H_k \\ \mathbf{H_{k+1}} \end{bmatrix} \tag{2.15}$$

$$v_{k+1} = \begin{bmatrix} v_k \\ \mathbf{v_{k+1}} \end{bmatrix} \tag{2.16}$$

Applying Eq.(2.8) directly yields

$$\hat{x}_{k+1_{LS}} = (H_{k+1}^T R_{k+1}^{-1} H_{k+1})^{-1} H_{k+1}^T R_{k+1}^{-1} z_{k+1} \tag{2.17}$$

where R_{k+1}^{-1} is the new weighting matrix for the $k+1$-observation case. Using the definition of Eq.(2.9), $\hat{\mathbf{x}}_{k+1_{LS}}$ may be written as

$$\hat{\mathbf{x}}_{k+1_{LS}} = \mathbf{P}_{k+1} H_{k+1}^T R_{k+1}^{-1} z_{k+1} \tag{2.18}$$

where

$$\mathbf{P}_{k+1} = (H_{k+1}^T R_{k+1}^{-1} H_{k+1})^{-1} \tag{2.19}$$

Such a solution would require the inversion of a $n \times n$ matrix and does not use the previous calculations for $\hat{\mathbf{x}}_{k_{LS}}$. A recursive formula for Eq.(2.17) can be obtained by assuming

$$R_{k+1}^{-1} = \begin{bmatrix} R_k^{-1} & 0 \\ 0 & \mathbf{R_{k+1}^{-1}} \end{bmatrix} \tag{2.20}$$

and using the matrix inversion lemma.

Matrix Inversion Lemma: Suppose $(n \times n)$ matrix $\mathbf{B}$ and $\mathbf{R}$ are positive definite. Let $\mathbf{H}$ be any, possibly rectangular, matrix. Let $\mathbf{A}$ be an $(n \times n)$ matrix related to $\mathbf{B}$, $\mathbf{R}$ and $\mathbf{H}$ according to

$$\mathbf{A} = \mathbf{B} - \mathbf{B}\mathbf{H}^T[\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{H}\mathbf{B} \tag{2.21}$$

Then, $\mathbf{A}^{-1}$ is given by

$$\mathbf{A}^{-1} = \mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \tag{2.22}$$

The proof of this lemma is given in [24]. This leads after some algebraic manipulations [22] to

$$\hat{\mathbf{x}}_{k+1_{LS}} = \hat{\mathbf{x}}_{k_{LS}} + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^T\mathbf{R}_{k+1}^{-1}(\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k_{LS}}) \tag{2.23}$$

2.3 Discrete Kalman Filter

2.3.1 The System and Measurement Models

Consider a dynamical system whose state is described by a linear, vector difference equation

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_{k+1/k}\mathbf{x}_k + \mathbf{w}_k \tag{2.24}$$

$\mathbf{x}_k$ is the n-dimensional state vector, $\mathbf{\Phi}_{k+1/k}$ is the state transition matrix for the time interval (t_k, t_{k+1}) and $\mathbf{w_k}$ is a vector random sequence with known statistics

$$E[\mathbf{w}_k] = 0 \qquad \textit{for all } k \tag{2.25}$$

$$E[\mathbf{w}_k\mathbf{w}_j^T] = \mathbf{Q}_k\delta_{kj} \tag{2.26}$$

where $\delta_{k,j}$ is the Kronecker delta. The initial state $\mathbf{x}_0$ is considered to be a vector random variable with the following statistics

$$E[\mathbf{x}_0] = \hat{\mathbf{x}}_{0/-1} \tag{2.27}$$

$$E[\mathbf{x}_0\mathbf{x}_0^T] = \mathbf{P}_{0/-1} \tag{2.28}$$

$$E[\mathbf{w}_k\mathbf{x}_0^T] = 0 \qquad for\ all\ k \tag{2.29}$$

where

$$\hat{\mathbf{x}}_{0/-1} = State\ estimate\ at\ t_0\ given\ measurement\ at\ t_{-1}$$

$$\mathbf{P}_{0/-1} = Covariance\ matrix\ of\ the\ error\ in\ \hat{\mathbf{x}}_{0/-1}$$

Suppose that at each time t_k there are m measurements $\mathbf{z}_k$ that are linearly dependent on the state and corrupted by additive noise. That is

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.30}$$

where $\mathbf{H}_k$ is a known $(m \times n)$ -dimensional observation matrix. The vector $\mathbf{v_k}$ is an additive, random sequence with

$$E[\mathbf{v}_k] = 0 \qquad for\ all\ k \tag{2.31}$$

$$E[\mathbf{v}_k\mathbf{v}_j^T] = \mathbf{R}_k\delta_{kj} \tag{2.32}$$

Further, assume that the random processes $\mathbf{v}_k$ and $\mathbf{w}_k$ are uncorrelated, i.e.,

$$E[\mathbf{v}_k\mathbf{w}_j^T] = 0 \qquad for\ all\ k,j \tag{2.33}$$

Eq.(2.24) is the signal model and the Eq.(2.30) is referred to as the observation (measurement) model. A block diagram representation of those two models is illustrated in Fig. 2.1. This mathematical model will be used for the derivation of the Kalman filter.

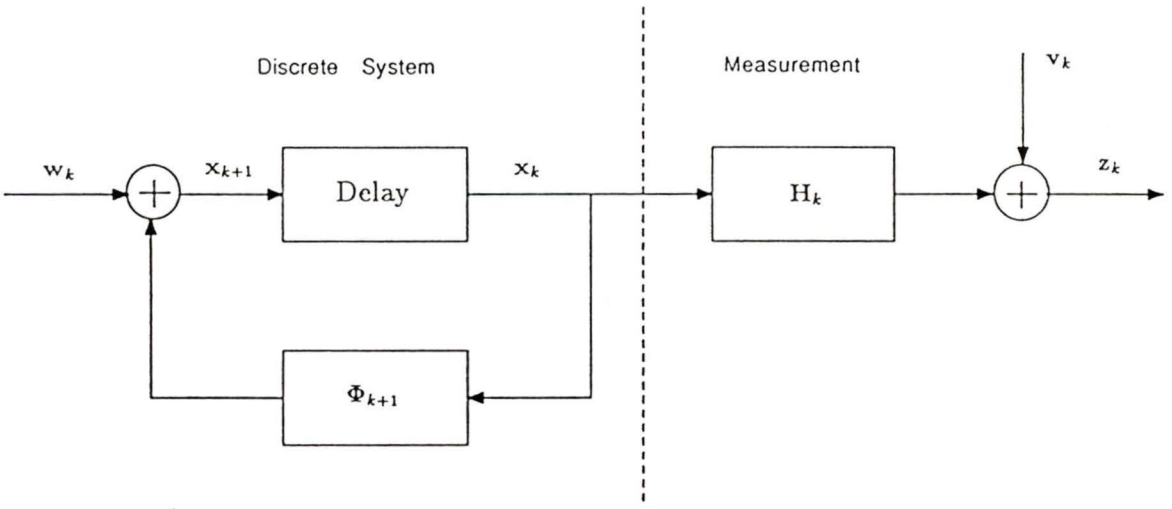


Figure 2.1: Model of Signal and Measurement Processes

2.3.2 Recursive Equations for the Kalman Filter

• **Problem Statement**

Given the preceding model, determine an estimate $\hat{\mathbf{x}}_k$ of the state $\mathbf{x}_k$ at t_k that is a linear combination of an estimate at t_{k-1} and the measurement data $\mathbf{z}_k$. The estimate $\hat{\mathbf{x}}_k$ must be "best" in the sense that the mean-square error

$$E[(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T(\hat{\mathbf{x}}_k - \mathbf{x}_k)]$$

is minimun.

• **Kalman Filter Equations**

An estimate $\hat{\mathbf{x}}_{k/k}$ of the state $\mathbf{x}_k$ is to be computed from the data $z_0, z_1, \ldots, z_k$ so as to minimize the mean- square error in the estimate. This estimate is supposed to be computed only from the measurement z_k and the previous best estimate $\hat{\mathbf{x}}_{k-1/k-1}$. This approach leads to a recursive solution that provides an estimate that is equivalent to the one obtained by processing all the data simultaneously and with reduced computing requirements. The estimate of the signal

$$\mathbf{s}_k = \mathbf{H}_k\mathbf{x}_k \tag{2.34}$$

is given by

$$\hat{\mathbf{s}}_{k/k} = \mathbf{H}_k\hat{\mathbf{x}}_{k/k} \tag{2.35}$$

The solution of this recursive, linear estimation problem can be determined from the orthogonality principle [22]

$$E[\tilde{s}_{k/k}z_i^T] = 0 \qquad i = 0,\ 1,\ \ldots,\ n \tag{2.36}$$

where $\tilde{s}_{k/k}$ is the estimation error given by

$$\tilde{s}_{k/k} = s_k - \hat{s}_{k/k} \tag{2.37}$$

and $z_i (i = 0, 1, \ldots, n)$ are measurements. Thus, the state estimate $\hat{\mathbf{x}}_{k/k}$ is

$$\hat{\mathbf{x}}_{k/k} = \mathbf{\Phi}_{k/k-1}\hat{\mathbf{x}}_{k-1/k-1} + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k\mathbf{\Phi}_{k/k-1}\mathbf{x}_{k-1/k-1}] \tag{2.38}$$

The gain matrix $\mathbf{K}_k$ is chosen to minimize

$$E[(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k/k-1})^T(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k/k-1})]$$

and is given by

$$\mathbf{K}_k = \mathbf{P}_{k/k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k/k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{2.39}$$

The matrix $\mathbf{P}_{k/k-1}$ is the covariance of the error in the predicted estimate and is given by

$$\begin{aligned} \mathbf{P}_{k/k-1} &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k/k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k/k-1})^T] \\ &= \mathbf{\Phi}_{k/k-1}\mathbf{P}_{k-1/k-1}\mathbf{\Phi}_{k/k-1}^T + \mathbf{Q}_{k-1} \end{aligned} \tag{2.40}$$

The $\mathbf{P}_{k/k}$ is the covariance of the error in the estimate $\hat{\mathbf{x}}_{k/k}$ and is given by

$$\begin{aligned} \mathbf{P}_{k/k} &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k/k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k/k})^T] \\ &= \mathbf{P}_{k/k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k/k-1} \end{aligned} \tag{2.41}$$

Table 2.1 gives the definitions and dimensions of the variables and matrices used in the above equations.

Eqs.(2.38)–(2.41) are the recursive Kalman filter equations. The sequence of computational steps are summarized as follows:

- Step 1 Store the filter state $[\hat{\mathbf{x}}_{k/k}, \mathbf{P}_{k/k}]$

- Step 2 Compute the predicted state $\hat{\mathbf{x}}_{k+1/k}$

$$\hat{\mathbf{x}}_{k+1/k} = \mathbf{\Phi}_{k+1/k}\hat{\mathbf{x}}_{k/k}$$

Variable	Definition	Dimension
$\mathbf{z}_k$	*Measurement at* t_k	$m \times 1$
$\hat{\mathbf{x}}_{k/k-1}$	*State estimate at* t_k *given* $\mathbf{z}_{k-1}$	$n \times 1$
$\hat{\mathbf{x}}_{k/k}$	*State estimate at* t_k *given* $\mathbf{z}_k$	$n \times 1$
$\mathbf{\Phi}_{k+1/k}$	*State transition matrix from* t_k *to* t_{k+1}	$n \times n$
$\mathbf{Q}_k$	*System noise covariance matrix*	$r \times r$
$\mathbf{R}_k$	*Measurement noise covariance matrix*	$m \times n$
$\mathbf{H}_k$	*Measurement matrix*	$m \times n$
$\mathbf{K}_k$	*Kalman filter gain matrix at* t_k	$n \times m$
$\mathbf{P}_{k/k-1}$	*Covariance matrix of the error in* $\hat{\mathbf{x}}_{k/k-1}$	$n \times n$
$\mathbf{P}_{k/k}$	*Covariance matrix of the error in* $\hat{\mathbf{x}}_{\mathbf{k/k}}$	$n \times n$

Table 2.1: Kalman Filter Variables

- Step 3 Compute the predicted error covariance matrix $\mathbf{P}_{k+1/k}$

$$\begin{aligned}\mathbf{P}_{k+1/k} &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1/k})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1/k})^T] \\ &= \mathbf{\Phi}_{k+1/k}\mathbf{P}_{k/k}\mathbf{\Phi}_{k+1/k}^T + \mathbf{Q}_k\end{aligned}$$

- Step 4 Compute the filter gain matrix $\mathbf{K}_{k+1}$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1/k}\mathbf{H}_{k+1}^T(\mathbf{H}_{k+1}\mathbf{P}_{k+1/k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1}$$

- Step 5 Process the observation $\mathbf{z}_{k+1}$

$$\hat{\mathbf{x}}_{k+1/k+1} = \hat{\mathbf{x}}_{k+1/k} + \mathbf{K}_{k+1}[\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1/k}]$$

- Step 6 Compute the new error covariance matrix $\mathbf{P}_{k+1/k+1}$

$$\begin{aligned}\mathbf{P}_{k+1/k+1} &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1/k+1})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1/k+1})^T] \\ &= \mathbf{P}_{k+1/k} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{P}_{k+1/k}\end{aligned}$$

- Step 7 Set $k = k + 1$ and return to Step 1

2.3.3 An Alternative Form of the Discrete Kalman Filter

• Problem Statement

Suppose it is required to estimate an unknown constant based on a sequence of independent noisy measurements of that constant where very little is known about the statistics of the process initially. The constant is equally likely to be positive or negative, and its magnitude could be quite large. It might be thought of as a random variable with a flat probability density function extending from ∞ to $-\infty$. They can be expressed by the following equations

$$\hat{x}_{0/-1} = 0 \tag{2.42}$$

$$P_{0/-1} = \infty \tag{2.43}$$

where $\hat{x}_{0/-1}$ and $P_{0/-1}$ are defined in Section 2.3.1. Obviously, Eq.(2.39) and Eq.(2.41) can not be used because they lead to an indeterminant form ∞/∞.

In order to avoid this problem, an alternative form of gain matrix $\mathbf{K}_k$ and error covariance matrix $\mathbf{P}_{k/k}$ in the Kalman filter is derived. It is based on the matrix inversion lemma presented in Section 2.2.2 using the following correspondences

$$\mathbf{A} \sim \mathbf{P}_{k/k}$$

$$\mathbf{B} \sim \mathbf{P}_{k/k-1}$$

$$\mathbf{H} \sim \mathbf{H}_k$$

$$\mathbf{R} \sim \mathbf{R}_k$$

Assume that $\mathbf{R}_k$ and $\mathbf{P}_k$ are positive definite and apply the Eq.(2.22)

$$\mathbf{P}_{k/k}^{-1} = \mathbf{P}_{k/k-1}^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \tag{2.44}$$

where

$$\mathbf{P}_{k/k-1}^{-1} = [\mathbf{\Phi}_{k/k-1} \mathbf{P}_{k-1/k-1} \mathbf{\Phi}_{k/k-1}^T + \mathbf{Q}_{k-1}]^{-1} \tag{2.45}$$

Now, rewrite Eq.(2.39)

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{k/k-1} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \\ &= (\mathbf{P}_{k/k} \mathbf{P}_{k/k}^{-1}) \mathbf{P}_{k/k-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} [\mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} + \mathbf{I}]^{-1} \end{aligned} \tag{2.46}$$

which gives using Eq.(2.44)

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_{k/k}[\mathbf{I}+\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{P}_{k/k-1}] \\ &\quad \cdot\mathbf{H}_k^T\mathbf{R}_k^{-1}[\mathbf{H}_k\mathbf{P}_{k/k-1}\mathbf{H}_k^T\mathbf{R}_k^{-1}+\mathbf{I}]^{-1} \\ &= \mathbf{P}_{k/k}\mathbf{H}_k^T\mathbf{R}_k^{-1} \qquad (2.47)\end{aligned}$$

Note that in Eq.(2.44) the updated error covariance $\mathbf{P}_{k/k}$ can be computed without first finding the gain. Therefore, if Eq.(2.47) is used, $\mathbf{K}_k$ must be computed after $\mathbf{P}_{k/k}$ has been computed. Thus, the order in which the $\mathbf{P}_{k/k}$ and $\mathbf{K}_k$ computations appear in the recursive algorithm is reversed from that presented in Section 2.3.2. To clarify the above statement, let's summarize the alternative algorithm of the discrete Kalman filter as follows:

- Step 1 Store the filter state $[\hat{\mathbf{x}}_{k/k-1}, \mathbf{P}_{k/k-1}]$

- Step 2 Compute the updated error covariance matrix $\mathbf{P}_{k/k}$ from

$$\mathbf{P}_{k/k}^{-1} = \mathbf{P}_{k/k-1}^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k$$

$$\mathbf{P}_{k/k} = (\mathbf{P}_{k/k}^{-1})^{-1}$$

- Step 3 Compute the filter gain matrix from

$$\mathbf{K}_k = \mathbf{P}_{k/k}\mathbf{H}_k^T\mathbf{R}_k^{-1}$$

- Step 4 Update estimate

$$\hat{\mathbf{x}}_{k/k} = \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k/k-1})$$

- Step 5 Project ahead

$$\hat{\mathbf{x}}_{k+1/k} = \mathbf{\Phi}_{k+1/k}\hat{\mathbf{x}}_{k/k}$$

$$\mathbf{P}_{k+1/k} = \mathbf{\Phi}_{k+1/k}\mathbf{P}_{k/k}\mathbf{\Phi}_{k+1/k} + \mathbf{Q}_k$$

- Step 6 Invert $\mathbf{P}_{k+1/k}$

- Step 7 Set $k = k + 1$ and return to Step 1

The disadvantage of this algorithm is that if the order of the state is large, the two $(n \times n)$ matrix inversions could lead to numerical problems.

2.4 Relationship between Deterministic Least-Squares Estimation and Kalman Filter

There is a connection between Kalman filter and deterministic least-squares estimation due to the fact that the criterion for optimization is minimum mean-square error and not the squared error in a deterministic sense.

Consider a set of m linear equations in $\mathbf{x}$ specified in matrix form by

$$\mathbf{M}\mathbf{x} = \mathbf{b} \tag{2.48}$$

In Eq.(2.48), we think of $\mathbf{M}$ and $\mathbf{b}$ as being given, and $\mathbf{x}$ is $(n \times 1)$, $\mathbf{b}$ is $(m \times 1)$, and thus $\mathbf{M}$ is $(m \times n)$. Let us assume that $m > n$, and that $\mathbf{x}$ is overdetermined by the system of equations represented by Eq.(2.48). Thus no solution for $\mathbf{x}$ will satisfy all equations. This situation arises frequently in physical experiments where redundant noisy measurement are made of linear combinations of fixed parameters. In such cases, it is logical to ask, "What solution will best fit all the equations?" The term best must, of course, be

defined and it is frequently defined to be the particular $\mathbf{x}$, say $\mathbf{x_{opt}}$, that minimizes the sum of the squared residuals $(\mathbf{b} - \hat{\mathbf{x}})$. That is, move $\mathbf{b}$ to the left side of Eq.(2.48) and substitute $\mathbf{x_{opt}}$ for $\mathbf{x}$. This yields a residual vector $\mathbf{e}$ given by

$$\mathbf{e} = \mathbf{M}\mathbf{x}_{opt} - \mathbf{b} \tag{2.49}$$

and $\mathbf{x}_{opt}$ is chosen such that $\mathbf{e}^T\mathbf{e}$ is minimized. A perfect fit, of course, would make $\mathbf{e}^T\mathbf{e} = 0$.

This point can be generalized and consider a weighted sum of squared residuals specified by

$$\begin{bmatrix} \textbf{weighted sum of} \\ \textbf{squared residuals} \end{bmatrix} = (\mathbf{M}\mathbf{x}_{opt} - \mathbf{b})^T\mathbf{W}(\mathbf{M}\mathbf{x}_{opt} - \mathbf{b}) \tag{2.50}$$

It is assumed that the weighting matrix $\mathbf{W}$ is symmetric and positive definite and, hence, so is its inverse. If we wish to have equal weighting of the residuals, we simply let $\mathbf{W}$ be the identity matrix. The problem now is to find the particular $\mathbf{x}$ (i.e.,$\mathbf{x}_{opt}$) that minimize the weighted sum of the residuals. Toward this end, the expression given by Eq.(2.50) may be expanded and differentiated term by term and then set equal to zero. This leads to

$$\frac{d}{d\mathbf{x_{opt}}}[\mathbf{x}_{\mathbf{opt}}^{\mathbf{T}}(\mathbf{M}^{\mathbf{T}}\mathbf{W}\mathbf{M})\mathbf{x_{opt}} - \mathbf{b}^{\mathbf{T}}\mathbf{W}\mathbf{M}\mathbf{x_{opt}} - \mathbf{x}_{\mathbf{opt}}^{\mathbf{T}}\mathbf{M}^{\mathbf{T}}\mathbf{W}\mathbf{b} + \mathbf{b}^{\mathbf{T}}\mathbf{b}]$$

$$= 2(\mathbf{M}^{\mathbf{T}}\mathbf{W}\mathbf{M})\mathbf{x_{opt}} - (\mathbf{b}^{\mathbf{T}}\mathbf{W}\mathbf{M})^{\mathbf{T}} - \mathbf{M}^{\mathbf{T}}\mathbf{W}\mathbf{b} = 0 \tag{2.51}$$

Eq.(2.51) may now be solved for $\mathbf{x_{opt}}$. The result is

$$\mathbf{x}_{opt} = [(\mathbf{M}^T\mathbf{W}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{W}]\mathbf{b} \tag{2.52}$$

and this is the solution of the deterministic least-squares problem.

Next consider the Kalman filter solution for the same measurement situation. The vector $\mathbf{x}$ is assumed to be a constant, so the differential equation for $\mathbf{x}$ is

$$\dot{\mathbf{x}} = 0 \tag{2.53}$$

The corresponding discrete model is then

$$\mathbf{x}_{k+1} = \mathbf{x}_k \tag{2.54}$$

and the measurement equation is

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.55}$$

where $\mathbf{z}_k$ and $\mathbf{H}_k$ play the same roles as $\mathbf{b}$ and $\mathbf{M}$ in the deterministic problem. Since time is of no consequence, it is assumed all measurements occur simultaneously. Furthermore, it is assumed that we have no prior knowledge of $\mathbf{x}$, so the initial $\hat{\mathbf{x}}_{0/-1}$ will be zero and its associated error covariance will be ∞. Therefore, using the alternative form of the Kalman filter (Section 2.3.3), it follows that

$$\begin{aligned} \mathbf{P}_{0/-1} &= (\infty)^{-1} + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 \\ &= \mathbf{H}_0\mathbf{R}_0^{-1}\mathbf{H}_0 \end{aligned} \tag{2.56}$$

The Kalman gain is then

$$\mathbf{K}_0 = (\mathbf{H}_0\mathbf{R}_0^{-1}\mathbf{H}_0)^{-1}\mathbf{H}_0^T\mathbf{R}_0^{-1} \tag{2.57}$$

and the Kalman filter estimate of $\mathbf{x}$ at $t = 0$ is

$$\hat{\mathbf{x}}_0 = [(\mathbf{H}_0\mathbf{R}_0^{-1}\mathbf{H}_0)^{-1}\mathbf{H}_0^T\mathbf{R}_0^{-1}]\mathbf{z}_0 \tag{2.58}$$

This is the same identical expression obtained for $\mathbf{x}_{opt}$ in the deterministic least-squares problem with $\mathbf{R}_0^{-1}$ playing the role of the weighting matrix $\mathbf{W}$.

Remark The condition under which Kalman filter estimate coincides with the least-squares estimate is summarized as follows:

- The system state vector is assumed to be a constant (the process dynamics are thus trivial).

- It is assumed that the measurement sequence was such as to yield an overdetermined system of linear equations (otherwise $(\mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{H}_0)^{-1}$ will not exist).

- It is assumed that there is no prior knowledge of the constant vector being estimated.

Chapter 3

A Generalized Algorithm for the Recursive Implementation of Polynomial Filters

3.1 Introduction

There are many applications such as inventory control, market analysis, radar tracking, missile guidance and gun fire control where the information about the system dynamics and noise statistics are not precisely known. Finite memory filters have a finite impulse response and are well suited to such situations where the measurement and process noise models are either unknown or too complex [11].

A special class of finite memory filters is the class of polynomial finite memory filters [25]. These filters approximate measurements in an observation window by a polynomial using a least-squares performance index. This type of filter, especially for high order, requires a considerable amount of computation time because a polynomial has to be fitted at each step. In order to reduce the computation time for applications where the measurements are made periodically, recursive polynomial filters were designed in [15]. Unfortunately, this recursive filters have removable sigularities on the boundary

of the stability region and therefore, can become unstable in practice. This problem was solved in [17] for the simple first order case using a weighted least-squares performance index which has the effect of introducing a stability margin and improving the stability behaviour of the resulting recursive finite memory filter.

In this chapter, the result of [15] and [17] are extended to high order filters. An nth order polynomial is used to approximate measurements in a window of length L using weighted least-squares which leads to a recursive finite memory filter. The equations for the nonrecursive polynomial filter are developed in Section 3.2 and in Section 3.3 the recursive implementation of these equations is considered. A comparison of the performance of the recursive and nonrecursive filter is presented in section 3.4. An example of filtering audio signals using the recursive filter is considered in Section 3.5.

3.2 Nonrecursive Filter

Consider a continuous signal $y(t)$ which is sampled every T seconds. Without loss of generality, T can be normalized to unity. It is required to estimate the sampled signal $y(k), k = 1, 2, \ldots$, given measurements of $y(k)$ in an observation window of length L. $\hat{y}(k)$, the estimate of $y(k)$ based on measurements in the interval $(k - L)$ to k, will be modeled by the following nth order polynomial

$$\hat{y}_n(k - L + i) = a_0(k) + a_1(k)i + a_2(k)i^2 + \ldots + a_n(k)i^n \tag{3.1}$$

where $i = 0, 1, 2, \ldots, L$. $a_0(k), a_1(k), \ldots, a_n(k)$ are determined using weighted least-squares estimation and the performance index

$$E(k) = \sum_{i=0}^{L} e^2(i)\beta^{i-L} \tag{3.2}$$

where

$$e(i) = [\hat{y}(k-L+i) - y(k-L+i)] \tag{3.3}$$

and β is a weighting coefficient. Then, Eq.(3.2) can be rewritten as

$$\begin{aligned} E(k) &= \sum_{i=0}^{L}[a_0(k) + a_1(k)i + a_2(k)i^2 \\ &\quad + \ldots + a_n(k)i^n - y(k-L+i)]^2\beta^{i-L} \end{aligned} \tag{3.4}$$

Minimizing E w.r.t. $a_0, a_1, a_2, \ldots, a_n$ we obtain

$$\mathbf{Ba} = \mathbf{c} \tag{3.5}$$

where

$$\mathbf{a}^T = \mathbf{a}^T(k) = [a_0(k), a_1(k), a_2(k), \ldots, a_n(k)] \tag{3.6}$$

and $\mathbf{B} \in R^{(n+1)\times(n+1)}$ is a Hankel matrix [26]

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & b_3 & \ldots & b_{n+1} \\ b_2 & b_3 & & \ldots & b_{n+2} \\ b_3 & & & & \\ \vdots & \vdots & & & \ldots \\ b_{n+1} & b_{n+2} & b_{n+3} & \ldots & b_{2n+1} \end{bmatrix} \tag{3.7}$$

where the elements have the following form

$$b_1 = \sum_{i=0}^{L}\beta^i = \frac{1-\beta^{L+1}}{1-\beta} \tag{3.8}$$

$$\begin{aligned} b_j &= \sum_{i=0}^{L} i^{j-1}\beta^i \\ &= \frac{1}{1-\beta}\{\sum_{p=1}^{j-1}(-1)^{p+1}\frac{(j-1)(j-2)\ldots(j-p)}{p!}b_{j-p} \\ &+ (-1)^{j-1} - L^{j-1}\beta^{L+1}\} \end{aligned}$$

$$for \quad j = 2, 3, \ldots, 2n+1 \tag{3.9}$$

The vector

$$\mathbf{c}^T = [c_1(k), c_2(k), c_3(k), \ldots, c_{n+1}(k)] \tag{3.10}$$

in Eq.(3.5) is given by

$$\begin{aligned} c_j(k) &= \sum_{i=0}^{L} i^{j-1}\beta^i y(k-L+i) \\ &= \sum_{i=k-L}^{k} (i-k+L)^{j-1}\beta^{i-k+L} y(i) \end{aligned}$$

$$for \quad j = 1, 2, \ldots, n+1 \tag{3.11}$$

Using Eq.(3.5) and

$$\mathbf{p}^T = [1, L, L^2, \ldots, L^n] \tag{3.12}$$

Eq.(3.1) can be expressed in the following vector form

$$\hat{y}_n(k) = \mathbf{a}^T\mathbf{p} \tag{3.13}$$

Using Eq.(3.5), Eq.(3.13) can be rewritten in the following form

$$\hat{y}_n(k) = (\mathbf{B}^{-1}\mathbf{c})^T\mathbf{p} = \mathbf{c}^T\mathbf{B}^{-1}\mathbf{p} = \mathbf{c}^T\mathbf{m} \tag{3.14}$$

where $\mathbf{m}$ is the solution of

$$\mathbf{B}\mathbf{m} = \mathbf{p} \tag{3.15}$$

Thus, $\hat{y}_n(k)$, the estimate of $y(k)$, using measurements of $y(k)$ in the interval $k-n, \ldots, k$, is obtained as

$$\hat{y}_n(k) = m_1c_1(k) + m_2c_2(k) + \ldots + m_{n+1}c_{n+1}(k)$$

$$= \sum_{i=0}^{L}(m_1 + im_2 + i^2m_3 + \ldots + i^n m_{n+1})\beta^i y(k-L+i)$$

$$= \sum_{i=k-L}^{k}[(m_1 + (i-k+L)m_2 + (i-k+L)^2 m_3$$

$$+ \ldots + (i-k+L)^n m_{n+1}]\beta^{i-k+L}y(i) \qquad (3.16)$$

Eq.(3.16) represents the nonrecursive polynomial filter.

3.3 Realization of the Recursive Filter

The main disadvantage of the nonrecursive filter derived above is that it requires a great deal of computation, especially when the order of the polynomial is high. To overcome this problem, the following algorithm of the recursive finite memory filter is developed. Eq.(3.11) can be evaluated recursively by knowing that every time a set of observations is added another set of observations is deleted. For $c_i(k), i = 1, 2, \ldots, n+1$, this gives the following recursive relations

$$c_1(k+1) = \sum_{i=k+1-L}^{k+1} \beta^{i-k+L-1}y(i)$$

$$= \frac{1}{\beta}c_1(k) - \frac{1}{\beta}y(k-L) + \beta^L y(k+1) \qquad (3.17)$$

$$c_j(k+1) = \sum_{i=k+1-L}^{k+1} (i-k+L-1)^{j-1}\beta^{i-k+L-1}y(i)$$

$$= \frac{1}{\beta}[c_j(k) - (j-1)c_{j-1}(k) + \frac{(j-1)(j-2)}{2!}c_{j-2}(k)$$

$$- \frac{(j-1)(j-2)(j-3)}{3!}c_{j-3}(k) + \ldots$$

$$+ \quad (-1)^{j-p}\frac{(j-1)(j-2)(j-3)\ldots(j-p)}{p!}c_{j-p}(k)$$

$$+ \quad \ldots$$

$$+ \quad (-1)^{j+1}c_1(k) + (-1)^j y(k-L) + L^{j-1}\beta^L y(k+1)]$$

$$for \quad j \;=\; 2,3,\ldots,n+1 \tag{3.18}$$

Substituting $c_j(k+1), j = 1,2,\ldots,n$ from Eq.(3.11) into Eq.(3.14) yields

$$\hat{y}_n(k+1) \;=\; \mathbf{c}^T(k+1)\mathbf{m} \tag{3.19}$$

This gives

$$\begin{aligned}
\hat{y}_n(k+1) \;&=\; m_1c_1(k+1) + m_2c_2(k+1) + \ldots + m_{n+1}c_{n+1} \\
&=\; \frac{1}{\beta}\hat{y}(k) + \frac{1}{\beta}\sum_{q=1}^{n}(-1)^q m_{q+1}c_1(k) \\
&+\; \frac{1}{\beta}\sum_{p=2}^{n}[\sum_{q=p}^{n}(-1)^{p+q-1} \\
&\quad \frac{q(q-1)(q-2)\ldots[q-(p-2)]}{(p-1)!}m_{q+1}]c_p(k) \\
&+\; \sum_{q=1}^{n+1}(-1)^q m_q y(k-L) + \beta^L\sum_{q=1}^{n+1}L^{q-1}m_q y(k+1)
\end{aligned} \tag{3.20}$$

Eq.(3.17)–(3.18) together with Eq.(3.20) constitute the realization of the recursive finite memory filter. They allow the computation of $\hat{y}_n(k+1)$ recursively. These equations actually represent the state equations and therefore they can be described in the following state-space realization

$$\mathbf{x}(k+1) \;=\; \mathbf{A}\mathbf{x}(k) + \mathbf{F}\mathbf{u}(k) \tag{3.21}$$

$$\hat{y}_n(k) = \mathbf{h}^T \mathbf{x}(k) \tag{3.22}$$

where

$$\mathbf{x}(k) = [c_1(k), c_2(k), \ldots, c_n(k), \hat{y}(k)]^T \tag{3.23}$$

and $\mathbf{A} \in R^{(n+1)\times(n+1)}$ is given by

$$\mathbf{A} = \beta^{-1} \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} & 0 \\ \vdots & \cdots & a_{i,j} & \cdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} & 0 \\ a_{n+1,1} & \cdots & a_{n+1,j} & \cdots & a_{n+1,n} & 1 \end{bmatrix} \tag{3.24}$$

where

$$a_{i,j} = \begin{cases} (-1)^{i+j} \frac{(i-1)(i-2)\ldots(i-j+1)}{(j-1)!} & \text{for } i \geq j \\ 0 & \text{for } i < j \end{cases}$$

$$i, j = 1, 2, \ldots, n \tag{3.25}$$

$$a_{n+1,1} = \sum_{q=1}^{n} (-1)^q m_{q+1} \tag{3.26}$$

$$a_{n+1,j} = \sum_{q=j}^{n} (-1)^{q+j-1} \frac{q(q-1)(q-2)\ldots[q-(j-2)]}{(j-1)!} m_{q+1}$$

$$j = 2, 3, \ldots, n \tag{3.27}$$

and $\mathbf{F} \in R^{(n+1)\times 2}$ is given by

$$\mathbf{F} = \beta^{-1} \begin{bmatrix} -1 & \beta^{L-1} \\ 1 & L\beta^{L-1} \\ -1 & L^2\beta^{L-1} \\ \vdots & \vdots \\ (-1)^n & L^{n-1}\beta^{L-1} \\ \sum_{q=1}^{n+1}(-1)^q m_q & \beta^{L-1}\sum_{q=1}^{n+1} L^{q-1} m_q \end{bmatrix} \tag{3.28}$$

	Recursive	Nonrecursive
Additions	$5+(n-1)(3+\frac{n}{2})$	L
Multiplications	$7+(n-1)(4+\frac{n}{2})$	$L+1$

Table 3.1: Computational Effort for the Recursive Eqs.(3.21)-(3.22) and Nonrecursive Filter

and

$$\mathbf{u}(k) = \begin{bmatrix} y(k-L) \\ y(k+1) \end{bmatrix} \tag{3.29}$$

$$\mathbf{h} = [\mathbf{z}|1]^T \tag{3.30}$$

where $\mathbf{z}$ is $1 \times n$ row vector with all elements zero.

3.4 Performance of the Recursive and Nonrecursive Filters

In this section, the performance of the recursive and nonrecursive filters will be compared with respect to the amount of computation required and noise characteristics. Also, the stability and the frequency response of the recursive and nonrecursive filters will be discussed.

3.4.1 Computational Effort

To compare the amount of computation, we consider the number of additions and multiplications needed to implement the filters. The number of the additions and multiplications for the recursive filter can be obtained from Eq.(3.21)–(3.22), while the corresponding number of the nonrecursive filter can be computed from Eq.(3.16). These are given in Table 3.1.

From Table 3.1, it can be seen that for the nonrecursive filter, the amount of computation depends on the window size and will remain unchanged as the order of the filter varies. On the other hand, the amount of computation of the recursive filter does not depend on the window size but on the order of the filter. Some further savings on the multiplications of the recursive filter can be achieved due to the specific structure of matrices A and F. By modifying Eq.(3.21), the following equation can be obtained

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{A}}\hat{\mathbf{x}}(k) + \hat{\mathbf{F}}\mathbf{u}(k) \tag{3.31}$$

where

$$\hat{\mathbf{x}}(k) = \beta\mathbf{x}(k) \tag{3.32}$$

$$\hat{\mathbf{A}} = \beta\mathbf{A} \tag{3.33}$$

$$\hat{\mathbf{F}} = \beta\mathbf{F} \tag{3.34}$$

This implementation shows that we can save $2n-1$ multiplications with respect to Table 3.1. Thus, the further comparison of computational effort between the recursive and nonrecursive filter is summerized in Table 3.2.

3.4.2 Stability

The nonrecursive finite memory filter is always stable because its impulse response is finite and therefore it is absolutely summable. To investigate the stability of the recursive filter, the state-space representation of the recursive filter is considered. Taking the z-transform of Eqs.(3.21)–(3.22), the transfer function $G(z)$ of the recursive filter is obtained

$$\mathbf{G}(z) = \mathbf{h}^T(\mathbf{zI} - \mathbf{A})^{-1}\mathbf{FU}(z) \tag{3.35}$$

order	Recursive		Nonrecursive	
	additions	multiplications	additions	multiplications
First	5	6	L	$L+1$
Second	9	9	L	$L+1$
Third	14	13	L	$L+1$
Fourth	20	18	L	$L+1$
Fifth	27	24	L	$L+1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
nth	$2+\frac{n(n+5)}{2}$	$4+\frac{n(n+3)}{2}$	L	$L+1$

Table 3.2: Comparison of Computational Effort for the Recursive (Eqs.(3.31)-(3.34)) and Nonrecursive Filter

From Eq.(3.24), we know that **A** is a low triangular matrix and has all β^{-1} in its diagonal. Hence, the eigenvalues of the characteristic equation of **A** are at $z=\beta^{-1}$ with multiplicity $n+1$. It can be verified that the transfer function of the recursive and nonrecursive finite memory filters are the same. This implies that there exist pole-zero cancellations in the recursive filter. As a result, the recursive finite memory filter of nth order is stable provided that $\beta > 1$.

Pole-zero cancellations are very sensitive to filter coefficient perturbations and therefore, in a finite word length implementation the filter may still have poles at β^{-1}. In such a case, the filter could become unstable if $\beta = 1$ as is the case with the filter in [15]. For $\beta > 1$, however, the recursive filter will always be stable. An increase of β implies an increased stability margin [27] for the filter, i.e. a larger distance of the poles from the unit circle. Consequently, although the stability of the recursive filter does not depend on β, in theory, by selecting $\beta > 1$ the stability of the filter upon implementation is assured.

3.4.3 Noise performance

During the implementation of the finite memory filter, noise is always present. Measurement noise is caused by errors in measurement, while process noise could be caused by round-off and quantization errors. Here the effect of measurement noise on the filter output is considered.

Eq.(3.16), the nonrecursive filter, can be expressed as

$$\hat{y}_n(k) = \mathbf{d}^T \mathbf{Y}(k) \tag{3.36}$$

where

$$\mathbf{Y}(k) = [y(k), y(k-1), \ldots, y(k-L)]^T \tag{3.37}$$

$$\mathbf{d} = \begin{bmatrix} (m_1 + Lm_2 + \ldots + L^n m_{n+1})\beta^L \\ (m_1 + (L-1)m_2 + \ldots + (L-1)^n m_{n+1})\beta^{L-1} \\ \vdots \\ (m_1 + m_2 + \ldots + m_{n+1})\beta \\ m_1 \end{bmatrix} \tag{3.38}$$

In the presence of noise, the measurement vector becomes

$$\mathbf{Y}^*(k) = \mathbf{Y}(k) + \eta(k) \tag{3.39}$$

where $\eta(k)$ is the measurement noise vector. The noise is assumed to have zero mean and variance $\sigma^2 I$, where I is the identity matrix. The estimate of $y(k)$ in the presence of noise is obtained as

$$\tilde{\mathbf{y}}(k) = \mathbf{d}^T \mathbf{Y}^*(k) \tag{3.40}$$

The error in the estimation of $y(k)$ due to the presence of noise is

$$e(k) = \tilde{\mathbf{y}}(k) - \hat{\mathbf{y}}(k) \tag{3.41}$$

The expected value of $e(k)$ is zero because of the assumption of zero mean for the noise and its variance is

$$\begin{aligned} Var[e(k)] &= \sigma^2 \mathbf{d}^T \mathbf{d} \\ &= \sigma^2 \sum_{i=0}^{L} [(m_1 + m_2 i + m_3 i^2 + \ldots + m_{n+1} i^n)\beta^i]^2 \end{aligned} \tag{3.42}$$

Eq.(3.42) shows that $Var[e(k)]$ is constant and depends only on β and L. For $n = 2$ and $n = 4$, the error variance is given in Table 3.3 for different values of β and L where σ is assumed to be unity.

From the result shown in Table 3.3, it can be seen that if β is increased, the error variance is also increased, while an increase in L causes a decrease of the error variance. If the order of the filter is increased, the error variance is increased as well. Theoretically, the error variance for the recursive filter is the same as for the nonrecursive filter, since the operations performed on the input data are equivalent. In a finite-word-length implementation, some small changes in the values of the filter coefficients usually occur, and therefore the behaviour of the recursive and nonrecursive filters may be different. It can even become unstable if the stability margin, which is equal to $|1-\beta^{-1}|$, is too small. Therefore, in the choise of β, the trade-off between increased stability margin and increased error variance has to be considered.

3.4.4 Frequency Response

The magnitude response of the recursive filter for different values of β, L and n are presented in Fig. 3.1 to Fig. 3.3. It can be seen that for a constant n, increasing the window size L narrows the bandwidth, while increasing the value of β reduces the filter selectivity. If the order n is increased for constant L and β, then the bandwidth of the filter is increased.

L \ n, β	$n=2$			$n=4$		
	1.0	1.2	1.4	1.0	1.2	1.4
6	0.7619	0.7805	0.8116	0.9870	0.9889	0.9916
9	0.6182	0.6539	0.7095	0.9371	0.9476	0.9615
12	0.5165	0.5680	0.6444	0.8720	0.8957	0.9249

Table 3.3: Error variance Var[e(k)] for some values of β, L and n, assuming $\sigma = 1.0$

3.5 Experimental Results

A noisy speech signal is processed with several finite memory filters of different order. The sentence, "*Did you buy father a plastic tie? He should be proud of wearing anything you choose*", is digitized from a recording. The sampling rate is 6,500 samples per second and therefore the total sampling points for the signal are 28,900. The speech signal is corrupted by adding white Gaussian noise resulting in 3dB signal-to-noise ratio.

One part of that digitized signal containing both voiced and unvoiced sound is chosen from the whole sentence and filtered using finite memory filters of three different orders. Fig. 3.4 shows the original signal and Fig. 3.5 shows the noisy signal. Three filtered signals are illustrated in Fig. 3.6.

For comparison, the relative square error between the original signal and the filtered one is considered. It is defined as

$$e = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N} y_i^2}} \tag{3.43}$$

where y_i is the value of the original signal at sampling point i, $\hat{y}_i$ is the value of the enhanced signal at sampling point i and N is the total sampling points processed. The errors for different L and n are summarized in Table 3.4.

From Table 3.4, it can be seen that the best results are from the first

L \ S	1	2	3
3	0.5297	0.6100	0.6347
6	0.4587	0.5501	0.6046
9	0.5073	0.5067	0.5807
12	0.6413	0.4863	0.5507
15	0.7836	0.5151	0.5309
18	0.9160	0.5693	0.5321
21	1.0404	0.6381	0.5389
24	1.1418	0.7297	0.5520

Table 3.4: Error between original signal and estimated signal for $\beta = 1.01$

order filter with $L = 6$, the second order filter with $L = 12$ and the third order filter with $L = 15$.

3.6 Conclusion

The recursive implemention of nth order polynomial filter is presented. The parameters of the filter have been obtained using a weighted least-squares performance index. The stability of the recursive filter is assured by introducing the weighting factor β in the performance index. The performance of the recursive and nonrecursive implementation is compared with respect to stability, computational effort and noise performance. The performance of the filter is illustrated with an example.

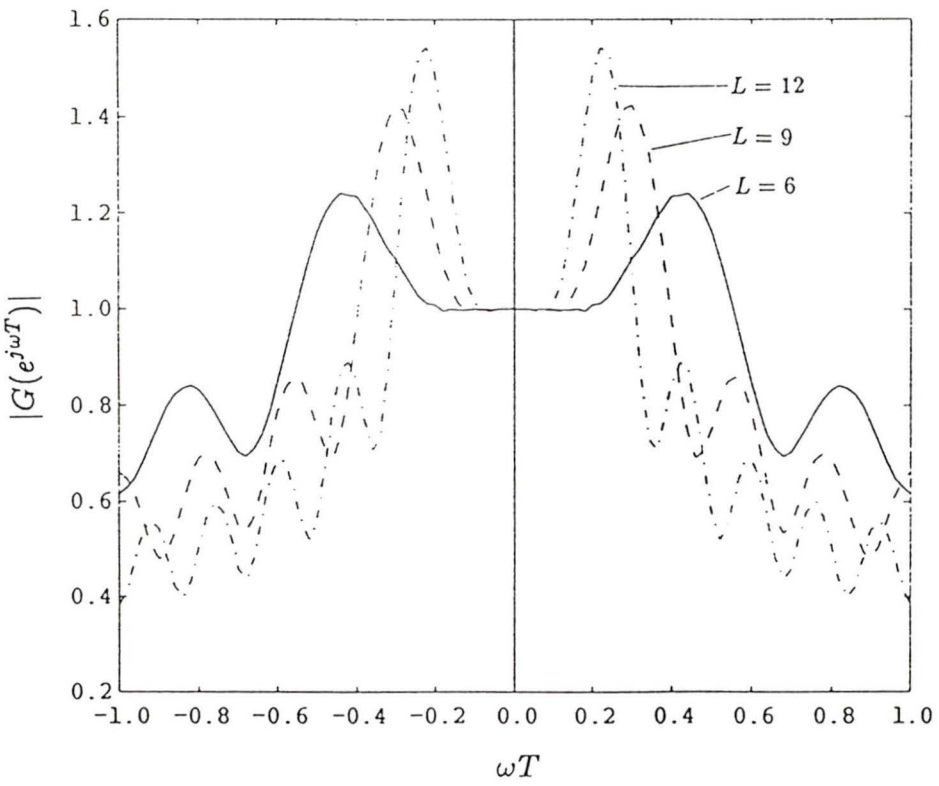


Figure 3.1: Magnitude Response of Finite Memory Filter for $\beta = 1,\ n = 3$

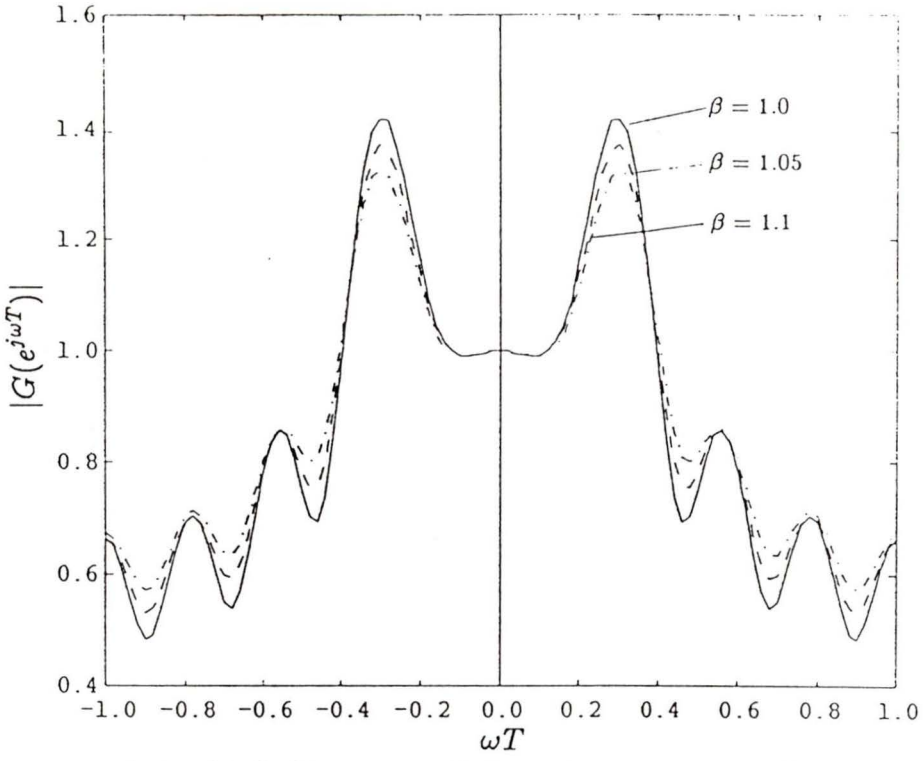


Figure 3.2: Magnitude Response of Finite Memory Filter for $L = 9$, $n = 3$

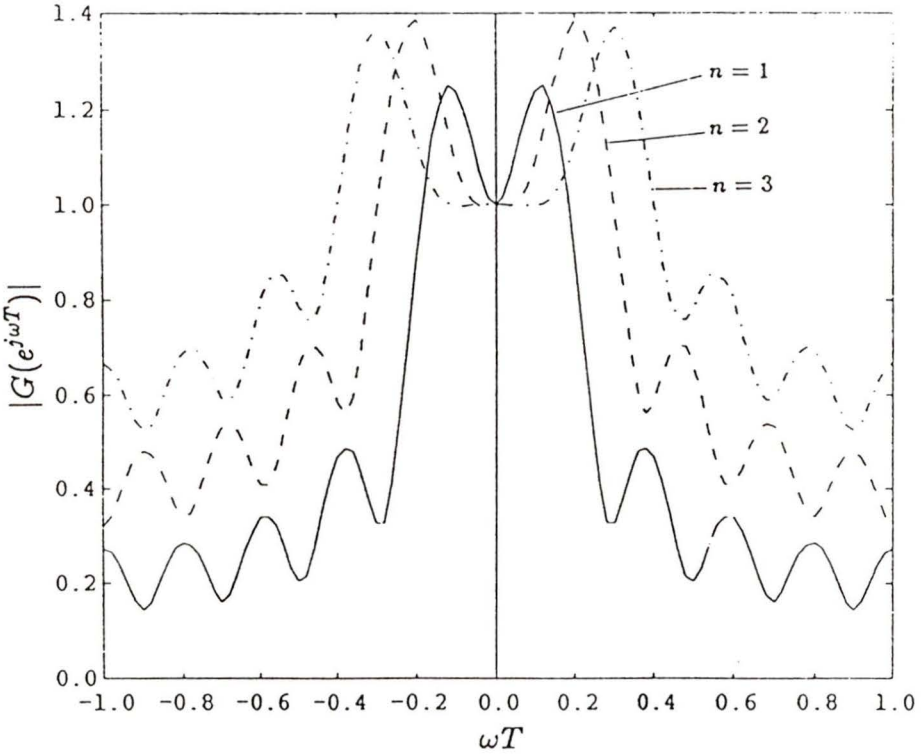


Figure 3.3: Magnitude Response of Finite Memory Filter for $\beta = 1.05$, $L = 9$

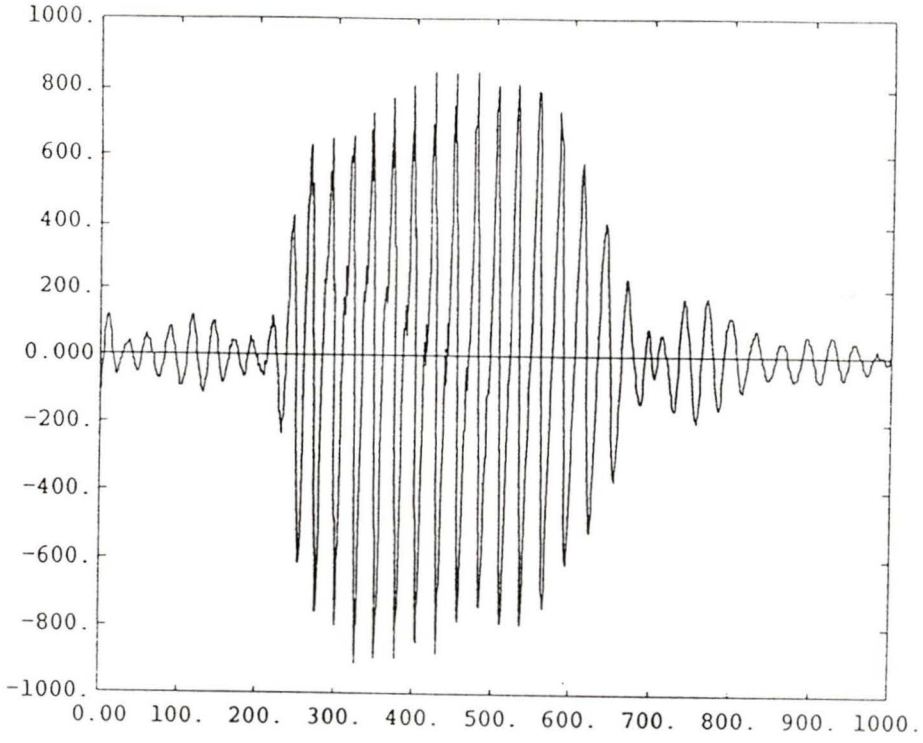


Figure 3.4: Original Signal

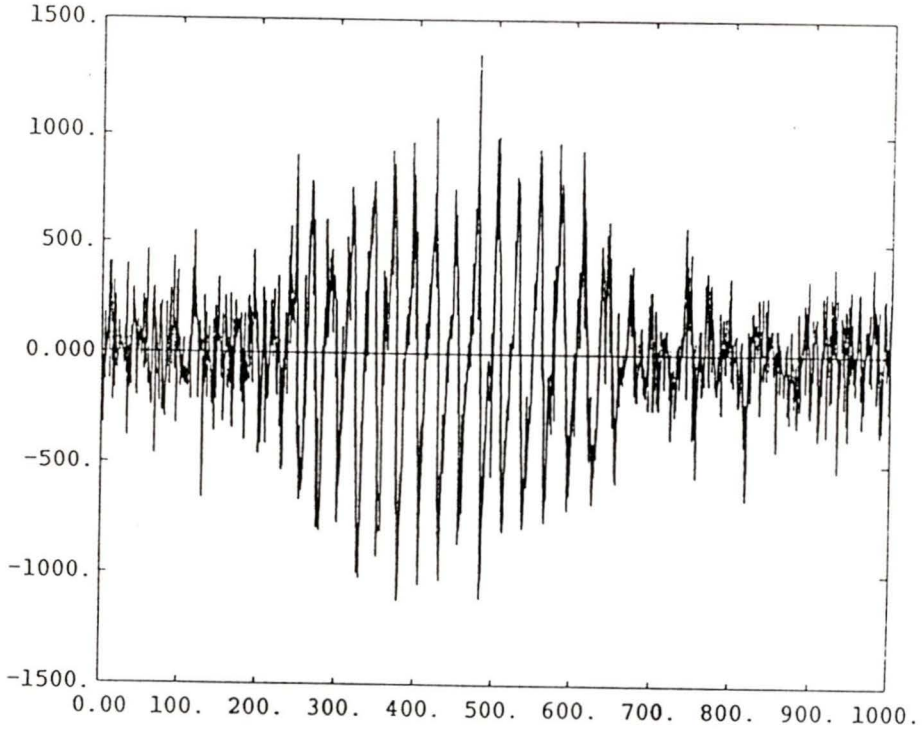


Figure 3.5: Signal Corrupted with Noise

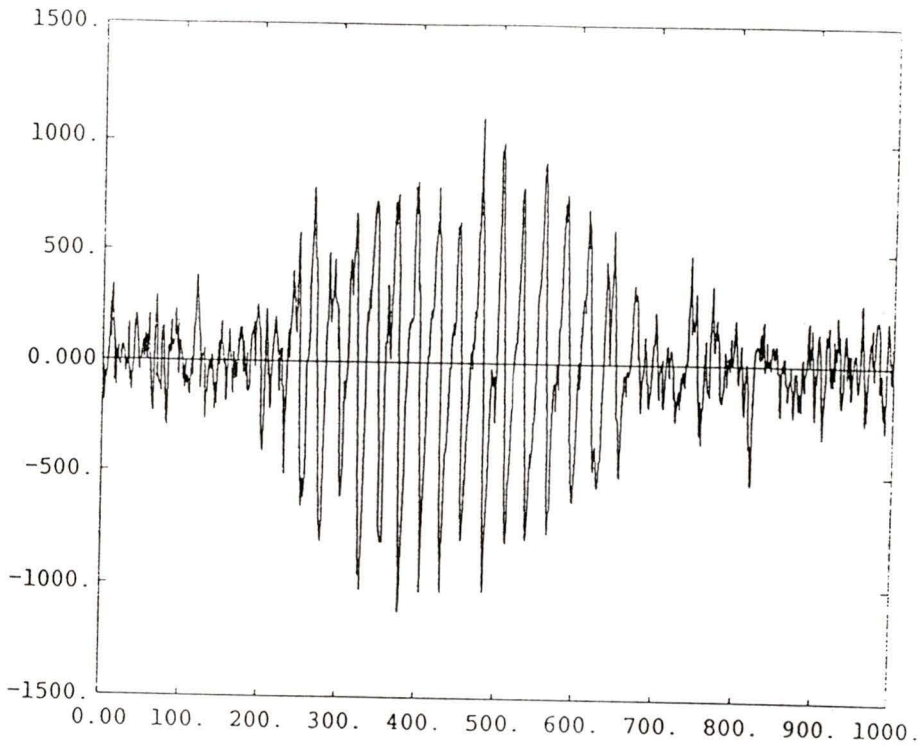


Figure 3.6: Filtered Signal for $\beta = 1.01,\ L = 6,\ n = 1$

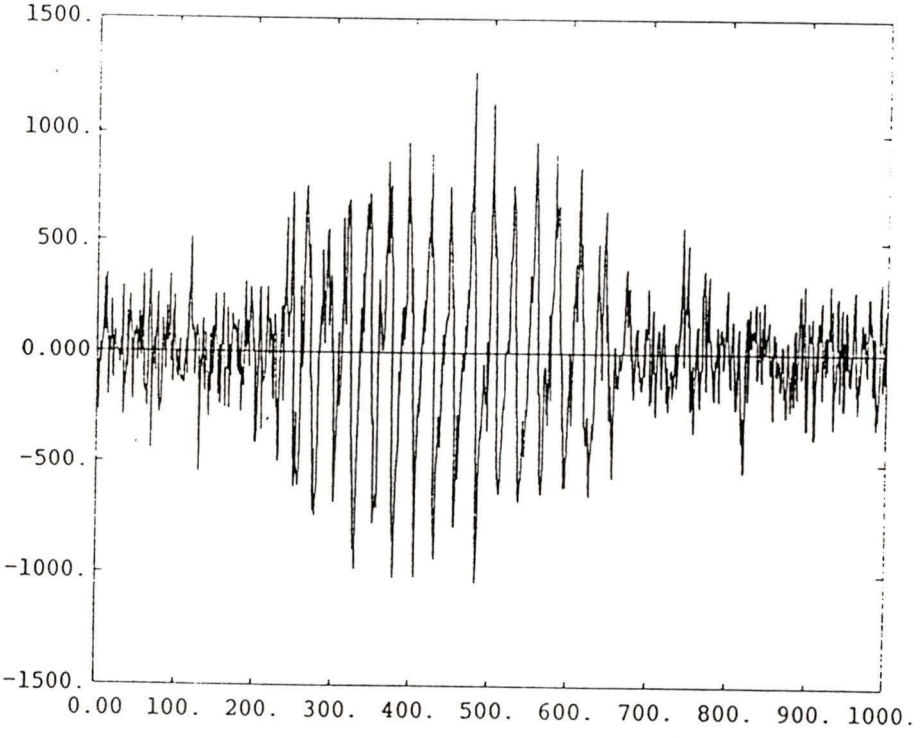


Figure 3.7: Filtered Signal for $\beta = 1.01,\ L = 6,\ n = 2$

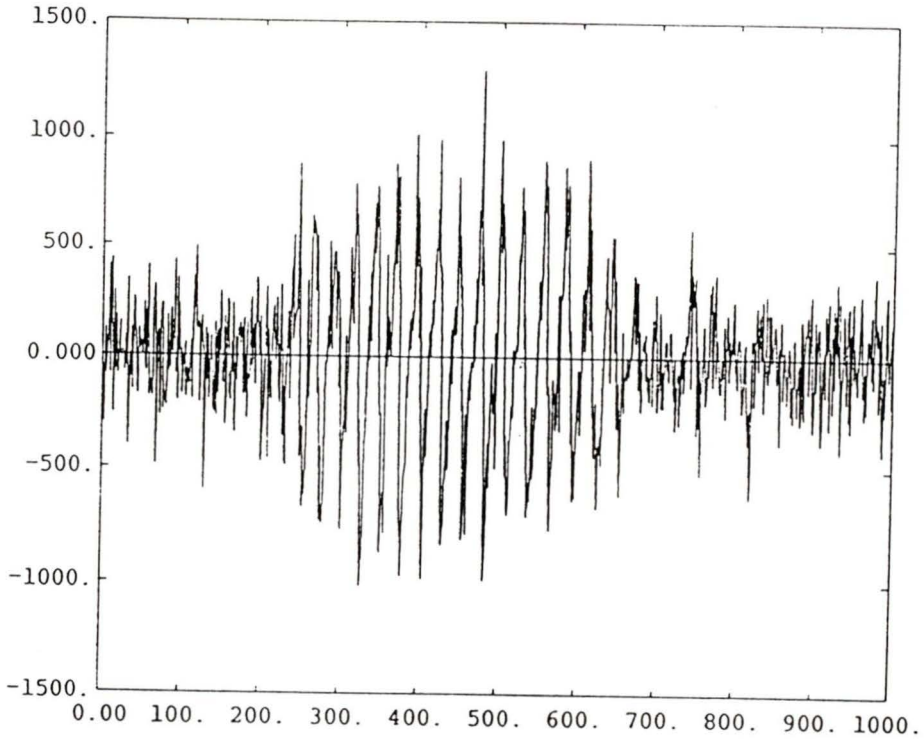


Figure 3.8: Filtered Signal for $\beta = 1.01,\ L = 6,\ n = 3$

Chapter 4

Comparison of Kalman Filter and Finite Memory Filter

4.1 Introduction

There are many different approaches to the problem of estimating parameters or states from observed data. In this chapter, two estimation techniques, Kalman filter and finite memory filter, are applied to the same unforced dynamical system. Two different cases are considered. In the first the noise is gaussian while in the second it has Rayleigh distribution. The results from Kalman filtering and finite memory filtering are compared and the criteria used are (1) computation time (2) relative square error between the estimated value and the real value from sampling points t_0 to t_N. The results of the comparison show that the Kalman filter is better than the finite memory filter in the presence of Gaussian noise while the finite memory filter is better for nongaussian noise.

This chapter is organized as follows. The problem statement is given in the following section. Next, the Kalman filter solution is presented in Section 4.3. In Section 4.4, the finite memory filter approach is applied to the same problem. The comparison results are shown in Section 4.5.

4.2 Problem Statement

A vector of measured (observed) data z is given, it is known that

$$z_i = at_i + b + noise \tag{4.1}$$

for some a and b, where z_i is the ith element of z and the measurements are corrupted with Gaussian white noise with all measurement errors uncorrelated. The goal is to obtain an estimate of $\hat{z}_k$ based on the sequence of the measurements z_0, z_1, z_2, ..., z_N with the observation window $w = 3$ (here $w = L+1$). The relationship between the real values and the measured values can be depicted in Fig. 4.1.

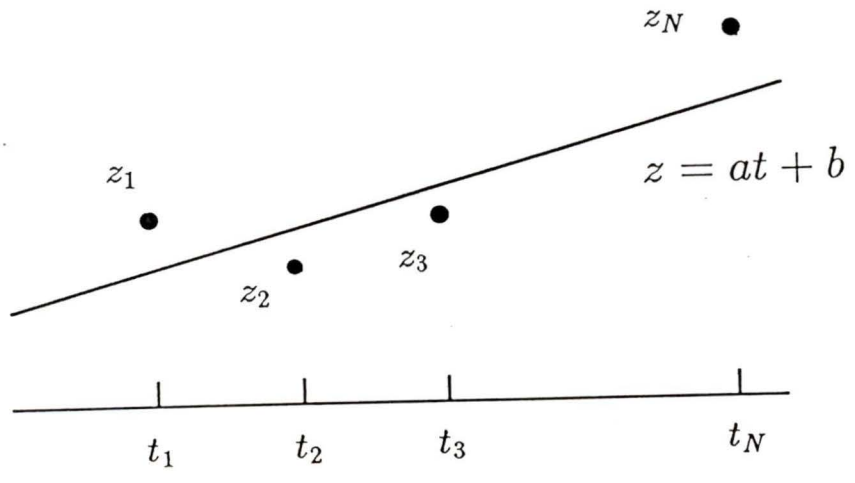


Figure 4.1: The Relationship between Real Values and Measured Values

4.3 Kalman Filter Approach

Consider a Kalman filter for estimating the two constants a and b. The system can be modelled as a linear difference equation

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_{k+1/k}\mathbf{x}_k + \mathbf{w}_k \tag{4.2}$$

and the observation model can be described as

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{4.3}$$

where

$$\mathbf{x}_k = \begin{bmatrix} a_k \\ b_k \end{bmatrix} \tag{4.4}$$

$$\mathbf{z}_k = \begin{bmatrix} z_{k-2} \\ z_{k-1} \\ z_k \end{bmatrix} \tag{4.5}$$

$$\mathbf{v}_k = \begin{bmatrix} v_{k-2} \\ v_{k-1} \\ v_k \end{bmatrix} \tag{4.6}$$

$$\mathbf{\Phi}_{k+1/k} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.7}$$

$$\mathbf{H}_k = \begin{bmatrix} k-2 & 1 \\ k-1 & 1 \\ k & 1 \end{bmatrix} \tag{4.8}$$

The statistics of the system is described by the following equations

$$\mathbf{w}_k = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.9}$$

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T] = 0 \tag{4.10}$$

$$\mathbf{R}_k = E[\mathbf{v}_k \mathbf{v}_k^T] = \sigma^2 \mathbf{I} \tag{4.11}$$

where the error variance $\sigma^2 = 1.15$.

The system is an unforced dynamical system since $\mathbf{w} = 0$. As discussed in Section 2.3.3, the initial condition of this system is given by

$$\hat{\mathbf{x}}_{2/1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.12}$$

and

$$\mathbf{P}_{2/1} = \infty \mathbf{I} \tag{4.13}$$

Obviously, the conventional Kalman filter algorithm discussed in Section 2.3.3 can not be used since it leads to indeterminant form. The alternative algorithm can be applied in this case, and $(\mathbf{P}_{2/-1})^{-1}$ rather than $\mathbf{P}_{2/-1}$ will be used in Eq.(2.44) which does not lead to any indeterminant form.

The alternative Kalman filter algorithm used is illustrated here with the procedure for the first 3 measurements.

- Step 1 Store the filter state $[\hat{\mathbf{x}}_{2/1},\ \mathbf{P}_{2/1}]$

- Step 2 Compute the updated error covariance matrix $\mathbf{P}_{2/2}$

$$\begin{aligned} \mathbf{P}_{2/2}^{-1} &= \mathbf{P}_{2/1}^{-1} + \mathbf{H}_2^T \mathbf{R}_2^{-1} \mathbf{H}_2 \\ &= \begin{bmatrix} 4.36 & 2.62 \\ 2.62 & 2.62 \end{bmatrix} \end{aligned} \tag{4.14}$$

$$\mathbf{P}_{2/2} = \begin{bmatrix} 0.57 & -0.57 \\ -0.57 & 0.96 \end{bmatrix} \tag{4.15}$$

- Step 3 Compute the filter gain matrix $\mathbf{K}_2$

$$\begin{aligned}\mathbf{K}_2 &= \mathbf{P}_{2/2}\mathbf{H}_2^T\mathbf{R}_2^{-1} \\ &= \begin{bmatrix} -0.50 & 0.00 & 0.50 \\ 0.83 & 0.33 & -0.17 \end{bmatrix}\end{aligned} \tag{4.16}$$

- Step 4 Update estimate $\hat{\mathbf{x}}_{2/2}$

$$\begin{aligned}\hat{\mathbf{x}}_{2/2} &= \hat{\mathbf{x}}_{2/1} + \mathbf{K}_2(\mathbf{z}_2 - \mathbf{H}_2\hat{\mathbf{x}}_{2/1}) \\ &= \begin{bmatrix} -0.50z_0 + 0.50z_2 \\ 0.83z_0 + 0.33z_1 - 0.17z_2 \end{bmatrix}\end{aligned} \tag{4.17}$$

- Step 5 Project ahead

$$\begin{aligned}\hat{\mathbf{x}}_{3/2} &= \mathbf{\Phi}_{3/2}\hat{\mathbf{x}}_{2/2} \\ &= \begin{bmatrix} -0.50z_0 + 0.50z_2 \\ 0.83z_0 + 0.33z_1 - 0.17z_2 \end{bmatrix}\end{aligned} \tag{4.18}$$

$$\begin{aligned}\mathbf{P}_{3/2} &= \mathbf{\Phi}_{3/2}\mathbf{P}_{2/2}\mathbf{\Phi}_{3/2}^T + \mathbf{Q}_2 \\ &= \mathbf{P}_{2/2}\begin{bmatrix} 0.57 & -0.57 \\ -0.57 & 0.96 \end{bmatrix}\end{aligned} \tag{4.19}$$

- Step 6 Invert $\mathbf{P}_{3/2}$

- Step 7 Set $k = k + 1$ and return to Step 1

Thus, from the first three measurements z_0, z_1, z_2, the estimate of the two constants a and b are obtained as

$$\hat{a}_{2/2} = -0.50z_0 + 0.50z_2 \tag{4.20}$$

$$\hat{b}_{2/2} = 0.83z_0 + 0.33z_1 - 0.17z_2 \tag{4.21}$$

In other words, with the presence of the noise, the linear equation

$$z(t) = at + b \tag{4.22}$$

in the interval t_0, t_1, t_2 can be approximated by

$$\hat{z}(t) = (-0.50z_0 + 0.50z_2)t + (0.83z_0 + 0.33z_2 - 0.17z_2) \tag{4.23}$$

Now, suppose the new measurement z_3 is received. Then if we follow the same step as we did before, it is easy to obtain the new Kalman filter gain matrix $\mathbf{K}_3$, the updated state $\hat{\mathbf{x}}_{3/3}$ and the updated error covariance matrix $\mathbf{P}_{3/3}$. This procedure can be continued recursively. At the last sampling point t_{99}, the estimate of two constants a and b are obtained as

$$\hat{\mathbf{x}}_{99/99} = \begin{bmatrix} \hat{a}_{99/99} \\ \hat{b}_{99/99} \end{bmatrix} = \begin{bmatrix} 2.00 \\ 15.08 \end{bmatrix} \tag{4.24}$$

4.4 Finite Memory Filters

4.4.1 Nonrecursive Algorithm

The estimate of $z(k)$ in the interval $(k - L)$ to k can be modelled by

$$\hat{z}(k - L + i) = a_0(k) + a_1(k)i \qquad i = 0,\ 1,\ 2,\ \ldots,\ L \tag{4.25}$$

where $a_1(k)$ refers to a while $a_0(k)$ refers to b. In order to determine $a_0(k)$ and $a_1(k)$, the following performance index

$$E^2(k) = \sum_{i=0}^{L} [a_0(k) + a_1(k)i - z(k-L+i)]^2 \beta^{i-L} \tag{4.26}$$

is minimized with respect to $a_0(k)$ and $a_1(k)$ where β is the constant weighting coefficient. Differentiating E with respect to $a_0(k)$ and $a_1(k)$ and setting the result to zero as discussed in Section 3.1 leads to the following linear equation

$$\mathbf{Ba} = \mathbf{c} \tag{4.27}$$

where

$$\mathbf{a} = \begin{bmatrix} a_0(k) \\ \\ a_1(k) \end{bmatrix} \tag{4.28}$$

$$\mathbf{B} = \begin{bmatrix} \sum_{i=0}^{L} \beta^i & \sum_{i=0}^{L} i\beta^i \\ \\ \sum_{i=0}^{L} i\beta^i & \sum_{i=0}^{L} i^2\beta^i \end{bmatrix} \tag{4.29}$$

$$\mathbf{c} = \begin{bmatrix} \sum_{i=0}^{L} \beta^i z(k-L+i) \\ \\ \sum_{i=0}^{L} i\beta^i z(k-L+i) \end{bmatrix} \tag{4.30}$$

Similarly as in Section 3.2, the equation for the nonrecursive finite memory filter can be obtained as

$$\begin{aligned} \hat{z}(k) &= m_1 c_1(k) + m_2 c_2(k) \\ &= \sum_{i=0}^{L} (m_1 + im_2)\beta^i z(k-L+i) \end{aligned} \tag{4.31}$$

which gives an estimate of $z(k)$ based on the measurements $z(0)$, $z(1)$, ..., $z(L)$.

4.4.2 Recursive Algorithm

The recursive algorithm can be obtained similarly as in Section 3.3 from the recursive equation Eq.(3.18)

$$\begin{aligned} c_1(k+1) &= \sum_{i=k+1-L}^{k+1} \beta^{i-k+L-1} z(i) \\ &= \frac{1}{\beta}[-z(k-L) + \sum_{i=k-L}^{k} \beta^{i-k+L} z(i)] + \beta^L z(k+1) \end{aligned} \tag{4.32}$$

$$\begin{aligned} c_2(k+1) &= \sum_{i=k+1-L}^{k+1} (i-k+L)\beta^{i-k+L-1} z(i) \\ &= \frac{1}{\beta}[\sum_{k-L}^{k} (i-k+L)\beta^{i-k+L} z(i) \\ &- \sum_{i=k-L}^{k} \beta^{i-k+L} z(i) + z(k-L)] + L\beta^L z(k+1) \end{aligned} \tag{4.33}$$

$c_1(k+1)$ and $c_2(k+1)$ can be rewritten as

$$c_1(k+1) = \frac{1}{\beta} c_1(k) - \frac{1}{\beta} z(k-L) + \beta^L z(k+1) \tag{4.34}$$

$$c_2(k+1) = \frac{1}{\beta} c_2(k) - \frac{1}{\beta} c_1 k + \frac{1}{\beta} z(k-L) + L\beta^L z(k+1) \tag{4.35}$$

Therefore, $\hat{z}(k+1)$ can be expressed by the following equation

$$\begin{aligned} \hat{z}(k+1) &= m_1 c_1(k+1) + m_2 c_2(k+1) \\ &= \frac{1}{\beta}\hat{z}(k) - \frac{m_2}{\beta} c_1(k) + (\frac{m_2}{\beta} - \frac{m_1}{\beta}) z(k-L) \\ &+ \beta^L (m_1 + Lm_2) z(k+1) \end{aligned} \tag{4.36}$$

Eq.(4.34) and Eq.(4.36) are the recursive equation of the finite memory filter. They allow the computation of $\hat{z}(k+1)$ recursively. Furthermore, the recursive algorithm can be expressed by the following state space representation

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}_1 z(k-L) + \mathbf{b}_2 z(k+1) \tag{4.37}$$

$$\hat{z}(k) = \mathbf{h}^T \mathbf{x}(k) \tag{4.38}$$

where

$$\mathbf{x(k)} = \begin{bmatrix} c_1(k) \\ \hat{z}(k) \end{bmatrix} \tag{4.39}$$

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\beta} & 0 \\ -\frac{m_2}{\beta} & \frac{1}{\beta} \end{bmatrix} \tag{4.40}$$

$$\mathbf{b_1} = \begin{bmatrix} -\frac{1}{\beta} \\ \frac{m_2 - m_1}{\beta} \end{bmatrix} \tag{4.41}$$

$$\mathbf{b_2} = \begin{bmatrix} \beta^L \\ \beta^L(m_1 + Lm_2) \end{bmatrix} \tag{4.42}$$

$$\mathbf{h} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.43}$$

4.5 Comparison Results

4.5.1 Computation Time

The basic equation of the Kalman filter consists of four matrix recursion relations. By using Kalman filter, the computational effort, that is, the total

Equations	Multiplications	Additions
$\mathbf{P}_k = \mathbf{P}_{k/k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k/k-1}$	20	16
$\mathbf{K}_k = \mathbf{P}_{k/k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k/k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$	55	45
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k/k-1})$	12	13
$\mathbf{P}_{k/k-1} = \mathbf{\Phi}_{k/k-1}\mathbf{P}_k\mathbf{\Phi}_{k/k-1}^T$	16	8
Total	103	82

Table 4.1: Kalman Filter Computation Breakdown

multiplications and additions used for each step is summarized in Table 4.1. The inversion of the matrix $(\mathbf{H}_k\mathbf{P}_{k/k-1}\mathbf{H}_k^T + \mathbf{R}_k)$ is handled by the single division scheme [28]. The size of the matrices for the problem considered depends on w and is given by

$$\begin{array}{ll} \mathbf{\Phi} & 2 \times 2 \ matrix \\ \mathbf{P} & 2 \times 2 \ matrix \\ \mathbf{K} & 2 \times 3 \ matrix \\ \mathbf{R} & 3 \times 3 \ matrix \\ \mathbf{H} & 3 \times 1 \ vector \\ \hat{\mathbf{x}} & 2 \times 1 \ vector \\ \mathbf{z} & 3 \times 1 \ vector \end{array}$$

The finite memory filter,(either nonrecursive or recursive), on the other hand requires significantly less computations. This can be seen from Table 4.2 where the number of operation for the recursive and nonrecursive finite memory filters are given for different L. The number of operations for the recursive filter is constant and independent of L as shown in Section 3.3.

4.5.2 Filter Performance

The performance of the Kalman filter and recursive finite memory filter are compared by filtering the signal given by

$$y(k) = 2k + 15 + w(k) \tag{4.44}$$

L	Nonrecursive		Recursive	
	additions	multiplications	additions	multiplications
3	4	5	5	6
8	8	9	5	6
14	14	15	5	6

Table 4.2: Computational Effort for Nonrecursive and Recursive Filters

where $w(k)$ represents noise. In the first experiment $w(k)$ was chosen as a Gaussian noise with variance $\sigma_g^2 = 1.15$ while in the second experiment $w(k)$ is noise with Rayleigh distribution given by

$$p(z) = \frac{z}{\sigma_r^2} exp[-\frac{z^2}{2\sigma_r^2}] \qquad z \geq 0 \tag{4.45}$$

where $\sigma_r^2 = 2\sigma_g^2 = 2.30$.

In the first experiment the Kalman filter gives good results. The output converges very close to the exact values after approximately 35 points (within an error of 5%). The finite memory filter used with $\beta = 1.01$ and $L = 9$ gives approximately twice the error obtained with the Kalman filter. The comparison result is illustrated in Table 4.3 for different w where the relative square error is defined by Eq.(3.43). It can be easily seen that the performance of the Kalman filter is superior to that of the finite memory filter for Gaussian noise.

In the second experiment the finite memory filter gives almost the same result as that of Kalman filter. Fig. 4.2 and Fig. 4.3 show the relative errors between the original signal and the estimated one. In this case where the noise has a Rayleigh distribution, the relative square error for both filters is almost the same. However, the finite memory has significantly lower computational requirements. The performance of the two filters in the presence of Rayleigh noise are compared in Table 4.4.

Filter Type	Relative Errors		
	$w = 4$	$w = 8$	$w = 12$
Kalman Filter	28.88×10^{-4}	25.99×10^{-4}	24.48×10^{-4}
Finite Memory Filter	70.79×10^{-4}	54.60×10^{-4}	44.97×10^{-4}

Table 4.3: Comparison of Relative Square Errors for Gaussian Noise

Filter Type	Relative Errors		
	$w = 4$	$w = 8$	$w = 12$
Kalman Filter	1.11×10^{-2}	1.08×10^{-2}	1.07×10^{-2}
Finite Memory Filter	1.13×10^{-2}	1.06×10^{-2}	1.03×10^{-2}

Table 4.4: Comparison of Relative Square Errors for Rayleigh Noise

These two experiments confirm that the Kalman filter gives better results in the Gaussian noise case while the finite memory filter gives almost the same results but saves significant computation time in the case of nongaussian noise.

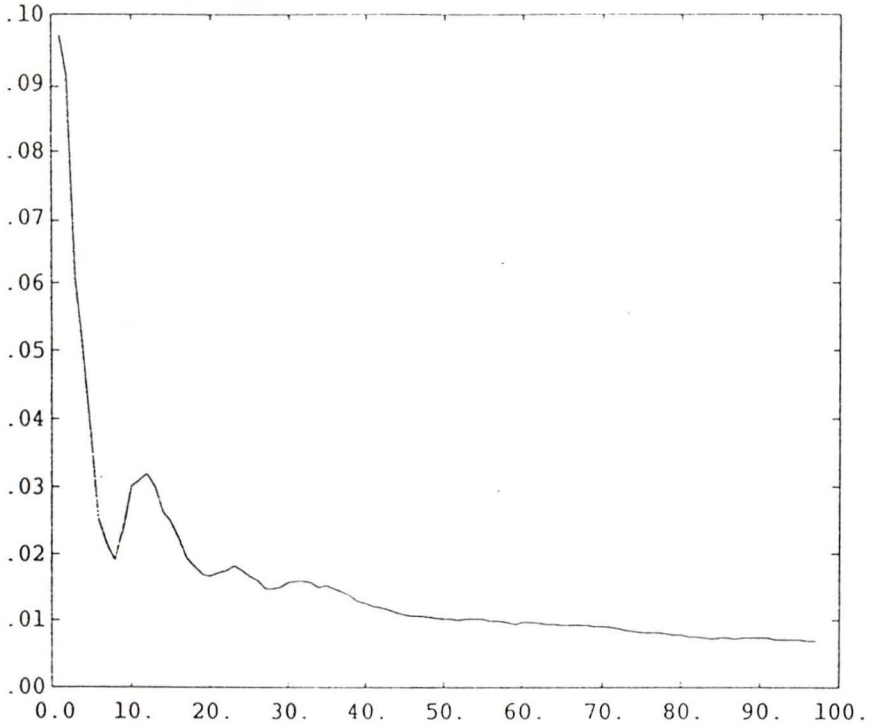


Figure 4.2: Relative Error Between the Original Signal and Estimated One using Kalman Filter for $L = 3$

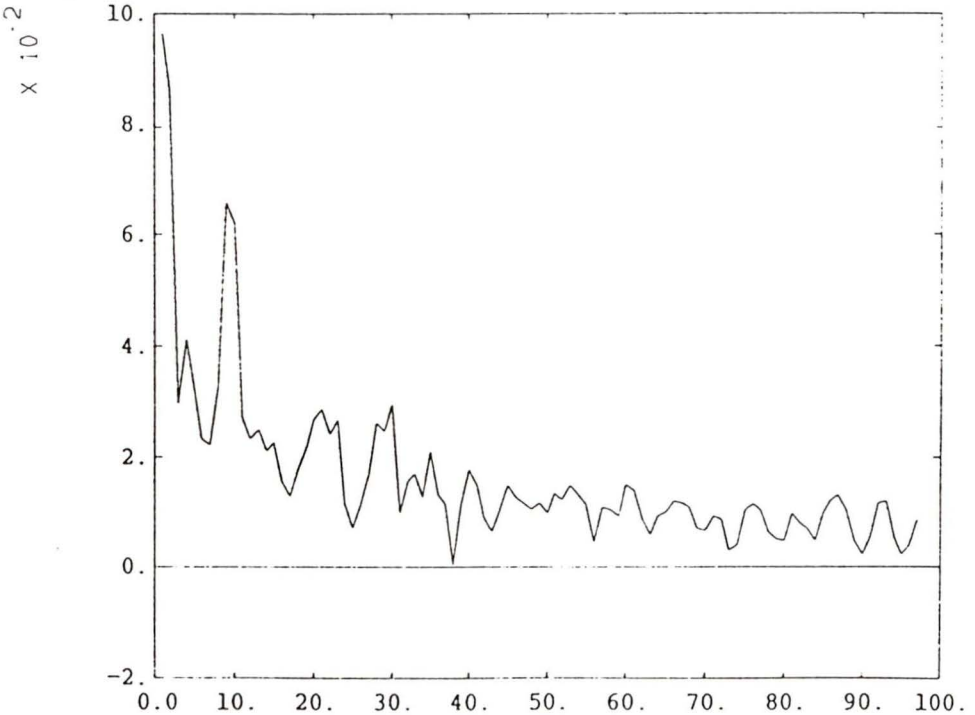


Figure 4.3: Relative Error Between the Original Signal and Estimated One using Finite Memory Filter for $L = 3$

Chapter 5

Recursive Implementation of 2-Dimensional Finite Memory Filters

5.1 Introduction

The advantages of finite memory filters when dealing with nongaussian noise has prompted the development of such a filter for 2-D applications. In [19] a 2-D finite memory filter was developed and applied to an image processing application. This is due to the fact that in many image processing applications, noise is rather spike noise and/or film-grain noise rather than Gaussian [30], and therefore 2-D finite memory filters are well suited for these types of applications. In [19] a surface is approximated at each step by a plane of appropriate orientation. In this chapter, a second order 2-D polynomial is used to approximate the same surface. This allows a better approximation of the original 2-D function than the one obtained using only a plane. The nonrecursive algorithm of 2-D finite memory filter is developed in section 5.2. The realization of the recursive algorithm is provided in section 5.3. A comparison of the performance of the recursive and nonrecursive filter is presented in section 5.4.

5.2 Nonrecursive Filter

Consider a 2-D continuous signal $y(t_1, t_2)$ sampled rectangularly with sampling rate T_1 in one direction and T_2 in another direction. It is required to estimate the sampled 2-D signal $y(i,j)$ using a linear model, given noisy measured values of $y(i,j)$ in an observation window of length L_1T_1 in the direction of i (horizontal) and L_2T_2 in the direction of j (vertical). $\hat{y}(k,l)$, the estimate of $y(k,l)$ in the window with vertices $(k-L_1, l-L_2), (k, l-L_2), (k-L_1, l), (k,l)$ will be modeled by a 2-D second order polynomial

$$\begin{aligned}\hat{y}(k-L_1+i, l-L_2+j) &= \\ \left[1,\ iT_1,\ i^2T_1\right] &\begin{bmatrix} a_{11}(k,l) & a_{12}(k,l) & a_{13}(k,l) \\ a_{21}(k,l) & a_{22}(k,l) & a_{23}(k,l) \\ a_{31}(k,l) & a_{32}(k,l) & a_{33}(k,l) \end{bmatrix} \begin{bmatrix} 1 \\ jT_2 \\ j^2T_2 \end{bmatrix}\end{aligned} \tag{5.1}$$

Without loss of generality, T_1 and T_2 are normalized to unity. The $a_{i,j}(i,j = 1,2,3)$ are obtained by minimizing

$$\begin{aligned}E &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2}[\hat{y}(k-L_1+i, l-L_2+j) \\ &\quad -y(k-L_1+i, l-L_2+j)]^2\beta_1^{i-L_1}\beta_2^{j-L_2}\end{aligned} \tag{5.2}$$

w.r.t. $a_{i,j}(i,j = 1,2,3)$. This leads to a set of linear equations

$$B\mathbf{a} = \mathbf{c} \tag{5.3}$$

where $\mathbf{B}$ is given in the Appendix A, $\mathbf{a}$ is given by

$$\begin{aligned}\mathbf{a} &= [a_{11}(k,l), a_{21}(k,l), a_{31}(k,l), a_{12}(k,l), \\ &\quad a_{22}(k,l), a_{32}(k,l), a_{13}(k,l), a_{23}(k,l), a_{33}(k,l)]^T\end{aligned} \tag{5.4}$$

and $\mathbf{c}$ is given by

$$\begin{aligned}\mathbf{c}(k,l) &= [c_{0,0}(k,l), c_{1,0}(k,l), c_{2,0}(k,l), c_{0,1}(k,l),\end{aligned}$$

$$c_{1,1}(k,l), c_{2,1}(k,l), c_{0,2}(k,l), c_{1,2}(k,l), c_{2,2}(k,l)]^T \tag{5.5}$$

where

$$c_{m,n}(k,l) = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^m j^n \beta_1^i \beta_2^j y(k-L_1+i, l-L_2+j) \quad m,n = 0,1,2 \tag{5.6}$$

From Eq.(5.1) (T_1 and T_2 are assumed to be 1) the estimate of $y(k,l)$ is given by

$$\begin{aligned}
\hat{y}(k,l) &= a_{11}(k,l) + a_{21}(k,l)L_1 + a_{31}(k,l)L_1^2 \\
&+ a_{12}(k,l)L_2 + a_{22}(k,l)L_1L_2 + a_{32}(k,l)L_1^2L_2 \\
&+ a_{13}(k,l)L_2^2 + a_{23}(k,l)L_1L_2^2 + a_{33}(k,l)L_1^2L_2^2 \\
&= \mathbf{a}^T\mathbf{p}
\end{aligned} \tag{5.7}$$

where

$$\mathbf{p} = [1, L_1, L_1^2, L_2, L_1L_2, L_1^2L_2, L_2^2, L_1L_2^2, L_1^2L_2^2]^T \tag{5.8}$$

Solving for Eq.(5.3) for $\mathbf{a}$ and substituting into Eq.(5.7), we obtain

$$\hat{y}(k,l) = \mathbf{mc}(k,l) \tag{5.9}$$

where $\mathbf{m} = [m_1, m_2, \ldots, m_9] \in R^9$ is the solution of $B\mathbf{m} = \mathbf{p}$, which is not a function of (k,l). Eq.(5.9) represents the nonrecursive finite memory filter.

5.3 Implementation of Recursive Filter

The nonrecursive filter requires a great deal of computation effort due to the fact that the sums $c_{m,n}, (m,n = 0,1,2)$ are evaluated at each instant (k,l). In order to reduce the amount of computation required, these sums can be evaluated recursively by noting that every time a set of observations is added

another set of observations is deleted. This yields to the following recursive relations for $c_{m,n}(k,l)(m,n=0,1,2)$ (see Appendix B for the derivations).

$$\begin{aligned} c_{0,0}(k+1,l+1) &= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\ &= \beta_1^{-1}c_{0,0}(k,l+1)+\beta_2^{-1}c_{0,0}(k+1,l)-\beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\ &+ \beta_1^{-1}\beta_2^{-1}y(k-L_1,l-L_2)-\beta_1^{-1}\beta_2^{L_2}y(k-L_1,l+1) \\ &- \beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2)+\beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1) \end{aligned} \tag{5.10}$$

$$\begin{aligned} c_{1,0}(k+1,l+1) &= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(i-k+L_1-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\ &= \beta_1^{-1}c_{1,0}(k,l+1)+\beta_2^{-1}c_{1,0}(k+1,l)-\beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) \\ &- \beta_1^{-1}c_{0,0}(k,l+1)+\beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\ &+ L_1\beta_1^{L_1}[\beta_2^{L_2}y(k+1,l+1)-\beta_2^{-1}y(k+1,l-L_2)] \\ &- \beta_1^{-1}[\beta_2^{L_2}y(k-L_1,l+1)-\beta_2^{-1}y(k-L_1,l-L_2)] \end{aligned} \tag{5.11}$$

$$\begin{aligned} c_{2,0}(k+1,l+1) &= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(i-k+L_1-1)^2\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\ &= \beta_1^{-1}c_{2,0}(k,l+1)+\beta_2^{-1}c_{2,0}(k+1,l)-\beta_1^{-1}\beta_2^{-1}c_{2,0}(k,l) \\ &- 2\beta_1^{-1}c_{1,0}(k,l+1)+2\beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) \\ &+ \beta_1^{-1}c_{0,0}(k,l+1)-\beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\ &+ L_1^2\beta_1^{L_1}[\beta_2^{L_2}y(k+1,l+1)-\beta_2^{-1}y(k+1,l-L_2)] \\ &- \beta_1^{-1}[\beta_2^{L_2}y(k-L_1,l+1)-\beta_2^{-1}y(k-L_1,l-L_2)] \end{aligned} \tag{5.12}$$

$$c_{0,1}(k+1,l+1) = \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(j-l+L_2-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j)$$

$$
\begin{aligned}
&= \beta_1^{-1}c_{0,1}(k,l+1) + \beta_2^{-1}c_{0,1}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{0,1}(k,l) \\
&- \beta_2^{-1}c_{0,0}(k+1,l) + \beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
&+ L_2\beta_2^{L_2}[\beta_1^{L_1}y(k+1,l+1) - \beta_1^{-1}y(k-L_1,l+1)] \\
&- \beta_2^{-1}[\beta_1^{L_1}y(k+1,l-L_2) - \beta_1^{-1}y(k-L_1,l-L_2)] \qquad (5.13)
\end{aligned}
$$

$$
\begin{aligned}
c_{1,1}(k+1,l+1) &= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(i-k+L_1-1)(j-k+L_2-1) \\
&\quad \beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&= \beta_1^{-1}c_{1,1}(k,l+1) + \beta_2^{-1}c_{1,1}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{1,1}(k,l) \\
&- \beta_1^{-1}c_{0,1}(k,l+1) + \beta_1^{-1}\beta_2^{-1}c_{0,1}(k,l) \\
&- \beta_2^{-1}c_{1,0}(k+1,l) + \beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) - \beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
&+ L_1\beta_1^{L_1}[L_2\beta_2^{L_2}y(k+1,l+1) + \beta_2^{-1}y(k+1,l-L_2)] \\
&+ \beta_1^{-1}[L_2\beta_2^{L_2}y(k-L_1,l+1) + \beta_2^{-1}y(k-L_1,l-L_2)]
\end{aligned}
\qquad (5.14)
$$

$$
\begin{aligned}
c_{2,1}(k+1,l+1) &= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(i-k+L_1-1)^2(j-l+L_2-1) \\
&\quad \beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&= \beta_1^{-1}c_{2,1}(k,l+1) + \beta_2^{-1}c_{2,1}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{2,1}(k,l) \\
&- 2\beta_1^{-1}c_{1,1}(k,l+1) + 2\beta_1^{-1}\beta_2^{-1}c_{1,1}(k,l) \\
&+ \beta_1^{-1}c_{0,1}(k,l+1) - \beta_1^{-1}\beta_2^{-1}c_{0,1}(k,l) \\
&- \beta_2^{-1}c_{2,0}(k+1,l) + \beta_1^{-1}\beta_2^{-1}c_{2,0}(k,l) \\
&- 2\beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) + \beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
&+ L_1^2\beta_1^{L_1}[L_2\beta_2^{L_2}y(k+1,l+1) - \beta_2^{-1}y(k+1,l-L_2)]
\end{aligned}
$$

$$- \beta_1^{-1}[L_2\beta_2^{L_2}y(k-L_1,l+1) - \beta_2^{-1}y(k-L_1,l-L_2)] \tag{5.15}$$

$$\begin{aligned}
c_{0,2}(k+1,l+1) &= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(j-l+L_2-1)^2\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&= \beta_1^{-1}c_{0,2}(k,l+1) + \beta_2^{-1}c_{0,2}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{0,2}(k,l) \\
&- 2\beta_2^{-1}c_{0,1}(k+1,l) + 2\beta_1^{-1}\beta_2^{-1}c_{0,1}(k,l) \\
&+ \beta_2^{-1}c_{0,0}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
&+ L_2^2\beta_2^{L_2}[\beta_1^{L_1}y(k+1,l+1) - \beta_1^{-1}y(k-L_1,l+1)] \\
&- \beta_2^{-1}[\beta_1^{L_1}y(k+1,l-L_2) - \beta_1^{-1}y(k-L_1,l-L_2)]
\end{aligned} \tag{5.16}$$

$$\begin{aligned}
c_{1,2}(k+1,l+1) &= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(i-k+L_1-1)(j-l+L_2-1)^2 \\
&\quad \beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&= \beta_1^{-1}c_{1,2}(k,l+1) + \beta_2^{-1}c_{1,2}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{1,2}(k,l) \\
&- 2\beta_2^{-1}c_{1,1}(k+1,l) + 2\beta_1^{-1}\beta_2^{-1}c_{1,1}(k,l) \\
&+ \beta_2^{-1}c_{1,0}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) \\
&- \beta_1^{-1}c_{0,2}(k,l+1) + \beta_1^{-1}\beta_2^{-1}c_{0,2}(k,l) \\
&- 2\beta_1^{-1}\beta_2^{-1}c_{0,1}(k,l) + \beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
&+ L_2^2\beta_2^{L_2}[L_1\beta_1^{L_2}y(k+1,l+1) - \beta_1^{-1}y(k-L_1,l+1)] \\
&- \beta_2^{-1}[L_1\beta_1^{L_1}y(k+1,l-L_2) - \beta_1^{-1}y(k-L_1,l-L_2)]
\end{aligned} \tag{5.17}$$

$$c_{2,2}(k+1,l+1) = \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(i-k+L_1-1)^2(j-l+L_2-1)^2$$

$$
\begin{aligned}
& \beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
= \;& \beta_1^{-1}c_{2,2}(k,l+1)+\beta_2^{-1}c_{2,2}(k+1,l)-\beta_1^{-1}\beta_2^{-1}c_{2,2}(k,l) \\
- \;& 2\beta_1^{-1}c_{1,2}(k,l+1)-2\beta_2^{-1}c_{2,1}(k+1,l) \\
- \;& 2\beta_1^{-1}\beta_2^{-1}c_{1,2}(k,l)+2\beta_1^{-1}\beta_2^{-1}c_{2,1}(k,l) \\
+ \;& \beta_1^{-1}c_{0,2}(k,l+1)-\beta_1^{-1}\beta_2^{-1}c_{0,2}(k,l) \\
+ \;& \beta_2^{-1}c_{2,0}(k+1,l)-\beta_1^{-1}\beta_2^{-1}c_{2,0}(k,l) \\
- \;& 4\beta_1^{-1}\beta_2^{-1}c_{1,1}(k,l)+2\beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) \\
+ \;& 2\beta_1^{-1}\beta_2^{-1}c_{0,1}(k,l)-\beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
+ \;& L_1^2\beta_1^{L_1}[L_2^2\beta_2^{L_2}y(k+1,l+1)-\beta_2^{-1}y(k+1,l-L_2)] \\
- \;& \beta_1^{-1}[L_2^2\beta_2^{L_2}y(k-L_1,l+1)-\beta_2^{-1}y(k-L_1,l-L_2)]
\end{aligned}
\tag{5.18}
$$

Substituting $c_{m,n}(k+1,l+1)(m,n=0,1,2)$ from Eqs.(5.10)–(5.18) into the nonrecursive filter equation we obtain a recursive equation for $\hat{y}(k+1,l+1)$

$$
\begin{aligned}
\hat{y}(k+1,l+1) &= \mathbf{mc}(k+1,l+1) \\
&= \beta_1^{-1}\hat{y}(k,l+1)+\beta_2^{-1}\hat{y}(k+1,l)-\beta_1^{-1}\beta_2^{-1}\hat{y}(k,l) \\
&- (m_2-m_3)\beta_1^{-1}c_{0,0}(k,l+1)-(m_4-m_7)\beta_2^{-1}c_{0,0}(k+1,l) \\
&+ (m_2-m_3+m_4-m_5+m_6-m_7+m_8-m_9) \\
&\quad \beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
&- 2m_3\beta_1^{-1}c_{1,0}(k,l+1)-(m_5-m_8)\beta_2^{-1}c_{1,0}(k+1,l) \\
&+ (2m_3+m_5-2m_6-m_8+2m_9)\beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) \\
&- 2m_7\beta_2^{-1}c_{0,1}(k+1,l)-(m_5-m_6)\beta_1^{-1}c_{0,1}(k+1,l) \\
&+ (2m_7+m_5-2m_8-m_6+2m_9)\beta_1^{-1}\beta_2^{-1}c_{0,1}(k,l) \\
&- (m_6-m_9)\beta_2^{-1}c_{2,0}(k+1,l)+(m_6-m_9)\beta_1^{-1}\beta_2^{-1}c_{2,0}(k,l)
\end{aligned}
$$

$$
\begin{aligned}
- \; & (m_8 - m_9)\beta_1^{-1} c_{0,2}(k, l+1) + (m_8 - m_9)\beta_1^{-1}\beta_2^{-1} c_{0,2}(k, l) \\
- \; & 2m_6\beta_1^{-1} c_{1,1}(k, l+1) - 2m_8\beta_2^{-1} c_{1,1}(k+1, l) \\
+ \; & (2m_6 + 2m_8 - 4m_9)\beta_1^{-1}\beta_2^{-1} c_{1,1}(k, l) \\
- \; & 2m_9\beta_2^{-1} c_{2,1}(k+1, l) + 2m_9\beta_1^{-1}\beta_2^{-1} c_{2,1}(k, l) \\
- \; & 2m_9\beta_1^{-1} c_{1,2}(k, l+1) + 2m_9\beta_1^{-1}\beta_2^{-1} c_{1,2}(k, l) \\
+ \; & (m_1 - m_2 + m_3 - m_4 + m_5 - m_6 + m_7 - m_8 + m_9) \\
& \beta_1^{-1}\beta_2^{-1} y(k - L_1, l - L_2) \\
- \; & [(m_1 - m_2 + m_3) + L_2(m_4 - m_5 + m_6) \\
& + L_2^2(m_7 - m_8 + m_9)]\beta_1^{-1}\beta_2^{L_2} y(k - L_1, l + 1) \\
- \; & [(m_1 + L_1 m_2 + L_1^2 m_3) - (m_4 + L_1 m_5 + L_1^2 m_6) \\
& + (m_7 + L_1 m_8 + L_1^2 m_9)]\beta_1^{L_1}\beta_2^{-1} y(k + 1, l - L_2) \\
+ \; & (m_1 + L_1 m_2 + L_1^2 m_3 + L_2 m_4 + L_1 L_2 m_5 + L_1^2 L_2 m_6 \\
& + L_2^2 m_7 + L_1 L_2^2 m_8 + L_1^2 L_2^2 m_9)\beta_1^{L_1}\beta_2^{L_2} y(k + 1, l + 1)
\end{aligned}
\tag{5.19}
$$

From Eqs.(5.10)–(5.19) it can be seen that $\hat{y}(k, l)$ can be computed recursively using Eqs.(5.10)–(5.17) and Eq.(5.19). These nine equations are the state equations for the recursive finite memory filter. They can be written in the following form (Fornasini-Machesini's model) [29]

$$
\begin{aligned}
\mathbf{x}(k+1, l+1) &= \mathbf{A}_1\mathbf{x}(k, l+1) + \mathbf{A}_2\mathbf{x}(k+1, l) \\
&+ \mathbf{A}_3\mathbf{x}(k, l) + \mathbf{F}\mathbf{u}(k, l)
\end{aligned}
\tag{5.20}
$$

$$
\hat{y}(k, l) = \mathbf{h}^T\mathbf{x}(k, l) \tag{5.21}
$$

where

$$
\mathbf{x}(k, l) = [c_{0,0}(k, l), c_{1,0}(k, l), c_{2,0}(k, l), c_{0,1}(k, l)
$$

$$c_{1,1}(k,l), c_{2,1}(k,l), c_{0,2}(k,l), c_{1,2}(k,l), \hat{y}(k,l)]^T \tag{5.22}$$

and

$$\mathbf{A}_1 =$$

$$\beta_1^{-1} \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & & & & & & \\ -1 & 1 & 0 & & 0 & & & 0 & \\ 1 & -2 & 1 & & & & & & \\ \hline & & & 1 & 0 & 0 & & & \\ & 0 & & -1 & 1 & 0 & & 0 & \\ & & & 1 & -2 & 1 & & & \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ q_1 & -2m_3 & 0 & q_2 & -2m_6 & 0 & q_3 & -2m_9 & 1 \end{array}\right] \tag{5.23}$$

where

$$q_1 = -(m_2 - m_3) \tag{5.24}$$

$$q_2 = -(m_5 - m_6) \tag{5.25}$$

$$q_3 = -(m_8 - m_9) \tag{5.26}$$

$$\mathbf{A}_2 =$$

$$\beta_2^{-1} \left[\begin{array}{ccc|ccc|cc} 1 & 0 & 0 & & & & & \\ 0 & 1 & 0 & & 0 & & 0 & \\ 0 & 0 & 1 & & & & & \\ \hline -1 & 0 & 0 & 1 & 0 & 0 & & \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & -1 & 0 & 0 & 1 & & \\ \hline 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ r_1 & r_2 & r_3 & -2m_7 & -2m_8 & -2m_9 & 0 & 0 \end{array}\right] \tag{5.27}$$

where

$$r_1 = -(m_4 - m_7) \tag{5.28}$$

$$r_2 = -(m_5 - m_8) \tag{5.29}$$

$$r_3 = -(m_6 - m_9) \tag{5.30}$$

$$\mathbf{A}_3 = \beta_1^{-1}\beta_2^{-1} \left[\begin{array}{ccc|ccc|ccc} -1 & 0 & 0 & & & & & & \\ 1 & -1 & 0 & & 0 & & & 0 & \\ -1 & 2 & -1 & & & & & & \\ \hline 1 & 0 & 0 & -1 & 0 & 0 & & & \\ -1 & 1 & 0 & 1 & -1 & 0 & & 0 & \\ 1 & -2 & -1 & -1 & 2 & -1 & & & \\ \hline -1 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & -2 & 2 & 0 & 1 & -1 & 0 \\ s_1 & s_2 & s_3 & s_4 & s_5 & 2m_9 & s_6 & 2m_9 & -1 \end{array}\right] \tag{5.31}$$

where

$$s_1 = m_2 - m_3 + m_4 - m_5 + m_6 - m_7 + m_8 - m_9 \tag{5.32}$$

$$s_2 = 2m_3 + m_5 - 2m_6 - m_8 + 2m_9 \tag{5.33}$$

$$s_3 = m_6 - m_9 \tag{5.34}$$

$$s_4 = m_5 - m_6 + 2m_7 - 2m_8 + 2m_9 \tag{5.35}$$

$$s_5 = 2m_6 + 2m_8 - 4m_9 \tag{5.36}$$

$$s_6 = m_8 - m_9 \tag{5.37}$$

$$\mathbf{F} = \begin{bmatrix} \beta_1^{L_1}\beta_2^{L_2} & -\beta_1^{L_1}\beta_2^{-1} & -\beta_1^{-1}\beta_2^{L_2} & \beta_1^{-1}\beta_2^{-1} \\ L_1\beta_1^{L_1}\beta_2^{L_2} & -L_1\beta_1^{L_1}\beta_2^{-1} & \beta_1^{-1}\beta_2^{L_2} & -\beta_1^{-1}\beta_2^{-1} \\ L_1^2\beta_1^{L_1}\beta_2^{L_2} & -L_1^2\beta_1^{L_1}\beta_2^{-1} & -\beta_1^{-1}\beta_2^{L_2} & \beta_1^{-1}\beta_2^{-1} \\ L_2\beta_1^{L_1}\beta_2^{L_2} & \beta_1^{L_1}\beta_2^{-1} & -L_2\beta_1^{-1}\beta_2^{L_2} & -\beta_1^{-1}\beta_2^{-1} \\ L_1L_2\beta_1^{L_1}\beta_2^{L_2} & L_1\beta_1^{L_1}\beta_2^{-1} & L_2\beta_1^{-1}\beta_2^{L_2} & \beta_1^{-1}\beta_2^{-1} \\ L_1^2L_2\beta_1^{L_1}\beta_2^{L_2} & L_1^2\beta_1^{L_1}\beta_2^{-1} & -L_2\beta_1^{-1}\beta_2^{L_2} & -\beta_1^{-1}\beta_2^{-1} \\ L_2^2\beta_1^{L_1}\beta_2^{L_2} & -\beta_1^{L_1}\beta_2^{-1} & -L_2^2\beta_1^{-1}\beta_2^{L_2} & \beta_1^{-1}\beta_2^{-1} \\ L_1L_2^2\beta_1^{L_1}\beta_2^{L_2} & -L_1\beta_1^{L_1}\beta_2^{-1} & L_2^2\beta_1^{-1}\beta_2^{L_2} & -\beta_1^{-1}\beta_2^{-1} \\ f_{91}\beta_1^{L_1}\beta_2^{L_2} & -f_{92}\beta_1^{L_1}\beta_2^{-1} & -f_{93}\beta_1^{-1}\beta_2^{L_2} & f_{94}\beta_1^{-1}\beta_2^{-1} \end{bmatrix} \tag{5.38}$$

where

$$\begin{aligned} f_{91} &= m_1 + L_1m_2 + L_1^2m_3 + L_2m_4 + L_1L_2m_5 + L_1^2L_2m_6 \\ &\quad + L_2^2m_7 + L_1L_2^2m_8 + L_1^2L_2^2m_9 \end{aligned} \tag{5.39}$$

$$\begin{aligned} f_{92} &= -(m_1 + L_1m_2 + L_1^2m_3) + (m_4 + L_1m_5 + L_1^2m_6) \\ &\quad -(m_7 + L_1m_8 + L_1^2m_9) \end{aligned} \tag{5.40}$$

$$\begin{aligned} f_{93} &= -(m_1 - m_2 + m_3) - (m_4 - m_5 + m_6)L_2 \\ &\quad -(m_7 - m_8 + m_9)L_2^2 \end{aligned} \tag{5.41}$$

$$f_{94} = m_1 - m_2 + m_3 - m_4 + m_5 - m_6 + m_7 - m_8 + m_9 \tag{5.42}$$

and

$$\mathbf{u}(k,l) = [y(k+1,l+1), y(k+1,l), y(k,l+1), y(k,l)]^T \tag{5.43}$$

$$\mathbf{h} = [\mathbf{z}|1]^T \tag{5.44}$$

where $\mathbf{z}$ is 1×8 row vector with all zero elements.

	Recursive	Nonrecursive
Additions	107	$L_1L_2 - 1$
Multiplications	116	L_1L_2

Table 5.1: Comparison of computations

5.4 Performance Analysis of 2-D Recursive and Nonrecursive Finite Memory Filters

In this section, the computational effort, stability, noise performance and frequency response of the recursive and nonrecursive filters are studied.

5.4.1 Computational Effort

To compare the amount of computation, consider the number of additions and multiplications required to compute the output of filters. The results are shown in Table 5.1. It can be seen that for the recursive filter, the amount of computation is independent of the window size. On the contrary, the amount of computation of the nonrecursive filter is proportional to the window size. For $L_1 > 11$ and $L_2 > 11$, the amount of computation necessary for the recursive filter is less than that for the nonrecursive filter.

5.4.2 Stability

The nonrecursive finite memory filter is stable since it has a finite impulse response. To investigate the stability of the recursive filter, consider the state space model given by Eq.(5.20) and Eq.(5.21). Taking the z-transform of these two equations, the transfer function $\mathbf{G}(z_1, z_2)$ of the recursive finite memory filter is obtained

$$G(z_1, z_2) = \mathbf{h}^T(z_1z_2\mathbf{I} - z_2\mathbf{A}_1 - z_1\mathbf{A}_2 - \mathbf{A}_3)^{-1}\mathbf{F}U(z_1, z_2) \qquad (5.45)$$

The characteristic polynomial of the 2-D state space model is given by

$$H(z_1, z_2) = |z_1 z_2 \mathbf{I} - z_2 \mathbf{A}_1 - z_1 \mathbf{A}_2 - \mathbf{A}_3| \tag{5.46}$$

Further investigation shows that $\mathbf{A}_1 \times \mathbf{A}_2 = -\mathbf{A}_3$ which implies that Eq(5.46) can be separated into the following form:

$$H(z_1, z_2) = |z_1 \mathbf{I} - \mathbf{A}_1||z_2 \mathbf{I} - \mathbf{A}_2| \tag{5.47}$$

It is not difficult to show that the characteristic equation of A_1 has nine eigenvalues at $z_1 = \beta_1^{-1}$. Similarly, A_2 has nine eigenvalues at $z_2 = \beta_2^{-1}$. Hence, the recursive finite memory filter is stable provided that $\beta_1 > 1$ and $\beta_2 > 1$. Furthermore, it is clear that the transfer function of the recursive and nonrecursive finite memory filter are the same and thus the recursive filter has pole-zero cancellation at $(z_1, z_2) = (\beta_1^{-1}, \beta_2^{-1})$. These cancellations do not create difficulties provided that $\beta_{1,2} > 1$. For $\beta_{1,2} = 1$ even small numerical errors might generate unstable response.

5.4.3 Noise Performance

Noise is unavoidable during the implementation of the finite memory filter. Measurement noise is caused by errors in measurement, while process noise could be caused by round-off and quantization errors. Here the effect of measurement noise on the filter output is considered.

Eq.(5.9), the nonrecursive filter, can be rewritten as

$$\hat{y}(k, l) = \mathbf{d}^T \mathbf{Y}(k, l) \tag{5.48}$$

where

$$\mathbf{Y}(k, l) = [y(k, l), \ldots, y(k, l - L_2), \ldots, y(k - L_1, l - L_2)]^T \tag{5.49}$$

and

$$\begin{aligned}\mathbf{d} &= [(m_1 + L_1 m_2 + L_1^2 m_3 + L_2 m_4 + L_1 L_2 m_5 + L_1^2 L_2 m_6 \\ &\quad + L_2^2 m_7 + L_1 L_2^2 m_8 + L_1^2 L_2^2 m_9)\beta_1^{L_1}\beta_2^{L_2}, \ldots, \\ &\quad (m_1 + m_2 + m_3)\beta_1, \ldots, (m_1 + m_4 + m_7)\beta_2, m_1]^T \end{aligned} \tag{5.50}$$

In the presence of noise, the measurement vector becomes

$$\mathbf{Y}^*(k,l) = \mathbf{Y}(k,l) + \eta(k,l) \tag{5.51}$$

where $\eta(k,l)$ is the measurement noise vector. The noise is assumed to have zero mean and variance $\sigma^2 I$, where I is the identity matrix. The estimate of $y(k,l)$ in the presence of noise is obtained as

$$\tilde{\mathbf{y}}(k,l) = \mathbf{d}^T \mathbf{Y}^*(k,l) \tag{5.52}$$

The error in the estimation of $y(k,l)$ due to the presence of noise is

$$e(k,l) = \tilde{\mathbf{y}}(k,l) - \hat{\mathbf{y}}(k,l) \tag{5.53}$$

The expected value of $e(k,l)$ is zero because of the assumption of zero mean for the noise and its variance is

$$\begin{aligned} Var[e(k,l)] &= \sigma^2 \mathbf{d}^T \mathbf{d} \\ &= \sigma^2 \sum_{i=0}^{L_1} \sum_{j=0}^{L_2} [(m_1 + i m_2 + i^2 m_3 + j m_4 + ij m_5 \\ &\quad + i^2 j m_6 + j^2 m_7 + i j^2 m_8 + i^2 j^2 m_9)\beta_1^i \beta_2^j]^2 \end{aligned} \tag{5.54}$$

which shows that $Var[e(k,l)]$ is constant for all (k,l). It depends only on the coefficients m_i, $i = 1,2,\ldots,9$, which, in turn, depend on β_1, β_2, L_1 and L_2. The error variance is given in Table 5.2 for some values of $\beta = \beta_1 = \beta_2$ and $L = L_1 = L_2$, assuming that σ^2 is unity. From this, it can be seen that if β_1

L \ β	1.00	1.05	1.10	1.15
3	0.9025	0.9030	0.9042	0.9060
6	0.5805	0.5831	0.5895	0.5985
9	0.3821	0.3864	0.3967	0.4109
12	0.2668	0.2721	0.2850	0.3024

Table 5.2: Error variance Var[e(k,l)] for some values of $\beta_1 = \beta_2$ and $L_1 = L_2$ assuming $\sigma = 1.0$

and/or β_2 is increased, the error variance is also increased, while an increase in L_1 and/or L_2 causes a decrease of the error variance. Theoretically, the error variance for the recursive filter is the same as for the nonrecursive filter, since the operations performed on the input data are equivalent.

5.4.4 Frequency Response

Considering the transfer functions for both recursive and nonrecursive filters, we know that they are polynomials which depend on β_1, β_2, L_1 and L_2. By increasing the window size $L_1 \times L_2$, the filter bandwidth is decreased (see Fig. 5.1 and Fig. 5.2), whereas increasing weighting coefficients reduces the filter selectivity (see Fig. 5.3).

5.5 Conclusion

A recursive implementation of a 2-D second order finite memory filter is presented. The aim of this filter is to fit a 2-D second order polynomial to a $L_1 \times L_2$ window of equally spaced measurements. The parameters of the filter have been obtained using a weighted least-squares performance index. It has been shown that this recursive filter requires less computational effort than the nonrecursive realization. The stability of the recursive filter is guaranteed

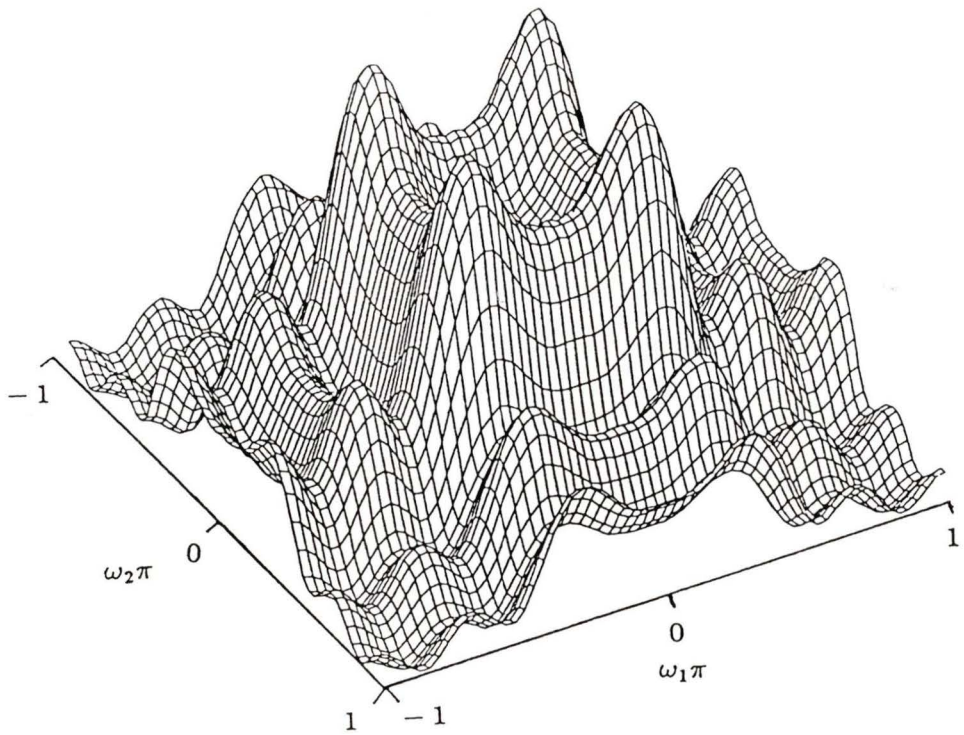


Figure 5.1: Magnitude Response of 2-D Finite Memory Filter for $\beta_1 = \beta_2 = 1.05$, $L_1 = L_2 = 6$

by introducing the weighting factors β_1 and β_2 in the performance index.

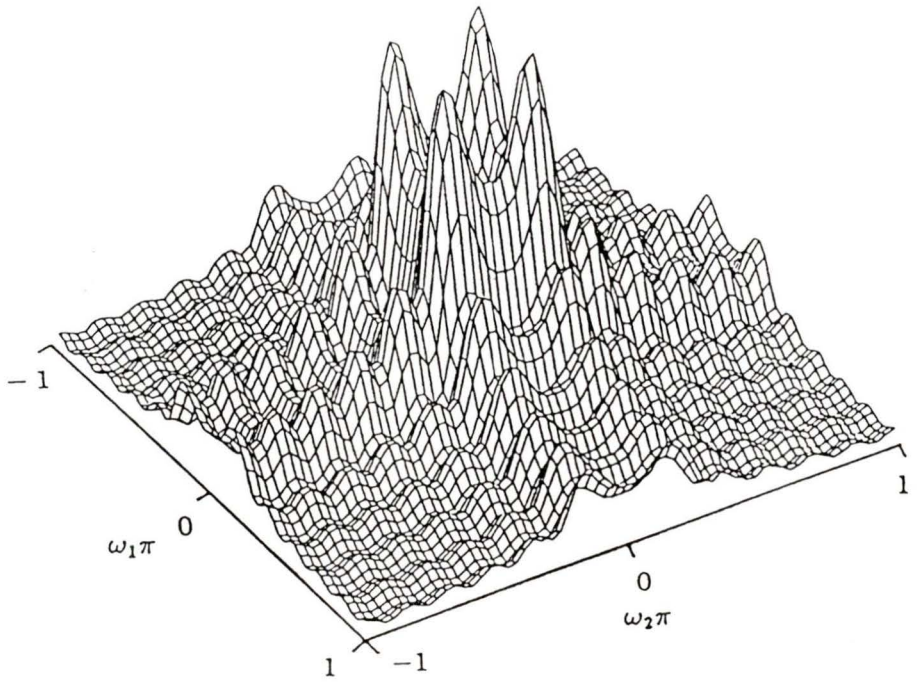


Figure 5.2: Magnitude Response of 2-D Finite Memory Filter for $\beta_1 = \beta_2 = 1.05$, $L_1 = L_2 = 12$

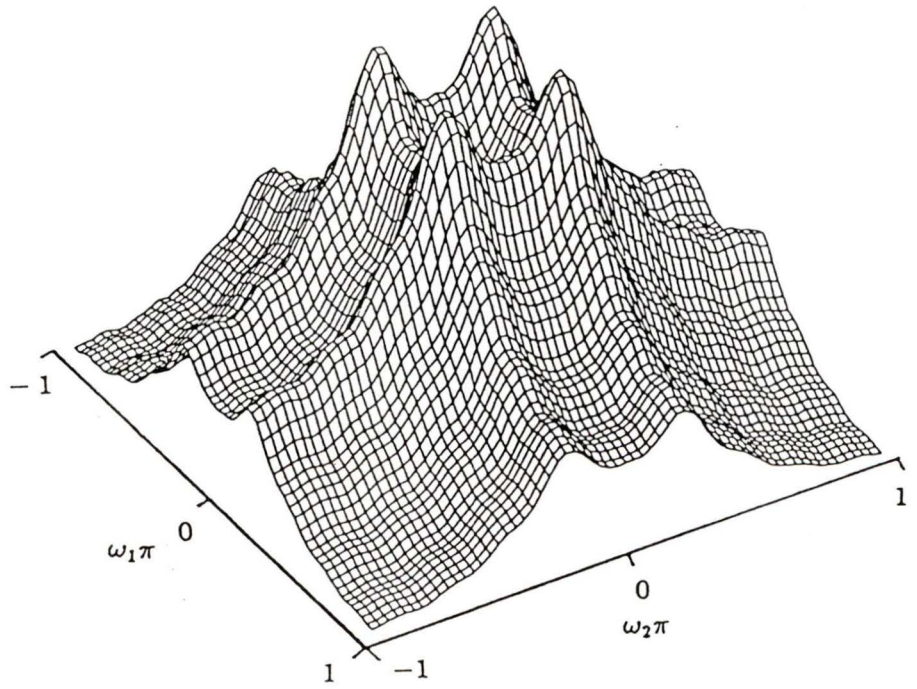


Figure 5.3: Magnitude Response of 2-D Finite Memory Filter for $\beta_1 = \beta_2 = 1.4$, $L_1 = L_2 = 6$

Chapter 6

Summary and suggestions

6.1 Summary

This thesis dealt with the design of the finite memory filters for both the one dimensional and the two dimensional case.

In Chapter 2, the least-squares estimation technique, the recursive least-squares and the Kalman filtering have been reviewed and briefly discussed. Two alternative forms of the Kalman filtering were presented and their particular features have been compared. The relationship between the deterministic least-squares estimation and the Kalman filter has been discussed.

In Chapter 3, a generalized recursive algorithm has been developed for an nth order finite memory filter. The parameters of the filter were determined by applying a weighted least-squares performance index, and therefore the stability of the finite memory filter was ensured. The implementations of the nonrecursive and recursive finite memory filters have been obtained and their stability, computational effort and frequency response have been studied. The performance of the finite memory filter has been illustrated with an example of filtering an audio signal.

In Chapter 4, the Kalman filter and the finite memory filter have been

applied to the problem in which the parameters of a first order polynomial were estimated. The comparison of those two filters were conducted. The results of the comparison showed that the Kalman filter gives better performance in the presence of Gaussian noise while the finite memory filter gives almost the same results but saves significant computation time in the case of nongaussian noise.

In Chapter 5, a second order 2-dimensional finite memory filter has been designed. The corresponding equations for the nonrecursive and recursive finite memory filter were derived using a performance index which included a weighting factor. The stability of the 2-D finite memory filter was guaranteed. The noise performance and the frequency response of the filters have been studied.

6.2 Suggestions for Further Research

There are several interesting topics which may be further extended in future work.

In Chapter 3, the finite memory filters were applied to the audio signal filtering. The investigation of the applications of higher order filters in the area of real time control is an interesting issue, especially for the situation where the noise statistics are nongaussian.

In Chapter 4, the comparison of the Kalman filter and the finite memory filter was conducted. This idea can be extended to 2-dimensional case. The comparison could be made between the 2-D block Kalman filter and 2-D polynomial filter. The applications of those filters can be sought in image restoration.

In Chapter 5, the algorithm of the second order 2-D recursive finite memory filter has been developed. It can be applied to image enhancement problem where the picture has spike and/or film-grain noise. More applica-

tions for which this type of 2-dimensional digital filters is useful can be probed. It is interesting to compare the filter performance by using the first order and the second order 2-D polynomials.

The generalized recursive algorithm for 2-D finite memory filter may be developed. This will lead to lengthy recursive equations which can be used to compare the performance of the recursive and the nonrecursive filters.

The hardware implementation of finite memory filter is of particular interest. Finite memory filters are well suited for systolic array implementations and this would allow a significant speed improvement.

Bibliography

[1] G.T. Wilson, "The factorization of matrical spectral densities," SIAM J. Appl. Math.,vol. 23,pp. 420-426, Dec. 1972.

[2] T. Kailath, "A view of three decades of linear filtering theory," IEEE Trans. on Information Theory, IT-20:No.2, 146-181 (March 1974).

[3] K.F., Gauss, Theory of the motion of the heavenly bodies about the sun in conic sections, Dover, New York, 1963.

[4] H.W., Sorenson, "Estimation theory: A historical perspective," in proc. SW IEEE conf., 1972.

[5] R.A., Fisher, "On an absolute criterion for fitting frequency curves," Messenger of Math, vol.41,1912,p.155.

[6] N., Wiener, The extrapolation, interpolation and smoothing of stationary time series, Join Wiley & Sons, Inc., New York, 1949.

[7] A.N. Kolmogorov, "Interpolation and extrapolation von stationaren zufalligen folgen," Bull. Acad. Sci. USSR, Ser. Math. 5,1941,pp. 3-14.

[8] M. Blum, "Recursion formulas for growing memory digital filters," IRE Trans. Information Theory, Vol. IT-4, pp. 24-30, Mar. 1958.

[9] R.E. Kalman, "A new approach to linear filtering and prediction problems," J. Basic Eng., March 1960, pp.35-46.

[10] R.E. Kalman and R.S. Bucy, "New results in linear filtering and prediction theory," J. Basic Eng., March 1961, pp.95-108.

[11] A.H. Jazwinski, Stochastic processes and filtering theory, New York:Academic, 1970

[12] F.C. Schweppe, Uncertain dynamic systems, Englewood Cliffs, NJ:Prentice-Hall, 1973.

[13] A.H. Jazwinski, "Limited memory optimal filtering," IEEE Trans. on Automat. Contr.,vol. AC-13, Oct. 1968, pp.558-563.

[14] A.M. Bruckstein and T. Kailath, "Recursive limited memory filtering and scattering theory," IEEE Trans. on Information Theory, IT-31:No.3, May 1985, pp.440-443.

[15] F.W. Nesline and P. Zarchan, "Finite memory filters for real time digital control," Automat. Contr. Theory & Appl., 1979, pp.63-66.

[16] M. Schwarty, Information transmission, modulation, and noise, New York:McGraw-Hill, 1959.

[17] P. Agathoklis and M.H. Hamza, "Recursive fixed-memory filter," in IEEE Proc., vol-132,pt. G. No.4, Aug. 1985, pp.119-122.

[18] P. Agathoklis and M.H. Hamza, "A two dimensional finite memory filter," in Proc.of the IASTED symp. on modeling, identification and control, Davos, Switzerland, 1982, pp.286-290.

[19] P. Agathoklis and S. Foda, "A recursive 2-dimensional fixed-memory filter," in Proc.of IEEE Montreal Conf., Montreal, Que., pp. 75-78, Nov. 1987.

[20] P. Agathoklis and H. Xu, "Recursive implementation of 2-dimensional finite memory filters," in Proc. of Canadian Conf. on Elec. & Compu. Engineering, Van., B.C., pp.640-643, Nov. 1988.

[21] P. Agathoklis and H. Xu, "A generalized algorithm for the recursive implementation of polynomial filters," in Proc. of IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing, pp.181-184, June 1989.

[22] A.P. Sage and J.L. Melsa, Estimation theory with applications to communications and control, New York:McGraw-Hill, 1971.

[23] Y.C. Ho, "The method of least squares and optimal filtering theory," Rand Corp. Mem., RM-3329-PR, 1962.

[24] H.W. Sorenson, "Kalman filtering techniques," in advances in control systems (vol.3), C.T. Leondes,(Ed.), New York:Academic Press, 1966.

[25] N. Morrison, Introduction to sequential smoothing and prediction, New York:McGraw-Hill, 1969.

[26] P. Lancaster, The theory of matrix, London:Academic Press Inc., 1985.

[27] M. Mansour, E.I. Jury and F.C.L. Chapparo, "Estimation of the margin of stability for linear continuous and discrete systems," Int. J. Control, 30, pp.49-69, 1979.

[28] D.K. Faddeev and V.N. Faddeeva, Computational method of linear algebra, W.H. Freeman and Company, 1965.

[29] T. Kaczorek, Two-dimensional linear systems, Springer-Verlag Berlin, Heidelberg, 1985.

[30] I. Pitas and A.N. Venetsanopoulos, "A new filter structure for the implementation of certain classes of image processing operations," IEEE Trans. Circuits Syst., vol. CAS-35, pp. 636-647, June 1988.

Appendix A

Matrix B

The matrix B given in Eq.(5.3) has the following form:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} \\ \mathbf{B}_{12} & \mathbf{B}_{13} & \mathbf{B}_{23} \\ \mathbf{B}_{13} & \mathbf{B}_{23} & \mathbf{B}_{33} \end{bmatrix} \tag{A.1}$$

The $B_{ij} \in R^{3\times 3}(i,j = 1,2,3)$ are given by

$$\mathbf{B}_{ij} = \begin{bmatrix} {}^{ij}b_{11} & {}^{ij}b_{12} & {}^{ij}b_{13} \\ {}^{ij}b_{12} & {}^{ij}b_{13} & {}^{ij}b_{23} \\ {}^{ij}b_{13} & {}^{ij}b_{23} & {}^{ij}b_{33} \end{bmatrix} \tag{A.2}$$

with the following expressions for the entries ${}^{ij}b_{k.l}$ $(i,j,k,l = 1,2,3)$

$${}^{11}b_{11} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2}\beta_1^i\beta_2^j = \frac{1-\beta_1^{L_1+1}}{1-\beta_1}\frac{1-\beta_2^{L_2+1}}{1-\beta_2} \tag{A.3}$$

$${}^{11}b_{12} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i\beta_1^i\beta_2^j = \frac{1}{1-\beta_1}[{}^{11}b_{11} - (1+L_1\beta_1^{L_1+1})(\frac{1-\beta_2^{L_2+1}}{1-\beta_2})] \tag{A.4}$$

$${}^{11}b_{13} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^2\beta_1^i\beta_2^j = \frac{1}{1-\beta_1}[2{}^{11}b_{12} - {}^{11}b_{11} + (1-L_1^2\beta_1^{L_1+1})(\frac{1-\beta_2^{L_2+1}}{1-\beta_2})] \tag{A.5}$$

$$
\begin{aligned}
{}^{11}b_{23} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^3\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_1}[3^{11}b_{13} - 3^{11}b_{12} + {}^{11}b_{11} - (1+L_1^3\beta_1^{L_1+1})(\frac{1-\beta_2^{L_2+1}}{1-\beta_2})] \qquad \text{(A.6)}
\end{aligned}
$$

$$
\begin{aligned}
{}^{11}b_{33} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^4\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_1}[4^{11}b_{23} - 6^{11}b_{13} + 4^{11}b_{12} - {}^{11}b_{11} + (1-L_1^4\beta_1^{L_1+1})(\frac{1-\beta_2^{L_2+1}}{1-\beta_2})]
\end{aligned}
\qquad \text{(A.7)}
$$

$$
{}^{12}b_{11} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} j\beta_1^i\beta_2^j = \frac{1}{1-\beta_2}[{}^{11}b_{11} - \frac{1-\beta_1^{L_1+1}}{1-\beta_1}(1+L_2\beta_2^{L_2+1})] \qquad \text{(A.8)}
$$

$$
\begin{aligned}
{}^{12}b_{12} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} ij\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_2}\{{}^{11}b_{12} - \frac{1}{1-\beta_1}[\frac{1-\beta_1^{L_1+1}}{1-\beta_1} - (1+L_1\beta_1^{L_1+1})](1+L_2\beta_2^{L_2+1})]\}
\end{aligned}
\qquad \text{(A.9)}
$$

$$
\begin{aligned}
{}^{12}b_{13} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^2j\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_2}\{{}^{11}b_{13} - \frac{1}{1-\beta_1}[\frac{2-2\beta_1^{L_1+1}}{(1-\beta_1)^2} - \frac{3+\beta_1^{L_1+1}(2L_1-1)}{1-\beta_1} \\
&+ (1-L_1^2\beta_1^{L_1+1})](1+L_2\beta_2^{L_2+1})\} \qquad \text{(A.10)}
\end{aligned}
$$

$$
\begin{aligned}
{}^{12}b_{23} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^3 j \beta_1^i \beta_2^j \\
&= \frac{1}{1-\beta_2}\{{}^{11}b_{23} - \frac{1}{1-\beta_1}[\frac{6-6\beta_1^{L_1+1}}{(1-\beta_1)^3} - \frac{12+\beta_1^{L_1+1}(6L_1-6)}{(1-\beta_1)^2} \\
&+ \frac{7-\beta_1^{L_1+1}(3L_1^2-3L_1+1)}{1-\beta_1} - (1+L_1^3\beta_1^{L_1+1})](1+L_2\beta_2^{L_2+1})\} \qquad \text{(A.11)}
\end{aligned}
$$

$$
\begin{aligned}
{}^{12}b_{33} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^4 j \beta_1^i \beta_2^j \\
&= \frac{1}{1-\beta_2}\{{}^{11}b_{33} - \frac{1}{1-\beta_1}[\frac{24-24\beta_1^{L_1+1}}{(1-\beta_1)^4} - \frac{60+\beta_1^{L_1+1}(24L_1-36)}{(1-\beta_1)^3} \\
&+ \frac{50-\beta_1^{L_1+1}(12L_1^2-24L_1+14)}{(1-\beta_1)^2} \\
&- \frac{15+\beta_1^{L_1+1}(4L_1^3-6L_1^2+4L_1-1)}{1-\beta_1} \\
&+ (1-L_1^4\beta_1^{L_1+1})](1+L_2\beta_2^{L_2+1})\} \qquad \text{(A.12)}
\end{aligned}
$$

$$
{}^{13}b_{11} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} j^2 \beta_1^i \beta_2^j = \frac{1}{1-\beta_2}[2^{12}b_{11} - {}^{11}b_{11} + (\frac{1-\beta_1^{L_1+1}}{1-\beta_1})(1-L_2^2\beta_2^{L_2+1})] \qquad \text{(A.13)}
$$

$$
\begin{aligned}
{}^{13}b_{12} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i j^2 \beta_1^i \beta_2^j \\
&= \frac{1}{1-\beta_2}\{2^{12}b_{12} - {}^{11}b_{12} + \frac{1}{1-\beta_1}[\frac{1-\beta_1^{L_1+1}}{1-\beta_1} - (1+L_1\beta_1^{L_1+1})](1-L_2^2\beta_2^{L_2+1})\}
\end{aligned}
$$

(A.14)

$$
{}^{13}b_{13} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^2 j^2 \beta_1^i \beta_2^j
$$

$$= \frac{1}{1-\beta_2}\{2^{12}b_{13} - {}^{11}b_{13} + \frac{1}{1-\beta_1}[\frac{2-2\beta_1^{L_1+1}}{(1-\beta_1)^2} - \frac{3+\beta_1^{L_1+1}(2L_1-1)}{1-\beta_1}$$

$$+ (1-L_1^2\beta_1^{L_1+1})](1-L_2^2\beta_2^{L_2+1})\} \quad \text{(A.15)}$$

$${}^{13}b_{23} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^3 j^2 \beta_1^i \beta_2^j$$

$$= \frac{1}{1-\beta_2}\{212b_{23} - {}^{11}b_{23} + \frac{1}{1-\beta_1}[\frac{6-6\beta_1^{L_1+1}}{(1-\beta_1)^3} - \frac{12+\beta_1^{L_1+1}(6L_1-6)}{(1-\beta_1)^2}$$

$$+ \frac{7-\beta_1^{L_1+1}(3L_1^2-3L_1+1)}{1-\beta_1} - (1+L_1^3\beta_1^{L_1+1})](1-L_2^2\beta_2^{L_2+1})\}$$

(A.16)

$${}^{13}b_{33} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^4 j^2 \beta_1^i \beta_2^j$$

$$= \frac{1}{1-\beta_2}\{2^{12}b_{33} - {}^{11}b_{33} + \frac{1}{1-\beta_1}[\frac{24-24\beta_1^{L_1+1}}{(1-\beta_1)^4} - \frac{60+\beta_1^{L_1+1}(24L_1-36)}{(1-\beta_1)^3}$$

$$+ \frac{50-\beta_1^{L_1+1}(12L_1^2-24L_1+14)}{(1-\beta_1)^2} - \frac{15+\beta_1^{L_1+1}(4L_1^3-6L_1^2+4L_1-1)}{1-\beta_1}$$

$$+ (1-L_1^4\beta_1^{L_1+1})](1-L_2^2\beta_2^{L_2+1})\} \quad \text{(A.17)}$$

$${}^{23}b_{11} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} j^3 \beta_1^i \beta_2^j$$

$$= \frac{1}{1-\beta_2}[3^{13}b_{11} - 3^{12}b_{11} + {}^{11}b_{11} - (\frac{1-\beta_1^{L_1+1}}{1-\beta_1})(1+L_2^3\beta_2^{L_2+1})] \quad \text{(A.18)}$$

$${}^{23}b_{12} = \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} ij^3 \beta_1^i \beta_2^j$$

$$= \frac{1}{1-\beta_2}\{3^{13}b_{12} - 3^{12}b_{12} + {}^{11}b_{12} - \frac{1}{1-\beta_1}[\frac{1-\beta_1^{L_1+1}}{1-\beta_1} - (1+L_1\beta_1^{L_1+1})]$$

$$(1+L_2^3\beta_2^{L_2+1})\} \quad \text{(A.19)}$$

$$
\begin{aligned}
{}^{23}b_{13} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^2 j^3 \beta_1^i \beta_2^j \\
&= \frac{1}{1-\beta_2}\{3^{13}b_{13} - 3^{12}b_{13} + {}^{11}b_{13} - \frac{1}{1-\beta_1}[\frac{2-2\beta_1^{L_1+1}}{(1-\beta_1)^2} - \frac{3+\beta_1^{L_1+1}(2L_1-1)}{1-\beta_1} \\
&+ (1-L_1^2\beta_1^{L_1+1})](1+L_2^3\beta_2^{L_2+1})\} \qquad \text{(A.20)}
\end{aligned}
$$

$$
\begin{aligned}
{}^{23}b_{23} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^3 j^3 \beta_1^i \beta_2^j \\
&= \frac{1}{1-\beta_2}\{3^{13}b_{23} - 3^{12}b_{23} + {}^{11}b_{23} - \frac{1}{1-\beta_1}[\frac{6-6\beta_1^{L_1+1}}{(1-\beta_1)^3} - \frac{12+\beta_1^{L_1+1}(6L_1-6)}{(1-\beta_1)^2} \\
&+ \frac{7-\beta_1^{L_1+1}(3L_1^2-3L_1+1)}{1-\beta_1} - (1+L_1^3\beta_1^{L_1+1})](1+L_2^3\beta_2^{L_2+1})\} \qquad \text{(A.21)}
\end{aligned}
$$

$$
\begin{aligned}
{}^{23}b_{33} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^4 j^3 \beta_1^i \beta_2^j \\
&= \frac{1}{1-\beta_2}\{3^{13}b_{33} - 3^{12}b_{33} + {}^{11}b_{33} - \frac{1}{1-\beta_1}[\frac{24-24\beta_1^{L_1+1}}{(1-\beta_1)^4} \\
&- \frac{60+\beta_1^{L_1+1}(24L_1-36)}{(1-\beta_1)^3}\frac{50-\beta_1^{L_1+1}(12L_1^2-24L_1+14)}{(1-\beta_1)^2} \\
&- \frac{15+\beta_1^{L_1+1}(4L_1^3-6L_1^2+4L_1-1)}{1-\beta_1} + (1-L_1^4\beta_1^{L_1+1})](1+L_2^3\beta_2^{L_2+1})\}
\end{aligned}
$$

$$
\text{(A.22)}
$$

$$
\begin{aligned}
{}^{33}b_{11} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} j^4 \beta_1^i \beta_2^j \\
&= \frac{1}{1-\beta_2}[4^{23}b_{11} - 6^{13}b_{11} + 4^{12}b_{11} - {}^{11}b_{11} \\
&+ (\frac{1-\beta_1^{L_1+1}}{1-\beta_1})(1-L_2^4\beta_2^{L_2+1})] \qquad \text{(A.23)}
\end{aligned}
$$

$$
\begin{aligned}
{}^{33}b_{12} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} ij^4\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_2}\{4^{23}b_{12} - 6^{13}b_{12} + 4^{12}b_{12} - {}^{11}b_{12} \\
&+ \frac{1}{1-\beta_1}[\frac{1-\beta_1^{L_1+1}}{1-\beta_1} - (1+L_1\beta_1^{L_1+1})](1-L_2^4\beta_2^{L_2+1})\}
\end{aligned}
\tag{A.24}
$$

$$
\begin{aligned}
{}^{33}b_{13} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^2j^4\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_2}\{4^{23}b_{13} - 6^{13}b_{13} + 4^{12}b_{13} - {}^{11}b_{13} + \frac{1}{1-\beta_1}[\frac{2-2\beta_1^{L_1+1}}{(1-\beta_1)^2} \\
&- \frac{3+\beta_1^{L_1+1}(2L_1-1)}{1-\beta_1} + (1-L_1^2\beta_1^{L_1+1})](1-L_2^4\beta_2^{L_2+1})\}
\end{aligned}
\tag{A.25}
$$

$$
\begin{aligned}
{}^{33}b_{23} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^3j^4\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_2}\{4^{23}b_{23} - 6^{13}b_{23} + 4^{12}b_{23} - {}^{11}b_{23} \\
&+ \frac{1}{1-\beta_1}[\frac{6-6\beta_1^{L_1+1}}{(1-\beta_1)^3} - \frac{12+\beta_1^{L_1+1}(6L_1-6)}{(1-\beta_1)^2} \\
&+ \frac{7-\beta_1^{L_1+1}(3L_1^2-3L_1+1)}{1-\beta_1} - (1+L_1^3\beta_1^{L_1+1})](1-L_2^4\beta_2^{L_2+1})\}
\end{aligned}
\tag{A.26}
$$

$$
\begin{aligned}
{}^{33}b_{33} &= \sum_{i=0}^{L_1}\sum_{j=0}^{L_2} i^4j^4\beta_1^i\beta_2^j \\
&= \frac{1}{1-\beta_2}\{4^{23}b_{33} - 6^{13}b_{33} + 4^{12}b_{33} - {}^{11}b_{33} + \frac{1}{1-\beta_1}[\frac{24-24\beta_1^{L_1+1}}{(1-\beta_1)^4}
\end{aligned}
$$

$$- \frac{60+\beta_1^{L_1+1}(24L_1-36)}{(1-\beta_1)^3} + \frac{50-\beta_1^{L_1+1}(12L_1^2-24L_1+14)}{(1-\beta_1)^2}$$

$$- \frac{15+\beta_1^{L-1+1}(4L_1^3-6L_1^2+4L_1-1)}{1-\beta_1} + (1-L_1^4\beta_1^{L_1+1})](1-L_2^4\beta_2^{L_2+1})\}$$

(A.27)

Appendix B

Derivation of Recursive Form

In this appendix, the derivation of the recursive algorithm for the vector **c** is presented. The derivations for $c_{0,1}$, $c_{0,2}$ and $c_{1,2}$ are similar to $c_{1,0}$, $c_{2,0}$ and $c_{2,1}$.

$$c_{0,0}(k+1,l+1)$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l+1-L_2}^{l+1} \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)$$

$$+ \sum_{i=k+1-L_1}^{k+1} \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i,l+1) - \beta_2^{-1} y(i,l-L_2)]$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)$$

$$+ \sum_{i=k-L_1}^{k} \beta_1^{i-k+L_1-1} [\sum_{j=l+1-L_2}^{l+1} \beta_2^{j-l+L_2-1} y(i,j) - \sum_{j=l-L_2}^{l} \beta_2^{j-l+L_2-1} y(i,j)]$$

$$+ [\beta_1^{L_1} \beta_2^{L_2} y(k+1,l+1) - \beta_1^{-1} \beta_2^{L_2} y(k-L_1,l+1)$$

$$- \beta_1^{L_1} \beta_2^{-1} y(k+1,l-L_2) + \beta_1^{-1} \beta_2^{-1} y(k-L_1,l-L_2)]$$

$$
\begin{aligned}
&= \beta_1^{-1} \sum_{i=k-L_1}^{k} \sum_{j=l+1-L_2}^{l+1} \beta_1^{i-k+L_1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \beta_2^{-1} \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2} y(i,j) \\
&- \beta_1^{-1}\beta_2^{-1} \sum_{i=k-L_1}^{k} \sum_{j=l-L_2}^{l} \beta_1^{i-k+L_1} \beta_2^{j-l+L_2} y(i,j) \\
&+ \beta_1^{L_1}[\beta_2^{L_2} y(k+1,l+1) - \beta_2^{-1} y(k+1,l-L_2)] \\
&- \beta_1^{-1}[\beta_2^{L_2} y(k-L_1,l+1) - \beta_2^{-1} y(k-L_1,l-L_2)] \\
&= \beta_1^{-1} c_{0,0}(k,l+1) + \beta_2^{-1} c_{0,0}(k+1,l) - \beta_1^{-1}\beta_2^{-1} c_(k,l) \\
&+ \beta_1^{-1}\beta_2^{-1} y(k-L_1,l-L_2) - \beta_1^{-1}\beta_2^{L_2} y(k-L_1,l+1) \\
&- \beta_1^{L_1}\beta_2^{-1} y(k+1,l-L_2) + \beta_1^{L_1}\beta_2^{L_2} y(k+1,l+1) \qquad \text{(B.1)}
\end{aligned}
$$

$$
\begin{aligned}
&c_{1,0}(k+1,l+1) \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l+1-L_2}^{l+1} (i-k+L_1-1) \beta_1^{i-k+L_1} \beta_2^{j-l+L_2} y(i,j) \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1) \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} (i-k+L_1-1) \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i,l+1) - \beta_2^{-1} y(i,l-L_2)] \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1) \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} (i-k+L_1) \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i,l+1) - \beta_2^{-1} y(i,l-L_2)] \\
&- \sum_{i=k+1-L_1}^{k+1} \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i,l+1) - \beta_2^{-1} y(i,l-L_2)]
\end{aligned}
$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j)$$

$$+ \sum_{i=k-L_1}^{k} (i-k+L_1)\beta_1^{i-k+L_1-1}$$

$$[\sum_{j=l+1-L_2}^{l+1} \beta_2^{j-l+L_2-1}y(i,j) - \sum_{j=l-L_2}^{l} \beta_2^{j-l+L_2-1}y(i,j)]$$

$$+ (L_1+1)\beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1) - (L_1+1)\beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2)$$

$$- \{\sum_{i=k-L_1}^{k} \beta_1^{i-k+L_1-1}[\sum_{j=l+1-L_2}^{l+1} \beta_2^{j-l+L_2-1}y(i,j) - \sum_{j=l-L_2}^{l} \beta_2^{j-l+L_2-1}y(i,j)]$$

$$+ \beta_1^{-1}\beta_2^{-1}y(k-L_1,l-L_2) - \beta_1^{-1}\beta_2^{L_2}y(k-L_1,l+1)$$

$$- \beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2) + \beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1)\}$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j)$$

$$+ \sum_{i=k-L_1}^{k} (i-k+L_1)\beta_1^{i-k+L_1-1}$$

$$[\sum_{j=l+1-L_2}^{l+1} \beta_2^{j-l+L_2-1}y(i,j) - \sum_{j=l-L_2}^{l} \beta_2^{j-l+L_2-1}y(i,j)]$$

$$- \sum_{i=k-L_1}^{k} \beta_1^{i-k+L_1-1}[\sum_{j=l+1-L_2}^{l+1} \beta_2^{j-l+L_2-1}y(i,j) - \sum_{j=l-L_2}^{l} \beta_2^{j-l+L_2-1}y(i,j)]$$

$$+ L_1\beta_1^{L_1}[\beta_2^{L_2}y(k+1,l+1) - \beta_2^{-1}y(k+1,l-L_2)]$$

$$+ \beta_1^{-1}[\beta_2^{L_2}y(k-L_1,l+1) - \beta_2^{-1}y(k-L_1,l-L_2)]$$

$$= \beta_1^{-1}c_{1,0}(k,l+1) + \beta_2^{-1}c_{1,0}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l)$$

$$- \beta_1^{-1}c_{0,0}(k,l+1) + \beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l)$$

$$+ L_1\beta_1^{L_1}[\beta_2^{L_2}y(k+1,l+1) - \beta_2^{-1}y(k+1,l-L_2)]$$

$$+ \quad \beta_1^{-1}[\beta_2^{L_2} y(k-L_1, l+1) - \beta_2^{-1} y(k-L_1, l-L_2)] \tag{B.2}$$

$c_{2,0}(k+1, l+1)$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l+1-L_2}^{l+1} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)$$

$$+ \sum_{i=k+1-L_1}^{k+1} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i, l+1) - \beta_2^{-1} y(i, l-L_2)]$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)$$

$$+ \sum_{i=k+1-L_1}^{k+1} (i-k+L_1)^2 \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i, l+1) - \beta_2^{-1} y(i, l-L_2)]$$

$$- 2 \sum_{i=k+1-L_1}^{k+1} (i-k+L_1) \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i, l+1) - \beta_2^{-1} y(i, l-L_2)]$$

$$+ \sum_{i=k+1-L_1}^{k+1} \beta_1^{i-k+L_1-1} [\beta_2^{L_2} y(i, l+1) - \beta_2^{-1} y(i, l-L_2)]$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)$$

$$+ \sum_{i=k-L_1}^{k} (i-k+L_1)^2 \beta_1^{i-k+L_1-1}$$

$$\left[\sum_{j=l+1-L_2}^{l+1} \beta_2^{j-l+L_2-1} y(i,j) - \sum_{j=l-L_2}^{l} \beta_2^{j-l+L_2-1} y(i,j) \right]$$

$$+ \quad (L_1+1)^2 \beta_1^{L_1} \beta_2^{L_2} y(k+1, l+1) - (L_1+1)^2 \beta_1^{L_1} \beta_2^{-1} y(k+1, l-L_2)$$

$$
\begin{aligned}
&- 2\{\sum_{i=k-L_1}^{k}(i-k+L_1)\beta_1^{i-k+L_1-1} \\
&\quad [\sum_{j=l+1-L_2}^{l+1}\beta_2^{j-l+L_2-1}y(i,j)-\sum_{j=l-L_2}^{l}\beta_2^{j-l+L_2-1}y(i,j)] \\
&- 2(L_1+1)\beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1)+2(L_1+1)\beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2) \\
&+ \sum_{i=k-L_1}^{k}\beta_1^{i-k+L_1-1}[\sum_{j=l+1-L_2}^{l+1}\beta_2^{j-l+L_2-1}y(i,j)-\sum_{j=l-L_2}^{l}\beta_2^{j-l+L_2-1}y(i,j)] \\
&+ \beta_1^{-1}\beta_2^{-1}y(k-L_1,l-L_2)-\beta_1^{-1}\beta_2^{L_2}y(k-L_1,l+1) \\
&- \beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2)+\beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1)\} \\
&= \beta_1^{-1}c_{2,0}(k,l+1)+\beta_2^{-1}c_{2,0}(k+1,l)-\beta_1^{-1}\beta_2^{-1}c_{2,0}(k,l) \\
&- 2\beta_1^{-1}c_{1,0}(k,l+1)+2\beta_1^{-1}\beta_2^{-1}c_{1,0}(k,l) \\
&+ \beta_1^{-1}c_{0,0}(k,l+1)-\beta_1^{-1}\beta_2^{-1}c_{0,0}(k,l) \\
&+ L_1^2\beta_1^{L_1}[\beta_2^{L_2}y(k+1,l+1)-\beta_2^{-1}y(k+1,l-L_2)] \\
&- \beta_1^{-1}[\beta_2^{L_2}y(k-L_1,l+1)-\beta_2^{-1}y(k-L_1,l-L_2)] \qquad \text{(B.3)}
\end{aligned}
$$

$c_{1,1}(k+1,l+1)$

$$
\begin{aligned}
&= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l+1-L_2}^{l+1}(i-k+L_1-1)(j-k+L_2-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2}y(i,j) \\
&= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l-L_2}^{l}(i-k+L_1-1)(j-l+L_2-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1}(i-k+L_1-1)\beta_1^{i-k+L_1-1}[L_2\beta_2^{L_2}y(i,l+1)+\beta_2^{-1}y(i,l-L_2)] \\
&= \sum_{i=k+1-L_1}^{k+1}\sum_{j=l-L_2}^{l}(i-k+L_1-1)(j-l+L_2-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j)
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} (i-k+L_1)\beta_1^{i-k+L_1-1}[L_2\beta_2^{L_2}y(i,l+1)+\beta_2^{-1}y(i,l-L_2)] \\
&- \sum_{i=k+1-L_1}^{k+1} (i-k+L_1)\beta_1^{i-k+L_1-1}[L_2\beta_2^{L_2}y(i,l+1)+\beta_2^{-1}y(i,l-L_2)] \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)(j-l+L_2-1) \\
&\quad \beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&- \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1}y(i,j) \\
&+ \sum_{i=k-L_1}^{k} (i-k+L_1)\beta_1^{i-k+L_1-1} \\
&\quad [\sum_{j=l+1-L_2}^{l+1} (j-l+L_2-1)\beta_2^{j-l+L_2-1}y(i,j) \\
&\quad - \sum_{j=l-L_2}^{l} (j-l+L_2-1)\beta_2^{j-l+L_2-1}y(i,j)] \\
&+ (L_1+1)L_2\beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1)+(L_1+1)\beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2) \\
&- \{\sum_{i=k-L_1}^{k} \beta_1^{i-k+L_1-1}[\sum_{j=l+1-L_2}^{l+1} (j-l+L_2-1)\beta_2^{j-l+L_2-1}y(i,j) \\
&\quad - \sum_{j=l-L_2}^{l} (j-l+L_2-1)\beta_2^{j-l+L_2-1}y(i,j)] \\
&- \beta_1^{-1}\beta_2^{-1}y(k-L_1,l-L_2)-L_2\beta_1^{-1}\beta_2^{L_2}y(k-L_1,l+1) \\
&+ \beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2)+L_2\beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1)\} \\
&= \beta_1^{-1}c_{1,1}(k,l+1)+\beta_2^{-1}c_{1,1}(k+1,l)-\beta_1^{-1}\beta_2^{-1}c_{1,1}(k,l)
\end{aligned}
$$

$$
\begin{aligned}
&- \beta_1^{-1} c_{0,1}(k, l+1) + \beta_1^{-1}\beta_2^{-1} c_{0,1}(k, l) \\
&- \beta_2^{-1} c_{1,0}(k+1, l) + \beta_1^{-1}\beta_2^{-1} c_{1,0}(k, l) - \beta_1^{-1}\beta_2^{-1} c_{0,0}(k, l) \\
&+ L_1 \beta_1^{L_1} [L_2 \beta_2^{L_2} y(k+1, l+1) + \beta_2^{-1} y(k+1, l-L_2)] \\
&+ \beta_1^{-1} [L_2 \beta_2^{L_2} y(k-L_1, l+1) + \beta_2^{-1} y(k-L_1, l-L_2)] \qquad \text{(B.4)}
\end{aligned}
$$

$c_{2,1}(k+1, l+1)$

$$
\begin{aligned}
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l+1-L_2}^{l+1} (i-k+L_1-1)^2 (j-l+L_2-1) \beta_1^{i-k+L_1} \beta_2^{j-l+L_2} y(i,j) \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 (j-l+L_2-1) \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} [L_2 \beta_2^{L_2} y(i, l+1) + \beta_2^{-1} y(i, l-L_2)] \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 (j-l+L_2) \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&- \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} (i-k+L_1)^2 \beta_1^{i-k+L_1-1} [L_2 \beta_2^{L_2} y(i, l+1) + \beta_2^{-1} y(i, l-L_2)] \\
&- 2 \sum_{i=k+1-L_1}^{k+1} (i-k+L_1) \beta_1^{i-k+L_1-1} [L_2 \beta_2^{L_2} y(i, l+1) + \beta_2^{-1} y(i, l-L_2)] \\
&+ \sum_{i=k+1-L_1}^{k+1} \beta_1^{i-k+L_1-1} [L_2 \beta_2^{L_2} y(i, l+1) + \beta_2^{-1} y(i, l-L_2)] \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 (j-l+L_2) \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j)
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k-L_1}^{k} (i-k+L_1)^2 \beta_1^{i-k+L_1-1} \\
&\quad [\sum_{j=l+1-L_2}^{l+1} \beta_2^{j-l+L_2-1} y(i,j) - \sum_{j=l-L_2}^{l} \beta_2^{j-l+L_2-1} y(i,j)] \\
&+ (L_1+1)^2 L_2 \beta_1^{L_1} \beta_2^{L_2} y(k+1,l+1) + (L_1+1)^2 \beta_1^{L_1} \beta_2^{-1} y(k+1,l-L_2) \\
&- 2\{\{ \sum_{i=k-L_1}^{k} (i-k+L_1) \beta_1^{i-k+L_1-1} \\
&\quad [\sum_{j=l+1-L_2}^{l+1} (i-l+L_2-1) \beta_2^{j-l+L_2-1} y(i,j) \\
&\quad - \sum_{j=l-L_2}^{l} (j-l+L_2-1) \beta_2^{j-l+L_2-1} y(i,j)] \\
&+ (L_1+1)^2 L_2 \beta_1^{L_1} \beta_2^{L_2} y(k+1,l+1) + (L_1+1)^2 \beta_1^{L_1} \beta_2^{-1} y(k+1,l-L_2)\} \\
&+ \sum_{i=k-L_1}^{k} \beta_1^{i-k+L_1-1} \\
&\quad [\sum_{j=l+1-L_2}^{l+1} (j-l+L_2-1) \beta_2^{j-l+L_2-1} y(i,j) \\
&\quad - \sum_{j=l-L_2}^{l} (j-l+L_2-1) \beta_2^{j-l+L_2-1} y(i,j)] \\
&- \beta_1^{-1} \beta_2^{-1} y(k-L_1,l-L_2) - L_2 \beta_1^{-1} \beta_2^{L_2} y(k-L_1,l+1) \\
&+ \beta_1^{L_1} \beta_2^{-1} y(k+1,l-L_2) + L_2 \beta_1^{L_1} \beta_2^{L_2} y(k+1,l+1)\} \\
&= \beta_1^{-1} c_{2,1}(k,l+1) + \beta_2^{-1} c_{2,1}(k+1,l) - \beta_1^{-1} \beta_2^{-1} c_{2,1}(k,l) \\
&- 2\beta_1^{-1} c_{1,1}(k,l+1) + 2\beta_1^{-1} \beta_2^{-1} c_{1,1}(k,l)
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad \beta_1^{-1} c_{0,1}(k, l+1) - \beta_1^{-1}\beta_2^{-1} c_{0,1}(k,l) \\
&- \quad \beta_2^{-1} c_{2,0}(k+1, l) + \beta_1^{-1}\beta_2^{-1} c_{2,0}(k,l) \\
&- \quad 2\beta_1^{-1}\beta_2^{-1} c_{1,0}(k,l) + \beta_1^{-1}\beta_2^{-1} c_{0,0}(k,l) \\
&+ \quad L_1^2 \beta_1^{L_1} [L_2 \beta_2^{L_2} y(k+1, l+1) - \beta_2^{-1} y(k+1, i - L_2)] \\
&- \quad \beta_1^{-1} [L_2 \beta_2^{L_2} y(k - L_1, l+1) - \beta_2^{-1} y(k - L_1, l - L_2)]
\end{aligned}
\tag{B.5}
$$

$c_{2,2}(k+1, l+1)$

$$
\begin{aligned}
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l+1-L_2}^{l+1} (i - k + L_1 - 1)^2 (j - l + L_2 - 1)^2 \\
&\quad \beta_1^{i-k+L_1} \beta_2^{j-l+L_2} y(i,j) \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i - k + L_1 - 1)^2 (j - l + L_2 - 1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} (i - k + L_1 - 1)^2 \beta_1^{i-k+L_1-1} [L_2^2 \beta_2^{L_2} y(i, l+1) - \beta_2^{-1} y(i, l - L_2)] \\
&= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i - k + L_1 - 1)^2 \\
&\quad [(j - l + L_2)^2 - 2(j - l + L_2) + 1] \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i - k + L_1 - 1)^2 \beta_1^{i-k+L_1-1} \beta_2^{j-l+L_2-1} y(i,j) \\
&+ \sum_{i=k+1-L_1}^{k+1} (i - k + L_1)^2 \beta_1^{i-k+L_1-1} [L_2^2 \beta_2^{L_2} y(i, l+1) - \beta_2^{-1} y(i, l - L_2)] \\
&- 2 \sum_{i=k+1-L_1}^{k+1} (i - k + L_1) \beta_1^{i-k+L_1-1} [L_2^2 \beta_2^{L_2} y(i, l+1) - \beta_2^{-1} y(i, l - L_2)]
\end{aligned}
$$

$$+ \sum_{i=k+1-L_1}^{k+1} \beta_1^{i-k+L_1-1}[\beta_2^{L_2} y(i,l+1) - \beta_2^{-1} y(i,l-L_2)]$$

$$= \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2$$

$$[(j-l+L_2)^2 - 2(j-l+L_2)+1]\beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1} y(i,j)$$

$$+ \sum_{i=k+1-L_1}^{k+1} \sum_{j=l-L_2}^{l} (i-k+L_1-1)^2 \beta_1^{i-k+L_1-1}\beta_2^{j-l+L_2-1} y(i,j)$$

$$+ \sum_{i=k-L_1}^{k} (i-k+L_1)^2 \beta_1^{i-k+L_1-1}$$

$$[\sum_{j=l+1-L_2}^{l+1} (j-l+L_2-1)^2 \beta_2^{j-l+L_2-1} y(i,j)$$

$$- \sum_{j=l-L_2}^{l} (j-l+L_2-1)\beta_2^{j-l+L_2-1} y(i,j)]$$

$$+ (L_1+1)^2 L_2^2 \beta_1^{L_1}\beta_2^{L_2} y(k+1,l+1) - (L_1+1)^2 \beta_1^{L_1}\beta_2^{-1} y(k+1,l-L_2)$$

$$- 2\{\sum_{i=k-L_1}^{k} (i-k+L_1)\beta_1^{i-k+L_1-1}$$

$$[\sum_{j=l+1-L_2}^{l+1} (j-l+L_2-1)^2 \beta_2^{j-l+L_2-1} y(i,j)$$

$$- \sum_{j=l-L_2}^{l} (j-l+L_2-1)^2 \beta_2^{j-l+L_2-1} y(i,j)]$$

$$+ (L_1+1) L_2^2 \beta_1^{L_1}\beta_2^{L_2} y(k+1,l+1) - (L_1+1)\beta_1^{L_1}\beta_2^{-1} y(k+1,l-L_2)$$

$$+ \sum_{i=k-L_1}^{k} \beta_1^{i-k+L_1-1}[\sum_{j=l+1-L_2}^{l+1} (j-l+L_2-1)^2 \beta_2^{j-l+L_2-1} y(i,j)$$

$$- \sum_{j=l-L_2}^{l} (j-l+L_2-1)^2 \beta_2^{j-l+L_2-1} y(i,j)]$$

$$
\begin{aligned}
&+ \beta_1^{-1}\beta_2^{-1}y(k-L_1,l-L_2) - L_2^2\beta_1^{-1}\beta_2^{L_2}y(k-L_1,l+1) \\
&- \beta_1^{L_1}\beta_2^{-1}y(k+1,l-L_2) + L_2^2\beta_1^{L_1}\beta_2^{L_2}y(k+1,l+1)\} \\
&= \beta_1^{-1}c_{2,2}(k,l+1) + \beta_2^{-1}c_{2,2}(k+1,l) - \beta_1^{-1}\beta_2^{-1}c_{2,2}(k,l) \\
&- 2\beta_1^{-1}c_{2,1}(k,l+1) - 2\beta_2^{-1}c_{2,1}(k+1,l) \\
&- 2\beta_1^{-1}\beta_2{-}1c_{2,1}(k,l) - 2\beta_1^{-1}\beta_2{-}1c_{1,2}(k,l) \\
&+ \beta_1^{-1}c_{0,2}(k,l+1) - \beta_1^{-1}\beta_2{-}1c_{0,2}(k,l) \\
&+ \beta_2^{-1}c_{2,0}(k+1,l) - \beta_1^{-1}\beta_2{-}1c_{2,0}(k,l) \\
&- 4\beta_1^{-1}\beta_2{-}1c_{1,1}(k,l) + 2\beta_1^{-1}\beta_2{-}1c_{1,0}(k,l) \\
&+ 2\beta_1^{-1}\beta_2{-}1c_{0,1}(k,l) - \beta_1^{-1}\beta_2{-}1c_{0,0}(k,l) \\
&+ L_1^2\beta_1^{L_1}[L_2^2\beta_2^{L_2}y(k+1,l+1) - \beta_2^{-1}y(k+1,l-L_2)] \\
&- \beta_1^{-1}[L_2^2\beta_2^{L_2}y(k-L_1,l+1) - \beta_2^{-1}y(k-L_1,l-L_2)] \qquad \text{(B.6)}
\end{aligned}
$$

VITA

Surname: *Xu* Given Name: *Hao*

Place of Birth: *Shanghai, China* Date of Birth: *October 20, 1957*

Educational Institutions Attended, with Dates of Entering and Leaving:
Shanghai Second Polytechnic University, Shanghai 1979 to 1983

Degrees, Diplomas, Etc., Awarded, with Dates and Names of Institutions:
B.Eng. 1983 Shanghai Second Polytechnic University, Shanghai

Honors and Awards:
University of Victoria Fellowship, 1986/1987

Publications:

1. P. Agathoklis and H. Xu, "Recursive Implementation of 2-dimensional finite memory filters," in Proc. of Canadian Conf. on Elec. & Compu. Engineering, Van., B.C., pp. 640-643, Nov. 1988.

2. P. Agathoklis and H. Xu, "A generalized algorithm for the recursive implementation of polynomial filters," in Proc. of IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing, pp.123-126, June 1989.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, on similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis

Design of Finite Memory Filters

Author

Signature

Hao Xu

Name

August 10, 1989

Date

ISBN 0-315-53725-6