Blind Deconvolution Based on Constrained Marginalized Particle Filters

BLIND DECONVOLUTION BASED ON CONSTRAINED MARGINALIZED PARTICLE FILTERS

BY

KRZYSZTOF S. MARYAN, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2008) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Blind Deconvolution Based on Constrained Marginalized Particle Filters
AUTHOR:	Krzysztof S. Maryan B.Sc., (Computer Engineering and Management) McMaster University, Hamilton, Ontario, Canada
SUPERVISOR:	Dr. J. Reilly
NUMBER OF PAGES:	xii, 74

Dedicated to my parents

and

to the hard working people in ITB-A202

Abstract

This thesis presents a new approach to blind deconvolution algorithms. The proposed method is a combination of a classical blind deconvolution subspace method and a marginalized particle filter. It is shown that the new method provides better performance than just a marginalized particle filter, and better robustness than the classical subspace method. The properties of the new method make it a candidate for further exploration of its potential application in acoustic blind dereverberation.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. James Reilly, for his guidance and assistance over the course of this degree; from accepting me to his research group to helping me revise this thesis. I would also like to recognize Dr. Tim Davidson for challenging me constantly during my years at McMaster. To my friends, I thank you for your support over the last two years. I am deeply indebted to the multitude of students in Electrical and Computer Engineering at McMaster who have helped me in my time at here, especially the residents of ITB-A202. Finally I would like to thank Lisa for her love and patience as I have worked towards completing this degree, and thank my parents to whom I owe a great deal of gratitude for their incredible support right from day one.

K. Maryan

September 17, 2008

Notation

$\mathbf{x}$	Lowercase letters in bold depict vectors
$\mathbf{V}$	Uppercase letters in bold depict matrices
$\mathbf{0}_{M \times N}$	The matrix of all zeros of dimension $M \times N$
$diag(a_1, a_2, a_3)$	A matrix of containing a_1, a_2, a_3 along the diagonal
x_t	x at time t
$x_{p,t}$	x for particle p, at time t
$q(z)$	q is a function of z
$y^{(n)}$	y corresponding to observation channel n
$\hat{x}$	The estimate of x
$p(x)$	The probability of x
$p(x\|y)$	The probability of x given y
$\mathcal{N}(\mu, \sigma)$	The Gaussian (normal) distribution, with mean μ and variance σ
$E[x]$	The expected value of x
$*$	The convolution operator

Contents

List of Figures

Chapter 1

Introduction

This thesis explores the use of a particle filter based estimator for solving the acoustic blind deconvolution problem. Acoustic blind deconvolution, discussed in more detail in the next section, is a complex estimation problem that suffers from a number of factors that make good solutions difficult. It is the assertion of this thesis that a particle filter based formulation of the problem can resolve some of the difficulties.

Particle filters can provide a very flexible framework for model based estimation problems. This can be exploited to solve problems where incorporating prior knowledge of a model for the system can be useful. Thus a model which can incorporate as much prior information as possible can be very beneficial to generating a good estimate.

1.1 Problem Description

Blind deconvolution, is the estimation of an unknown signal source based on observations obtained through unknown convolutive channels. The problem is seen in a number of areas, including acoustics, which will be the focus of this work.

- In acoustics, blind deconvolution, or blind dereverberation, deals with the recovery of a sound source, such as a person speaking, based on observations by a microphone or multiple microphones situated throughout the room. A microphone will record the sound, plus the cumulative effect of the sound reflecting off the walls and other surface in the room, the reverberation. Recovering the source is desirable for hands free telephones

and the professional audio field as part of the process of enhancing the quality of the sound.

Other areas include seismology, biomedical signal processing and image processing.

- In seismology, a technique for estimating the composition of a geological formation involves applying a disturbance, such as an explosion, at several points, and listening with seismographs at various points for the disturbance signal to return. Estimating the channel between the disturbance and the seismograph can provide information about the geological structure of the area [33].

- In medical applications, an electro-encephalography (EEG) recording is made by a number of sensors around a patients head. Events in the brain generate electrical signals that are received after convolutatively mixing with the environment [11]. The aim is to recover the original event transmissions to make inferences about activity in the brain.

- Blind deconvolution extends beyond time domain signals to also encompass areas such as image restoration. An image may be distorted due to an unknown point spread function of an imaging system and enhancement to restore the original is desired [31]. This problem comes up in many image processing applications, notably astronomical image processing and microscopy.

This thesis will consider only deconvolution of time domain signals from a single source with multiple receivers observing through convolutive channels: a single input, multiple out process (SIMO). Related problems are single channel blind deconvolution, where a single sensor records the source after it has passed through a channel; a single input, single output (SISO) problem. And also multiple inputs with multiple outputs (MIMO), where the sources must be separated from each other and the channels that distort them; blind source separation combined with blind deconvolution. These types of systems are pictured in figure 1.1.

From a conceptual point of view, acoustic blind deconvolution in a SIMO system is simple to consider. The single source is received at each microphone with a time delay and a change in amplitude corresponding to the distance

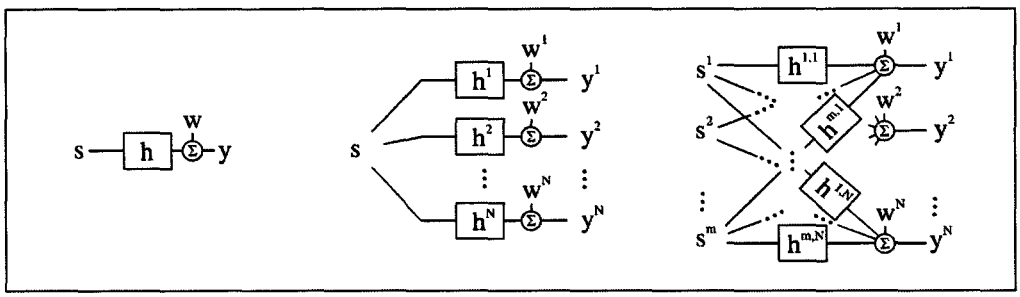

Figure 1.1: Single Input, Single Output (SISO), Single Input Multiple Output (SIMO) and Multiple Input Multiple Output (MIMO) Systems.

traveled. The remaining reverberative echo component received at each microphone should be different, provided the microphones are spaced sufficiently far apart. More details about blind deconvolution are presented in section 2.1.

1.1.1 Acoustic Reverberation

From a psychoacoustic point of view, an echo presents a challenging scenario. Human hearing can discern a speaker's voice when listening in a reverberant environment, because of the signal processing capability of the brain. However, when listening to a recording of a source in a reverberant environment, the brain has no spacial reference and the source simply comes through as poor sounding audio. Moreover, the reverberation is naturally highly correlated with the source, making it a type of noise which human hearing is especially sensitive to. Thus it is highly desirable to reduce the echo as much as possible.

The physics of room acoustics contribute significantly to the difficulty of blind deconvolution of speech. There are two key factors that complicate the typical acoustic impulse response. First, the space is typically small, such as an office or a conference room. This reduces the opportunity for the sound intensity to dissipate due to inverse square law fading before the sound is reflected and re-received at the receiver. Secondly, typical office walls are highly reflective. In the range of speech frequencies, reflection coefficients can be above 95% for brick and plaster wall materials and ceiling materials are around 70-80% [21][24].

The combined result of these two factors is that initial reflections are fairly strong, but the tail end of the impulse response continues for quite a while.

Figure 1.2 depicts a typical measured room acoustic impulse response. The most important feature is the exponentially decaying envelope of the response. In the figure, noticeable portions of the echo persist beyond 120ms.

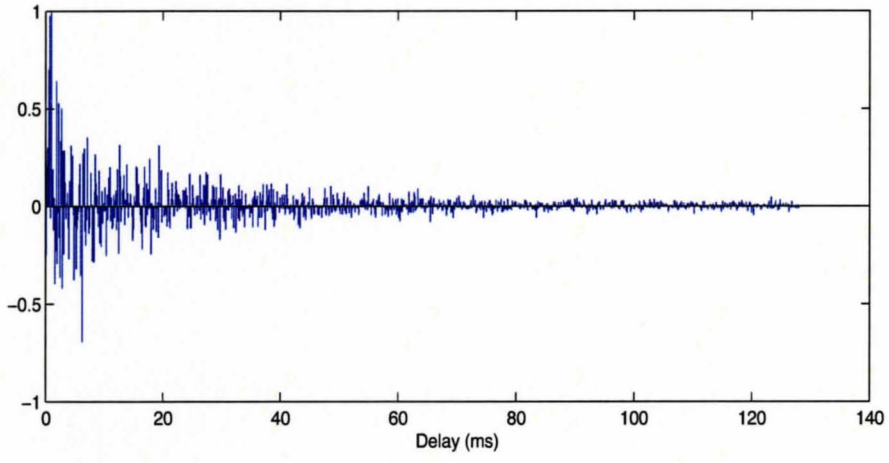

Figure 1.2: A typical acoustic impulse response measured in a small room.

1.1.2 Implications for Blind Deconvolution

The long decaying tail of an acoustic impulse response has significant implications for blind deconvolution. It creates the following key complications that a blind deconvolution algorithm must deal with. It is worth noting that some of these apply not only to blind deconvolution, but also channel identification in an acoustic environment, where the source, but not the channel is known, and non-blind deconvolution, where the channel is known and must be inverted to recover the source.

1. **Long length** - The long length makes solutions to the problem computationally intensive. It also affects the rate of convergence of continuously adaptive algorithms.

2. **Non-minimum phase** - The majority of multichannel blind deconvolution methods model the channel as an FIR filter. Acoustic impulse responses are typically not minimum phase, implying that even with the channels identified exactly, methods cannot rely on applying an IIR inverse filter since it will not be stable.

3. **Ill defined length** - The long decaying tail makes the exact length of the channel difficult to define. A number of blind deconvolution methods rely on knowing the length of the channel exactly. The end of the tail of an Acoustic channel is difficult to distinguish from any noise floor and any method used to recover the source must be robust against incorrectly estimated channel length.

4. **Effects difficult to distinguish in noise** - A related issue to the previous item, is that the small contribution of the individual elements in the tail may be hard to identify in a noisy environment.

5. **Possibility of common zeros or near common zeros** - The implications of the observation channels having common zeros are discussed in section 2.1. It suffices to say at this point that channels with zeros in common or situations where there are nearly common zeros cause difficulties with many blind deconvolution algorithms.

6. **Possibly non-stationary** - In a realistic environment, the source may be moving or the environment may otherwise be changing. This makes the channels time varying. Any method used must be capable of tracking a changing source adaptively or assume a block-stationary approach.

The combined effect of these complications makes acoustic blind dereverberation a challenging problem for which not all methods are suited. There are however some well known solutions to the first two items. Item 1 can be partially resolved through the use of filter banks to decompose the problem into subbands. This is known to improve the rate of convergence of some adaptive algorithms, and can also be used to reduce the computational complexity [38]. To resolve item 2, almost all multichannel blind deconvolution methods rely on methods that do not invert individual channels to recover the source. There are a variety of methods for this, references for which are given in section 2.1.

1.2 Problem Definition

With this background information the multichannel blind deconvolution problem is formally defined as follows: given a series of observations from N receivers over time 1 to T,

$$\{y_1^{(1)}, \cdots, y_T^{(1)}, y_1^{(2)}, \cdots, y_T^{(2)}, \cdots, y_1^{(N)}, \cdots, y_T^{(N)}\} \tag{1.1}$$

estimate $s_1, \cdots, s_t, \cdots, s_T$, where

$$\begin{aligned} y_t^{(1)} &= h^{(1)} * s_t + w_t^{(1)} \\ y_t^{(2)} &= h^{(n)} * s_t + w_t^{(2)} \\ &\vdots \\ y_t^{(N)} &= h^{(N)} * s_t + w_t^{(N)}. \end{aligned} \tag{1.2}$$

The operator $*$ represents convolution, $h^{(i)}$ are FIR filters representing the channel from the common source to the i-th microphone, with properties similar to the acoustic impulse response of a room and $w^{(i)}$ is white Gaussian noise with variance $\sigma_w^{(i)2}$. The source signal s is assumed to be representative of speech. The possibility of a non-stationary environment, where h is a function of time, will also be considered.

The specific variation of this problem that will be examined is identifying s_t given $\{y_1^{(1)}, \cdots, y_t^{(1)}, y_1^{(2)}, \cdots, y_t^{(2)}, \cdots, y_1^{(N)}, \cdots, y_t^{(N)}\}$, for t sequentially varying from 1 to T.

1.3 Contribution

This thesis accomplishes a number of tasks in demonstrating a particle filter solution to the blind deconvolution problem. Most notably,

- Formulation of the multichannel blind deconvolution problem as a non-linear, joint state-parameter estimation problem.
- Transformation of the problem into a non-linear state estimation problem and formulation of a particle filter solution to that problem.
- Reformulation of the solution to take advantage of linear-Gaussian substructure in the problem using a marginalized particle filter.
- Integration of a classical blind deconvolution method into the marginalized particle filter to increase the quality of the estimate.
- The estimator is demonstrated to address some of the challenges to acoustic blind dereverberation identified in section 1.1.2.

1.4 Organization

The remaining chapters of this thesis will provide more background information, describe the formulation of the problem and solution, and show experimental results for the algorithm that is developed.

Chapter 2 provides background information on the problem. It contains a brief literature review of multichannel blind deconvolution methods, along with an explanation of the necessary criteria for blind deconvolution. There is a brief review of speech modeling as it is necessary for the development of the algorithm in chapter 3. Also, recursive Bayesian estimation methods, including particle filters are described.

Chapter 3 describes the buildup of a blind deconvolution algorithm based on particle filtering. An initial algorithm is described, and then improved upon.

Chapter 4 presents the details of initializing and configuring the algorithm. Experimental results are then presented, with reference to the challenges of blind dereverberation as discussed in section1.1.2.

Chapter 5 summarizes the conclusions drawn from this work. Potential areas for future work are also discussed.

Chapter 2

Background

2.1 Blind Deconvolution

As discussed in chapter 1, blind deconvolution is the estimation of a source, given only observations through a channel, with no knowledge of the channel or source. The source can be identified within a scaling ambiguity, resulting from the indistinguishability of a scaling factor on the source from its inverse applied to the channel.

A subset of blind deconvolution is methods dealing strictly with deconvolution of time domain signals (as opposed to images, etc.) and within that is a further subset dealing with single sources observed through multiple channels. The most significant property of multichannel blind deconvolution is that, under conditions that are discussed later in this chapter, it is possible to perfectly recover a source using multichannel deconvolution, something that is not possible in single channel methods.

There are a number of methods for multichannel blind deconvolution that have been developed. These methods include a great deal of variety in the relationships that are exploited and the algorithms used. Some of the most widely cited approaches range from stochastic gradient descent [2], methods based on second order statistics [35] and methods based on direct algebraic relationships such as [49]. An overview of the most common approaches can be found in the survey papers [32], [44] and [1]. Of particular note are the particle filter method of [13] and the cross relation method of [49] upon which the work in this thesis is partially based.

2.1.1 Multichannel Deconvolution

An interesting result that forms the basis of multichannel blind deconvolution is that a SIMO system with FIR channels can be inverted perfectly with FIR filters. This is regardless of whether the channel responses are minimum phase or individually have FIR inverses. The only condition needed for this result is that the channels have no common zeros. This result was first observed in [34].

In a single FIR channel system, inversion of the system, even with the channel impulse response known perfectly is difficult. The natural inverse of an FIR channel is an all pole, IIR filter; an FIR inverse may exist but not generally. The IIR inverse suffers from the numerical stability problems for long IIR filters. More importantly, acoustic impulse responses are generally not minimum phase, implying that the inverse filter is not causal and stable. The inverse IIR filter may be approximated with an FIR filter, but this not perfect reconstruction and the filter will not be able to recover the frequencies lost to the zeros of the channel response. Thus single channel deconvolution cannot generally recover a source from observations through a channel.

Bezout Identity For multichannel systems, inversion can be accomplished using FIR filters using the Bezout Identity. The Bezout identity is the result for two polynomials $p(z)$ and $q(z)$ of length M and N respectively that there exist two unique polynomials, $\tilde{p}(z)$ of length $M-1$ and $\tilde{q}(z)$ of length $N-1$, such that

$$p(z)\tilde{p}(z) + q(z)\tilde{q}(z) = 1. \tag{2.1}$$

This is true if $p(z)$ and $q(z)$ have no common zeros [27]. The result can also be generalized to any number of polynomials. Additionally, the identity holds for $\tilde{p}(z)$ and $\tilde{q}(z)$ of longer lengths, but the solution is no longer unique.

When considered in terms of polynomials in z^{-1} representing FIR filters, the Bezout identity implies that perfect reconstruction of the original is possible by using FIR filters. This is done by identifying the equalizing filters $\tilde{p}(z)$ and $\tilde{q}(z)$ and applying them to the observed outputs as shown in figure 2.3.

Details of how to identify the inverting filters are given in Appendix 1. It should be noted that neither $\tilde{p}(z)$ nor $\tilde{q}(z)$ are true inverting filters for $p(z)$ and $q(z)$ individually, only the combination inverts the system. More importantly, the inversion is independent of whether the channel responses are minimum phase or individually have meaningful FIR inverses.

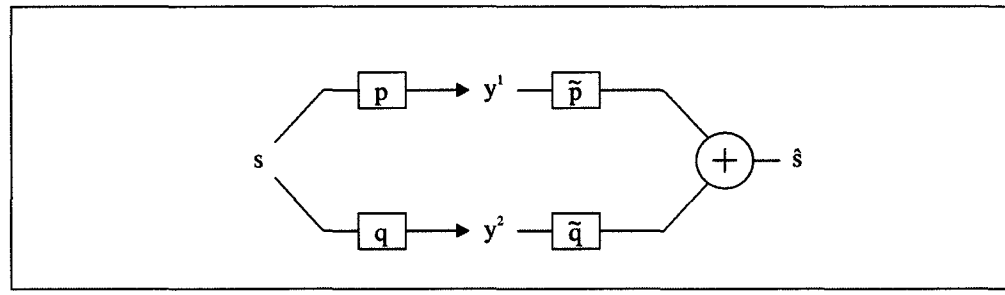

Figure 2.3: Bezout identity based multichannel equalization

2.1.2 Necessary Conditions

No Common Zeros

The Bezout identity gives rise to one of the necessary conditions for multichannel blind deconvolution. The requirement of no common zeros between channels is derived in a variety of ways in the blind deconvolution literature, for example in [49] and [41]. It is worth noting a simple physical explanation for the requirement here. In the noise free case, over a window of length N, the output of the convolution can be written in vector-matrix form as follows:

$$\begin{bmatrix} y_t^{(1)} \\ y_{t-1}^{(1)} \\ \vdots \\ y_{t-N}^{(1)} \\ y_t^{(2)} \\ y_{t-1}^{(2)} \\ \vdots \\ y_{t-N}^{(2)} \\ y_t^{(3)} \\ \vdots \end{bmatrix} = \begin{bmatrix} h_1^{(1)} & h_2^{(1)} & \cdots & \cdots & h_L^{(1)} & 0 & 0 \\ 0 & \ddots & & & & \ddots & \\ \vdots & & & & & & \\ 0 & 0 & h_1^{(1)} & h_2^{(1)} & \cdots & \cdots & h_L^{(1)} \\ h_1^{(2)} & h_2^{(2)} & \cdots & \cdots & h_L^{(2)} & 0 & 0 \\ 0 & \ddots & & & & \ddots & \\ \vdots & & & & & & \\ 0 & 0 & h_1^{(2)} & h_2^{(2)} & \cdots & \cdots & h_L^{(2)} \\ h_1^{(3)} & h_2^{(3)} & \cdots & \cdots & h_L^{(3)} & 0 & 0 \\ \vdots & & & & & & \end{bmatrix} \begin{bmatrix} s_t \\ s_{t-1} \\ s_{t-2} \\ \vdots \\ s_{t-N} \end{bmatrix} \tag{2.2}$$

$$\mathbf{y}^\star = \mathbf{H}^\star \mathbf{s}. \tag{2.3}$$

The structure of the matrix $\mathbf{H}^\star$ is known as a generalized Sylvester matrix or composite channel matrix[5]. Assuming a perfect knowledge of the channel impulse responses and the output $\mathbf{y}^\star$, it is clear that the input can only be perfectly reconstructed if $\mathbf{H}^\star$ is invertible.

Besides its use in expressing convolution, the Sylvester matrix has a long established property for testing polynomials for common zeros. If the coefficients of multiple polynomials are placed in a Sylvester matrix form, then all the polynomials have no common zeros if and only if the matrix has full rank. Since impulse responses are simply polynomials in z^{-1}, this and previous assertion implies that blind deconvolution is only possible if and only if the channels have no common zeros.

Moreover, as will become apparent in this thesis, a common zero becomes an unresolvable ambiguity, similar to the amplitude ambiguity in blind deconvolution. It cannot be distinguished between belonging to the source or belonging to the channels. In the case of an all-pole source, the common zero can encounter pole-zero cancellation between the source and the channels that results in the source being unidentifiable based on the observations.

Source Diversity

An additional requirement is that the channel can only be identified perfectly if the source has a sufficiently high level of complexity. The physical interpretation is that the source must stimulate the entire range of frequency components of the channel for the channel to be identified. This results in the commonly cited requirement of a linear complexity of two times the channel length or that the source have at least as many unique modes as the length of the channel[44][49]. This requirement is only necessary for blind deconvolution methods that require complete identification of the channel response.

Qualification of the Necessary Conditions

It is important to note that these necessary conditions are only imposed if the source is to be identified perfectly using a deterministic approach. For statistical methods more akin to single channel blind deconvolution methods, these necessary conditions are relaxed [44].

2.2 Acoustic Impulse Response Properties

The complications for acoustic blind dereverberation caused by acoustic impulse responses (section 1.1.2) have an explanation in terms of the Sylvester matrix discussed earlier.

The near common zeros and the exponential decay, specifically the large number of small values at the end of the impulse response relative to the large values at the start, has an adverse effect on condition the Sylvester matrix. Figure 2.4 shows the singular values of the Sylvester matrix for a pair of measured acoustic channels. In this example a number of the smallest singular values are significantly below the largest singular, the condition number of the Sylvester matrix is 1.8×10^6. While not singular, the matrix is ill conditioned, and the impulse responses would be difficult to identify or invert in the presence of noise.

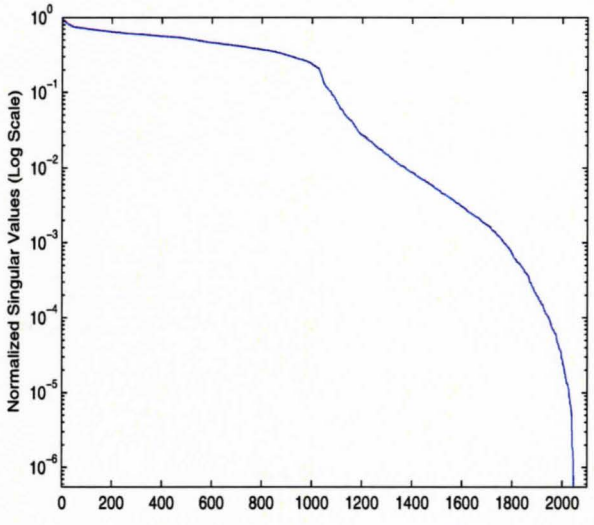

Figure 2.4: Singular values of the Sylvester matrix of two room acoustic impulse responses.

Thus typical reverberant acoustic environments such as small rooms fail to meet the necessary criteria for deterministic blind deconvolution.

2.3 Speech Modeling

In the estimator that is developed later in this work, a model for the source is required. With that intent in mind, the following review of speech modeling is presented.

Speech is the result of resonance of airflow through the vocal tract. The form of the driving air excitation is the basis of the main subdivisions of speech sounds [14].

- *Unvoiced* sounds are created by a constriction of the vocal tract causing the air to become turbulent. The resulting driving air into the resonant structures is similar to Gaussian noise. Examples of voiced sounds include the *sh* sound in "shade" and *th* in "thus."

- *Voiced* sounds are the result of the glottis, a portion of the vocal chords, vibrating. This creates excitation similar to a pulse train. When this resonates in the vocal tract, the result is sounds such as the m in "him" and a in "arm."

The transition between silence, voiced and unvoiced sounds can be gradual as in the word "she," or abrupt. In latter case, *plosive* sounds can be considered another form of excitation. They are quickly are followed by voiced or unvoiced sounds, but begin as a buildup of air that is suddenly released. The buildup can be within the mouth as in p in "put," or the throat as in g in "get." This sudden change requires that any model be able to quickly adapt to changing parameters. Waveforms corresponding to voiced and unvoiced speech signals are shown in figure 2.5.

The resonant behaviour can be modeled over short periods of time as an autoregressive (AR) process. Most methods of speech modeling rely on some form of AR model; from linear prediction modeling [21] to AR based noise enhancement methods [45][37]. The AR coefficients are typically modeled as either block-stationary, as in the case of linear prediction, or continuously time varying [37][46].

Thus speech can be modeled as a driving source, either a pulse train or Gaussian noise, followed by a continuously time varying or block-stationary AR process.

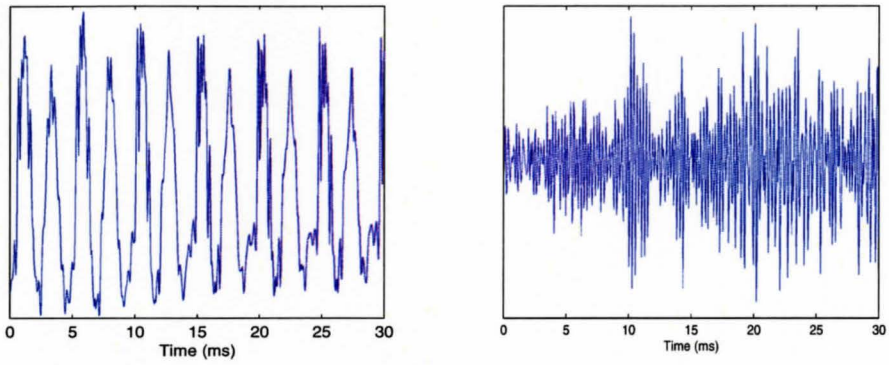

Figure 2.5: Voiced speech *ou* as in "your" (left) and unvoiced speech *sh* as in "she" (right) [19].

2.4 Sequential State Estimation

The state estimation problem is defined as identifying the distribution of the changing state $\mathbf{x}$ given a set of related observations $\mathbf{y}$: $p(\mathbf{x}|\mathbf{y})$. Once the density is available, any number of useful estimates can be taken by calculating

$$\mathrm{E}[h(\mathbf{x})] = \int h(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x}. \tag{2.4}$$

More specifically, the density of interest is usually based on a time series of past observations. Let $y_{1:t} = \{\mathbf{y}_t, \mathbf{y}_{t-1}, \cdots, \mathbf{y}_1\}$, the set of observations from time 1 to t.

In this thesis, the problem will be formulated in the form of a sequential estimation problem. Of interest will be the estimate of the state at time t, x_t, having already calculated the estimate of x_{t-1}. Then the problem of estimating $p(\mathbf{x}_t|y_{1:t})$ is known as filtering, and estimating $p(\mathbf{x}_{t+1}|y_{1:t})$ is known as prediction. There is also the related problem of $p(\mathbf{x}_{t_i}|y_{1:t})$, where $t_i < t$, this is known as smoothing [6]. Within the scope of this work, the focus will be on the filtering problem. This is of interest in online systems where the estimate of the current state is desired, given the most recent observation.

The discussion is restricted to first order Markov systems. This allows certain algebraic simplifications, but is still a general enough assumption to encompass models of interest. Additionally, the system is expressed in terms of a generalized state space model. Thus the model follows the following two

equations

$$x_{t+1} = f_t(x_t, v_t) \qquad \text{State transition} \quad p(x_{t+1}|x_t) \tag{2.5}$$
$$y_t = g_t(x_t, w_t) \qquad \text{Observation} \quad p(y_t|x_t) \tag{2.6}$$

where v_t and w_t are random variables representing the process noise and observation noise respectively.

This type of estimation is usually based around manipulations of the probabilities based on Bayes rule. Hence the name, Bayesian estiamtion. Details on Bayesian estimation methods for linear-Gaussian systems can be found in [6]. This reference also covers some nonlinear and non-Gaussian methods. Thorough surveys of popular methods of Bayesian estimation for nonlinear and non-Gaussian systems can be found in [23] and [10].

The state estimation problem can be formulated in an iterative form. Expanding and applying Bayes rule, $p(x_t|y_t)$ can be written as

$$p(x_t|y_{1:t}) = \frac{p(y_{1:t}|x_t)p(x_t)}{p(y_{1:t})} \tag{2.7}$$
$$= \frac{p(y_t, y_{1:t-1}|x_t)p(x_t)}{p(y_t, y_{1:t-1})} \tag{2.8}$$
$$= \frac{p(y_t|y_{1:t-1}, x_t)p(y_{1:t-1}, x_t)}{p(y_t, y_{1:t-1})} \tag{2.9}$$
$$= \frac{p(y_t|y_{t-1}, x_t)p(y_{t-1}, x_t)}{p(y_t|y_{t-1})p(y_{1:t-1})} \tag{2.10}$$
$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}. \tag{2.11}$$

The last step stems from the Markov property. In the last line, the distribution of interest $p(x_t|y_{1:t})$ is known as the posterior distribution (*after* the observation has been made). The factor $p(y_t|x_t)$ is known as the likelihood (the *likelihood* of seeing a particular observation, given the state). $p(x_t|y_{1:t-1})$ is known as the prior (*before* the observation was made. In this case it is a forward prediction of the distribution of x, later on it is reformulated as $p(x_{t-1}|y_{1:t-1})$. Finally, the denominator $p(y_t|y_{1:t-1})$, known as the evidence, is just a normalizing constant and can be obtained by integrating the numerator.

From this factorization and the Chapman-Kolmogorov equation, to give

the prediction density $p(x_t|y_{1:t-1})$, the following recursion can be formed

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \qquad \text{Prediction} \qquad (2.12)$$

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t} \qquad \text{Correction} \qquad (2.13)$$

The recursion is initiated by an initial prior $p(x_0)$. From the distribution $p(x_t|y_{1:t})$, any estimates of interest can be taken:

$$\mathrm{E}[f(x_t)|y_{1:t}] = \int f(x_t)p(x_t|y_{1:t})dx_t \qquad (2.14)$$

This method, recursive Bayesian estimation, is exact; however, the terms are difficult to calculate analytically and exact solutions are limited to a handful of special cases.

2.4.1 Kalman Filter

A well known exact analytical solution to equations 2.12 and 2.13 exists if the system to be estimated is a linear–Gaussian state space model, i.e. it can be expressed in the following form:

$$\mathbf{x}_{t+1} = \mathbf{A}_t\mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{v,t}) \qquad (2.15)$$

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{x}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{w,t}) \qquad (2.16)$$

where the gain matrices $\mathbf{A}$ and $\mathbf{C}$, and covariance matrices $\mathbf{\Sigma}_v$ and $\mathbf{\Sigma}_w$ are known and $\mathbf{v}$ and $\mathbf{w}$ are uncorrelated. The solution, developed in 1960 by Rudolf Kalman [28], is known as the Kalman filter. It provides the minimum MMSE estimate of the filtering and prediction distribution for linear–Gaussian systems.

For the following discussion, the following notation is introduced: $\mathbf{x}_{t-1|t-1}$ implies the estimate of $\mathbf{x}_{t-1}$ given the observation $\mathbf{y}_{t-1}$. Similarly $\mathbf{x}_{t|t-1}$ is the prediction of $\mathbf{x}_t$ given $\mathbf{y}_{t-1}$. The following steps are relatively simple to derive; an explanation of the process can be found [6]. As described for general recursive Bayesian state estimation in the previous section, the estimator has a prediction and correction step. First, the prediction step answers the problem of calculating the MMSE error estimate of $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$. Given an initial

estimate, with Gaussian probability distribution with mean $\mathbf{x}_{t-1|t-1}$ and with covariance $\mathbf{P}_{t-1|t-1}$, the one time step forward estimate is

$$\mathbf{x}_{t|t-1} = \mathbf{A}_{t-1}\mathbf{x}_{t-1|t-1} \tag{2.17}$$

$$\mathbf{P}_{t|t-1} = \mathbf{A}_{t-1}\mathbf{P}_{t-1|t-1}\mathbf{A}_{t-1}^T + \mathbf{\Sigma}_{v,t-1}, \tag{2.18}$$

thus

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) = \mathcal{N}(\mathbf{x}_{t|t-1}, \mathbf{P}_{t|t-1}). \tag{2.19}$$

This is equivalent to propagating the Gaussian pdf of $x_{t-1|t-1}$ through equation (2.15).

The correction step combines the forward estimate from above with the observation $\mathbf{y}_t$. First, two factors are calculated, the innovation covariance

$$\mathbf{S}_t = \mathbf{C}_t\mathbf{P}_{t|t-1}\mathbf{C}_t^T + \mathbf{\Sigma}_{w,t} \tag{2.20}$$

and the Kalman gain

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{C}_t^T\mathbf{S}_t^{-1}. \tag{2.21}$$

Then the filtering distribution is described by

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} - \mathbf{K}_t(\mathbf{y}_t - \mathbf{C}_t\mathbf{x}_{t|t-1}) \tag{2.22}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{C}_t\mathbf{P}_{t|t-1} \tag{2.23}$$

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = \mathcal{N}(\mathbf{x}_{t|t}, \mathbf{P}_{t|t}). \tag{2.24}$$

The analytical solution provided by the Kalman filter is possible because linear transformations of Gaussian distributions yield other Gaussian distributions. Other examples where this is possible are limited.

Non-linear Extensions

Although the Kalman filter is limited to linear–Gaussian systems, there are a number of variations of the Kalman filter than provide good approximations in certain nonlinear cases.

The extended Kalman filter (EKF) assumes that the state transition and observation equations are differentiable functions. It then linearizes the equations and uses the Kalman filter to produce estimates based on the linearization [48].

Another option is the unscented Kalman filter (UKF), first presented in

[26]. This variation propagates a pattern of points through the nonlinearity, which allows for a more accurate Gaussian approximation to the state distribution[47].

Despite these and other extensions, Kalman filter based methods are often a poor approximation in highly non-linear and non-Gaussian cases.

2.4.2 Particle Filter

Particle filtering, also known as sequential Monte Carlo methods, are an approximate solution to the recursive estimation problem. It has been described as a "randomized adaptive grid approximation"[20] solution to the problem. The basic forms of particle filters were developed in the 1950s, but the modern incarnation is generally attributed to [40].

The basis of particle filters is to create a discrete approximation of the posterior by sampling the distribution

$$p(x_t|y_{1:t}) \approx \sum_{p=1}^{N_{part}} w_p \delta(x_t - x_{p,t}) \tag{2.25}$$

Where N_{part} is the number of discrete approximations, known as particles. Each particle has a state sample $x_{p,t}$ and a weight w_p associated with it. The states are spread over the domain of $p(x_t|y_{1\ldots t})$ and the weights proportional to $p(x_{p,t}|y_{1\ldots t})$, normalized such that $\sum w_p = 1$. The weight is calculated to represent the likelihood that the particle state is the true state.

The estimate of the state can then be calculated based on the weights assigned to the particles by application of the comb property of the delta function to equation (2.14)

$$\mathrm{E}[x_t|y_{1:t}] \approx \sum_{p=1}^{N_{part}} w_p x_{p,t}. \tag{2.26}$$

Since drawing directly from $p(x_t|y_{1\ldots t})$ is difficult, a recursive solution can be formed where calculation of the weights is done iteratively [17]. Similar to equations (2.12) and (2.13), a different recursion for $p(x_t|y_{1\ldots t})$ can be formed

by factoring it as

$$p(x_t|y_{1:t}) = p(x_{t-1}|y_{1:t-1})\frac{p(y_t|x_t)p(x_t|x_{t-1})}{p(y_t|y_{1:t-1})}. \tag{2.27}$$

In general, this is no more analytically tractable than (2.12) and (2.13). However, the density $p(x_{t-1}|y_{1:t-1})$ can be represented by a set of particles, p, from the discrete approximation: equation (2.25). Then the posterior density then can be updated by sampling particle states from $x_{p,t} \sim p(x_{p,t}|x_{p,t-1})$, which stems from (2.5), for each discrete state $x_{p,t}$, and modifying the particle weights through

$$w_{p,t} = w_{p,t-1}\frac{p(y_t|x_{p,t})}{p(y_t|y_{1:t-1})}, \tag{2.28}$$

where $p(y_t|x_{p,t})$ is derived from equation (2.6). The denominator can be eliminated by normalizing the particles

$$w_{p,t} = \frac{w_{p,t}}{\sum_p^{N_{part}} w_{p,t}}, \tag{2.29}$$

leaving the update equation

$$w_{p,t} = w_{p,t-1}p(y_t|x_{p,t}), \tag{2.30}$$

The recursion is initiated by setting all the weights equal to N_{part}^{-1} and sampling the states x_p from an initial distribution $p(x_0)$ corresponding to the initial estimate of the particle states. Thus the weights propagate the particle likelihood, resulting in the discrete approximation bypassing the intractable analytical calculations in (2.12) and (2.13). This propagation of the weights is the fundamental principle of particle filters.

Equation (2.30) can be generalized further by introducing the idea of importance sampling. The distribution $p(x_t|x_{t-1})$ may be difficult or inefficient to draw from, more significantly, it may be possible to draw from other distributions that provide more information. An arbitrary probability distribution, known as the importance function, $\pi(x)$, can be introduced into (2.28) by sampling from it, and modifying the update to

$$w_{p,t} = w_{p,t-1}\frac{p(y_t|x_{p,t})p(x_{p,t}|x_{p,t-1})}{\pi(x_{p,t})}. \tag{2.31}$$

This is valid provided that the support of $\pi(x)$ includes the support of $p(x_t|y_t)$ [15].

The resulting algorithm is known as Sequential Importance Sampling and is summarized in algorithm 1 [4].

Algorithm 1: Sequential Importance Sampling (SIS) [4]

begin
  // Initialize
  foreach *particle* $p = 1$ *to* P **do**
    Draw x_p from an initial prior distribution $p(x_0)$;
  end
  for $t \leftarrow 1$ *to* T **do**
    foreach *particle* $p = 1$ *to* N_{part} **do**
      // Correct
      $w_{p,t} \leftarrow w_{p,t-1} \frac{p(y_t|x_{p,t})p(x_{p,t}|x_{p,t-1})}{\pi(x_{p,t})}$;
    end
    foreach *particle* $p = 1$ *to* N_{part} **do**
      $w_p \leftarrow w_p\{\sum_p w_p\}^{-1}$; // Normalize
    end
    $\hat{x}(t) \leftarrow \sum_p w_p x_p$; // Estimate
    foreach *particle* $p = 1$ *to* N_{part} **do**
      // Predict
      $x_{p,t} \sim \pi(x_t)$;
    end
  end
end

The choice of importance function can vary depending on ease of computation or analytical tractability of certain distributions, a discussion of some options is in [17]. In this regard, equation (2.30) is a special case of (2.31), where $\pi(x) = p(x_t|x_{t-1})$. The next section will also identify a choice of importance function that can improve the stability of the estimator.

Particle filter estimates based on equation (2.26) will converge to the Bayesian (MMSE) estimate as $N_{part} \rightarrow \infty$. With a limited number of particles the estimate is biased, however in practice moderate numbers of particles produce good estimates. The exact number necessary is largely dependent on

the specific model to be estimated, especially on the size of the state variable. The number of particles needed to achieve a given estimation variance increases approximately exponentially with the dimension of the state, limiting application of particle filters in high dimensional problems. Proof of the convergence of particle filter is analyzed in detail in [12]. A simpler discussion based on convergence of importance sampling estimates is in [15].

Degeneracy and the Optimal Importance Function

As a particle filter algorithm progresses, the particle states generally spread out progressing recursively through the importance function, while the likelihood distribution remains comparatively narrow. Thus the particle states will tend to enter regions that have a very low likelihood. As a result, the importance weights of many these particles will tend to zero, while only the small number that remain in high likelihood regions will have significant weights. Proof that this is largely unavoidable is outlined in [17]. This phenomenon is known as degeneracy. It generally leads to poor estimation since the areas of high likelihood will be poorly sampled by the particles.

Particle filter degeneracy is pictured in figure 2.6. In the figure, the circle positions represent particle states, with the diameter representing the particle weight. When the states are propagated through $\pi(x_t|x_{t-1})$, they spread out to the point that the particles no longer have significant weights.

Typically the metric known as the effective sample size, related to the variance of the particle weights is used to quantify degeneracy. With normalized particle weights, the effective sample size is

$$N_{eff} = \frac{1}{\sum_{p=1}^{N_{particles}} w_p^2}. \tag{2.32}$$

Nominally, the particles will have approximately equal non zero weights, implying that the above value will be equal to the number of particles. In the degenerate case, most of the particles will have near zero values and the metric will fall to a small level indicating that less than 100% of the particles are providing useful information.

The degeneracy problem can be decreased to some extent by the choice of importance function. Use of $\pi(x) = p(x_t|x_{t-1}, y_t)$ can be shown to be optimal in that it minimizes the variance of the importance weights [17]. Use of the optimal importance function can reduce the number of particles required to

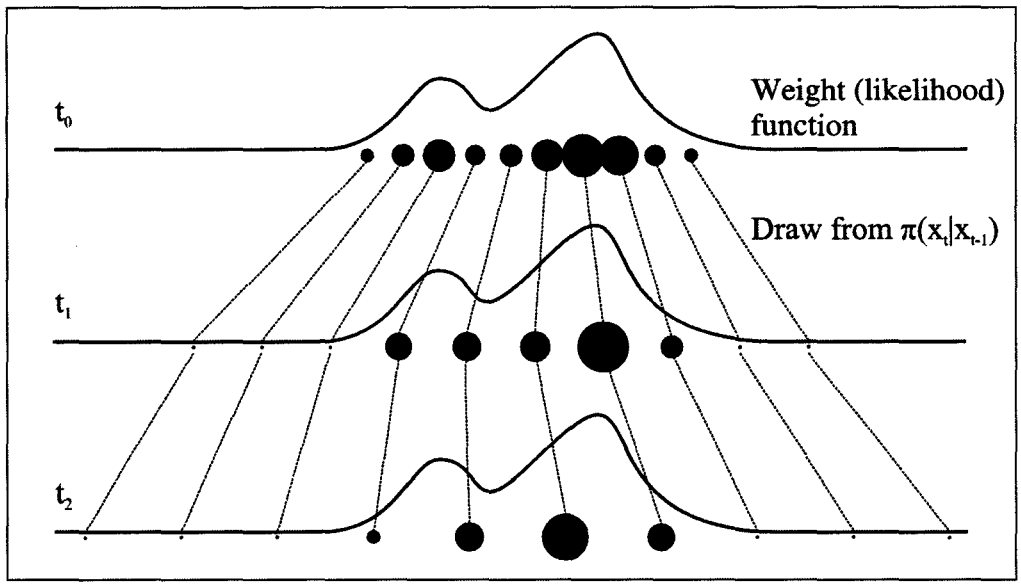

Figure 2.6: Degeneracy of a particle filter. The weights of most of the particles tend to zero.

achieve a given estimation variance. However, it does not solve the degeneracy problem, as the variance of the importance weights will still increase over time.

Resampling The solution to the degeneracy problem is to resample the particle distribution at each time step to better distribute the particle sampled states. There are a number of algorithms available, a number of which are discussed in [25], and each offers certain tradeoffs. Systematic resampling is recommended by some sources [25][4] because it is an intuitive and computationally simple algorithm. The method, detailed in algorithm 2, is to construct the cumulative density funciton (CDF) of the particle weights and then step through it with uniform steps, generating new particles corresponding to the current position in the CDF [4]. The result is that particles with high weights are duplicated, while those with low weights are discarded. The process is illustrated in figure 2.7. In the figure, the particles with low weights are eliminated, while the particles with high weights are duplicated proportionally to their weight. The weights of all the particles are then set to be equal.

Sequential importance sampling (algorithm 1) becomes the sampling – importance – resampling (SIR) algorithm with the addition of a resampling step

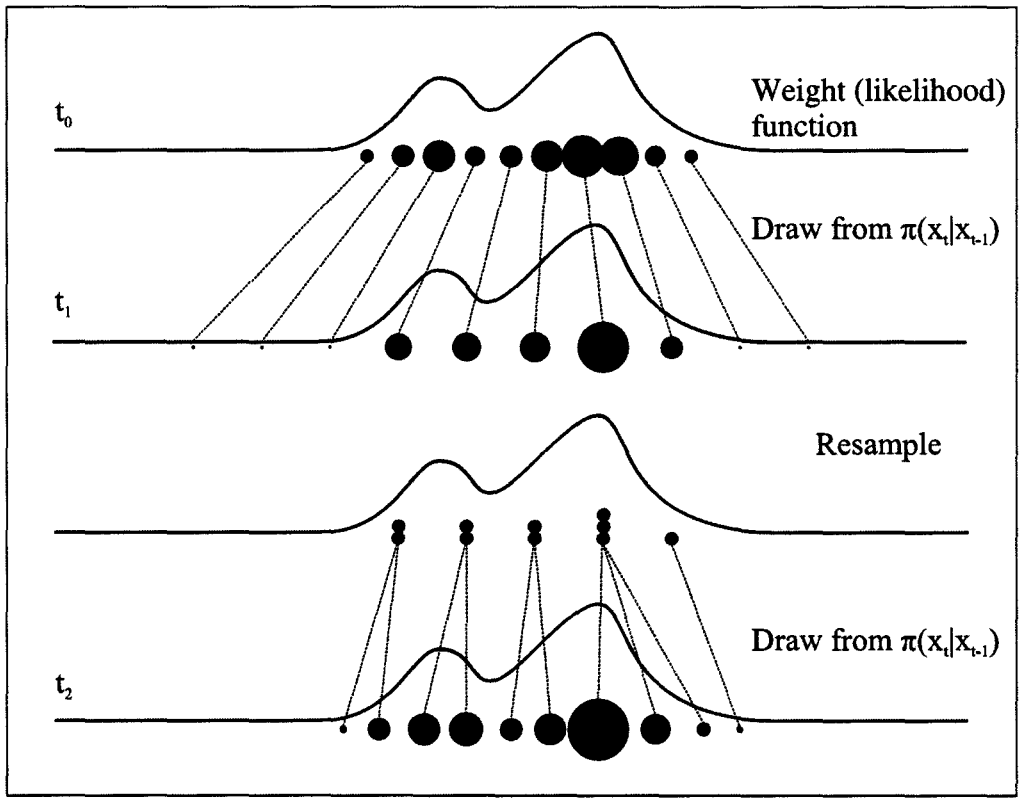

Figure 2.7: Resampling to eliminate degeneracy.

after the normalization of the particle weights. The traditional SIR algorithm [17] resamples only when the effective number of samples falls below a threshold, however the relatively low computational cost of the procedure means that it can be included at every iteration with little effect on the computational load.

In algorithm descriptions later in this work, the resampling step will be denoted as $\mathbf{x}_{p,t} \leftarrow R(w_p, \mathbf{x}_{p,t})$.

Resampling and Estimation From an estimation point of view, the resampled distribution is only an approximation of the original. Provided a sufficient number of particles, the estimate will still converge to the true distribution [7]. However it is advantageous to calculate the state estimate (equation (2.26)) for a given time step before the resampling step in the algorithm.

Algorithm 2: Systematic Resampling

Input: particles with weights w_i and state $\mathbf{x}_i$, $i = 1 \dots N_{part}$
Output: new particle weights $\tilde{w}_i$ and state $\tilde{\mathbf{x}}_i$
begin
  `// Calculate the CDF of the weights`
  Set $\mathbf{c}$ such that $\mathbf{c}(i) = \sum_{k=1}^{i} w_k$;
  `// Set an initial offset`
  Set $u_1 \sim U(0, 1)$;
  $i \leftarrow 1$;
  `// Iterate for all new particles`
  for $j \leftarrow 1$ *to* N_{part} **do**
    $u \leftarrow u_1 + (j-1)/N_{part}$;
    while $u > c(i)$ **do** $i \leftarrow i + 1$;
    $\tilde{\mathbf{x}}_j \leftarrow \mathbf{x}_i$;
    $\tilde{w}_j \leftarrow 1/N_{part}$;
  end
end

Resampling and Parallel Computing Finally, resampling forces all of the particles to interact at the same time. As is, the particle filter is more computationally expensive than the Kalman filter and its variants. The necessity of resampling prevents the possibility of distributing particles amongst parallel processors [17], thus limiting the possibilities for realtime application of particle filters. That said, there is some research into parallel approximations of resampling algorithms [8].

Diversity Resampling is necessary, however it has drawbacks. If the particle state changes relatively little from one step to the next, or the importance density does not track the likelihood well, the same particle states will be constantly duplicated by the resampling algorithm and the state range will not be sampled well. This leads to a lack of diverse samples known as sample impoverishment, illustrated in figure 2.8. The figure shows that the proposed importance density $\pi(x_t|x_{t-1})$ does not vary the particle states enough to place them where they would accurately cover the comparatively rapidly changing likelihood $p(y_t|x_t)$.

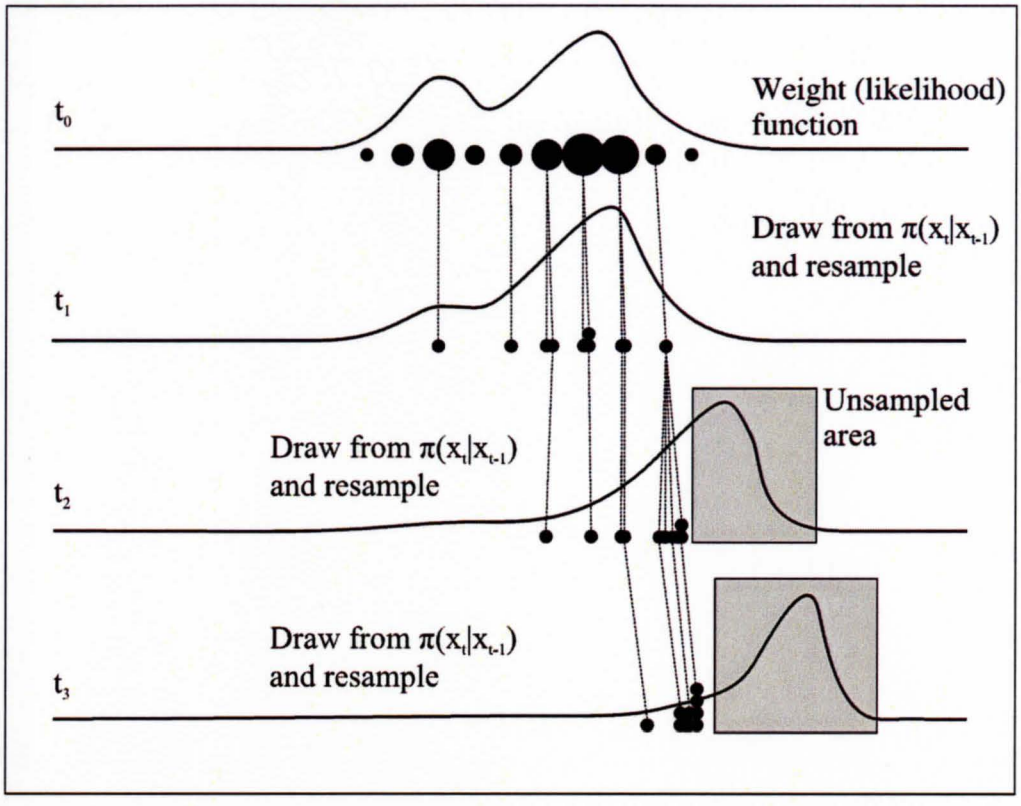

Figure 2.8: Sample impoverishment.

There are two solutions to this problem. The first is to estimate a continuous function to fit the discrete particle distribution and resample by sampling from the continuous distribution, this known as regularization [36]. If the model permits it, a simpler solution is to ensure that the importance distribution is such that the particles produce sufficient diversity when drawn to prevent impoverishment.

Further Reading More detailed descriptions of aspects of particle filters can be found in [15], [10] and [50].

With this background discussed, this thesis now continues with the application of this theory to the blind deconvolution problem. In the next chapter a particle filter formulation of the solution to the multichannel blind deconvolution problem is developed.

Chapter 3

Multichannel Blind Deconvolution using Particle Filters

In this chapter, the blind deconvolution problem is first expressed in terms of an observation model. This is coupled with a model for the source and the system is formulated as nonlinear state-space model. The blind deconvolution problem is expressed as a joint parameter-state estimation problem for the model, which is then transformed to a state estimation problem. A recursive filtering estimator for the system is then formulated using particle filtering.

The particle filter estimator is modified to take advantage of the structure of the model and a variation of the estimator is developed based on a marginalized particle filter. This is further improved upon by integrating a classical subspace based blind deconvolution method.

3.1 Observation Model

As described in chapter 1, the single input, multiple output system with FIR channels can be written as follows:

$$y^{(1)} = h^{(1)} * s + w^{(1)} \tag{3.1}$$
$$y^{(2)} = h^{(2)} * s + w^{(2)} \tag{3.2}$$
$$\vdots \tag{3.3}$$
$$y^{(N)} = h^{(N)} * s + w^{(N)}, \tag{3.4}$$

where $y^{(n)}$ are the received signals, $h^{(n)}$ are the channel impulse responses, s is the source signal and $w^{(n)}$ is the observation noise at each channel. Each of the noise sources is Gaussian, white noise, independent between the channels, with zero mean and variance $\sigma^{(n)2}$. It is worthwhile to note that this can be equally extended to a MIMO formulation (multiple sources) with channels from each source to each output, a description of that formulation can be found in [13].

In discrete time this can be expressed in vector-matrix form. First, the following definitions are made: the observation $\mathbf{y}_t$ is the vector of observations from all sensors at time t

$$\mathbf{y}_t = [y_t^{(1)}, y_t^{(2)}, \cdots, y_t^{(N)}]^T \tag{3.5}$$

the impulse response of channel i is the vector

$$\mathbf{h}^{(i)} = [h_1^{(i)}, h_2^{(i)}, \cdots, h_M^{(i)}]^T \tag{3.6}$$

the state $\mathbf{s}_{(M),t}$ to be the history of source samples from time t to $t - M$

$$\mathbf{s}_{(M),t} = [s_t, s_{t-1}, \cdots, s_{t-M}]^T \tag{3.7}$$

and the noise at time t for all channels is the vector

$$\mathbf{w}_t = [w_t^{(1)}, w_t^{(2)}, \cdots, w_t^{(N)}]^T \tag{3.8}$$

$\mathbf{w}_t$ is then Gaussian random variable with covariance matrix

$$\mathbf{\Sigma}_w = diag(\sigma^{(1)2}, \sigma^{(2)2}, \cdots, \sigma^{(N)2}). \tag{3.9}$$

The convolution operation at time t can then be expressed as

$$\begin{bmatrix} y_t^{(1)} \\ y_t^{(2)} \\ \vdots \\ y_t^{(N)} \end{bmatrix} = \begin{bmatrix} h_1^{(1)} & h_2^{(1)} & \cdots & h_M^{(1)} \\ h_1^{(2)} & h_2^{(1)} & \cdots & h_M^{(2)} \\ \vdots & & \ddots & \\ h_1^{(N)} & h_2^{(N)} & \cdots & h_M^{(N)} \end{bmatrix} \begin{bmatrix} s_t \\ s_{t-1} \\ \vdots \\ s_{t-M} \end{bmatrix} + \begin{bmatrix} w_t^{(1)} \\ w_t^{(2)} \\ \vdots \\ w_t^{(N)} \end{bmatrix}, \tag{3.10}$$

or simply as

$$\mathbf{y}_t = \mathbf{H}\mathbf{s}_{(M)t} + \mathbf{w}_t, \qquad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_w). \tag{3.11}$$

This is the vector-matrix form of a SIMO system with FIR observation channels that will be used throughout this chapter.

3.2 Source Model

Imparting prior knowledge based on the physical structure of the system to an estimator can have a very beneficial effect on the output [42]. Thus, applying a model to the source imparts a constraint on any estimate of the source that can improve the estimate. As discussed in section 2.3, speech can be described over short periods as being an autoregressive process driven by white noise or a pulse train.

It is proposed in this thesis that the pulse train assumption, though physically accurate for voiced speech, can be replaced with white noise. In the results presented in [46], voiced speech is accurately estimated by an AR process driven by white noise.

Thus the source is modeled as an AR process driven by white noise even if the speech is quasi-periodic. In discrete time, for an AR process of length K, this can be written as:

$$s_{t+1} = \sum_{k=0}^{K-1} a_k s_{t-k} + v_t, \tag{3.12}$$

where v_t is the driving noise; white, Gaussian which is assumed to be distributed with variance σ_v; and $a_1, a_2, \cdots, a_K$ are the AR coefficients. Alternatively, in vector-matrix form, this can be written as follows:

$$\begin{bmatrix} s_{t+1} \\ s_t \\ \vdots \\ s_{t-K-2} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \cdots & & a_K \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \vdots & \vdots \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_t \\ s_{t-1} \\ \vdots \\ s_{t-K-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} v_t. \tag{3.13}$$

More concisely, defining $\mathbf{a} = [a_1, a_2, \cdots, a_K]^T$ and $\mathbf{s}_{(K)t}$ similarly to 3.7 yields

$$\mathbf{s}_{(K)t+1} = \mathbf{A}\mathbf{s}_{(K)t} + \mathbf{v}_t \tag{3.14}$$

where

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{a}^T & \\ \hline \mathbf{I}_{K-1} & \mathbf{0}_{K-1,1} \end{array}\right], \qquad \mathbf{v} = \begin{bmatrix} v_t \\ \mathbf{0}_{K-1,1} \end{bmatrix}, \qquad v \sim \mathcal{N}(0, \sigma_v). \tag{3.15}$$

In this form, the first row of $\mathbf{A}$ performs the autoregressive dot product, while the remaining rows shift the source vector contents up by one.

3.3 System Model

Combining the source and observation models gives an overall model that is the basis of the blind estimation process

$$\mathbf{s}_{t+1} = \mathbf{A}\mathbf{s}_t + \mathbf{v}_t \tag{3.16}$$

$$\mathbf{y}_t = \mathbf{H}\mathbf{s}_t + \mathbf{w}_t. \tag{3.17}$$

Minor modifications to the original models are necessary to accommodate the mixed vector-matrix dimensions. The vector $\mathbf{s}_t$ is of length $L = \max(M, K)$, containing the longer of $\mathbf{s}_{(M)t}$ (see equation (3.11)) or $\mathbf{s}_{(K)t}$ (see equation (3.14)). Correspondingly, the AR and MA coefficients in $\mathbf{A}$ and $\mathbf{H}$ are zero padded to the longer length as necessary, depending on whether the AR coefficients or the MA coefficients are a longer sequence. Finally, the driving noise vector $\mathbf{v}_t$ is zero padded to the longer length as necessary. This makes $\mathbf{s}_t$ and

$\mathbf{v}_t$ now vectors of length L, the matrix $\mathbf{A}$ is an $L \times L$ matrix and $\mathbf{H}$ is $N \times L$.

This model is complete in the sense that it defines the relationship between the observed variable $\mathbf{y}$ and the state variable $\mathbf{s}$ and provides a description of the behaviour of the state variable.

3.4 Blind Deconvolution from a State Space Model

The goal is to calculate an estimate of the source $\mathbf{s}$. Based on the proposed model, this involves simultaneously estimating $\mathbf{s}$ along with the parameters $\mathbf{A}$, $\mathbf{H}$, $\mathbf{\Sigma}_w$ and σ_v. In the present form, this known as a joint state-space and parameter estimation problem.

The problem is well explored in literature. Older methods such as [29] rely on sampling the parameters from a prior distribution, then utilizing a particle filter to calculate the likelihood of a state estimate, conditional on the sampled parameter. However, these methods did not explore the parameter space after the initial sampling. Most of the recent approaches rely on forming an initial estimate of the parameters, using that to estimate the state, then modifying the parameters based on the observation likelihood, and then repeating the cycle. The authors in [3] demonstrated recursive maximum likelihood based on the expectation maximization (EM) algorithm. In [18] a recursive maximum likelihood method is developed around the particle filter. [30] uses a more conventional particle filter approach, but introduces a dynamic model to the sampled parameters to provide diversity and allow the parameters to be estimated as part of an extended state. This latter approach is the basis for the method that will be developed.

3.4.1 Model Redefinition

To apply the proposed method, the model is redefined as follows.

- Add a dynamic model to the AR coefficients, $\mathbf{a}$. This reflects the non-stationary nature of the AR coefficients of speech. It also provides variation from one time step to the next that creates diversity for the particle

filter. The chosen model is a random walk, forming a time varying autoregressive (TVAR) process as described in [46]:

$$\mathbf{a}_{t+1} = c_a \mathbf{a}_t + u_{a,t}, \qquad \mathbf{u}_a \sim \mathcal{N}(\mathbf{0}, \sigma_a). \tag{3.18}$$

The parameter c_a is a constant generally equal to 1, but left here for generality. The variance of u_a, σ_a is effectively a step size parameter, controlling the variance in the AR coefficients between each time step.

- Guarantee the stability of the AR portion by limiting the poles to be inside the unit circle. This is a more stringent requirement than what is necessary for stability, but it is a sufficient condition and is used successfully in [46]. The specific algorithm is defined in algorithm 3.

Algorithm 3: Constraining AR coefficients to ensure a stablity

Algorithm: Constrain AR Poles

Input: AR Coefficents $\mathbf{a}_t = [a_{1,t}, a_{2,t}, \cdots, a_{K,t}]$

begin

$P(z) \leftarrow 1 - \sum_{n=1}^{N} a_n(z^{-n})$;

Factor $P(z) = \prod_{n=1}^{N}(z^{-1} + p_n)$;

foreach *pole* p_n $n \in 1, \cdots, K$ **do**

if $|p_n| > 1$ **then** $\tilde{p}_n \leftarrow p_n/|p_n|$;

else $\tilde{p}_n \leftarrow p_n$;

end

$\tilde{P}(z) \leftarrow \prod_{n=1}^{N}(z^{-1} + \tilde{p}_n)$;

Expand $\tilde{P}(z) = 1 - \sum_{n=1}^{N} \tilde{a}_n(z^{-n})$;

$\mathbf{a}_t \leftarrow [\tilde{a}_{1,t}, \tilde{a}_{2,t}, \cdots, \tilde{a}_{K,t}]$;

end

- Add a dynamic model to the MA coefficients, $\mathbf{h}^{(i)}$. The time constant for changing the channel is generally slower than for the AR coefficients, however, change is still present. The random walk model is used again, it proved to be a useful model in [13] for this type of estimation:

$$\mathbf{h}_{t+1}^{(i)} = c_h \mathbf{h}_t^{(i)} + \mathbf{u}_{h,t}, \qquad i \in \{1, \cdots, N\}, \qquad \mathbf{u}_h \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_h). \tag{3.19}$$

- Add a dynamic model to the driving noise variance σ_w^2. Speech generally involves very large amplitude changes, typically over 40dB from the quietest tones to the loudest [22], thus applying a random walk model to the logarithm of the driving noise variance is appropriate. This also ensures that the variance is positive. This method is used in [46].

$$\phi_{v,t} = \ln(\sigma_{v,t}^2) \tag{3.20}$$

$$\phi_{v,t+1} = c_{\phi v}\phi_{v,t} + u_{\phi v,t}, \qquad u_{\phi v} \sim \mathcal{N}(0, \sigma_{\phi v}^2) \tag{3.21}$$

- A similar model is given to the observation noise

$$\phi_{wi,t} = \ln(\sigma_{wi,t}^2), \qquad i \in \{1, \cdots, N\} \tag{3.22}$$

$$\phi_{wi,t+1} = c_{\phi w}\phi_{wi,t} + u_{\phi w,t}, \qquad u_{\phi w} \sim \mathcal{N}(0, \sigma_{\phi w}^2). \tag{3.23}$$

By adding dynamic behaviour to the parameters, they can now be estimated as extended state variables. This also serves to provide the necessary diversity between time steps as discussed in section 2.4.2.

The constants in the model are now design parameters that can be estimated offline. They quantify measurable properties of acoustic systems that are relatively invariant for large classes of environments. It will also be shown that some act as estimator tuning parameters. In either case, the model parameters have been reduced to fixed quantities, while the parameters of the earlier model, the AR, MA parameters and noise variances, have been incorporated into the model state. Thus the joint state and parameter estimation problem has been transformed to a state estimation problem that can be tackled by conventional state estimation algorithms.

With this redefinition of the model, the state to be estimated is

$$\mathbf{x} = \{\mathbf{s}, \mathbf{h}, \mathbf{a}, \phi_v, \sigma_v, \phi_w^{(1)}, \cdots, \phi_w^{(N)}, \sigma_w^{(1)}, \cdots, \sigma_w^{(N)}\}, \tag{3.24}$$

which is perturbed by the random variables

$$\mathbf{e} = \{v, u_{\phi w}, u_{\phi v}, \mathbf{u}_a, \mathbf{u}_h\}. \tag{3.25}$$

The model can be generalized to a non-linear state space model, with a conditionally linear observation

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{e}_t) \tag{3.26}$$

$$\{\mathbf{H}_t, \mathbf{s}_t, \mathbf{\Sigma}_{w,t}\} \Leftarrow \mathbf{x}_t \tag{3.27}$$

$$\mathbf{y}_t = \mathbf{H}_t\mathbf{s}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim (\mathbf{0}, \mathbf{\Sigma}_{w,t}). \tag{3.28}$$

Recursive estimation of this type of system can be handled by a particle filter.

3.4.2 Bootstrap Particle Filter

The Sampling Importance Resampling particle filter algorithm requires the ability to evaluate and draw particles from a proposal distribution $\pi(\mathbf{x}_t)$. As discussed in section 2.4.2, the optimal choice to minimize the particle weight variance is $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$, however, this is difficult to obtain in this model. One alternative is to approximate the optimal importance function: [17] and [13] propose approximating it with a Gaussian distribution based on a linearization of the state equations using a method similar to the extended or unscented Kalman filter. As an alternative, the bootstrap approximation [4] is used where the importance density $\pi(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ is set to the prior density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The importance weight update then reduces to

$$w_{p,t} = w_{p,t-1}p(\mathbf{y}_t|\mathbf{x}_{p,t}). \tag{3.29}$$

Or, by resampling the particles at every iteration, the particle weights are forced to be equal. Thus

$$w_{p,t} = p(\mathbf{y}_t|\mathbf{x}_{p,t}), \tag{3.30}$$

which can be derived explicitly from the observation likelihood

$$p(\mathbf{y}_t|\mathbf{x}_{p,t}) = \mathcal{N}(\mathbf{H}_t\mathbf{s}_t, \mathbf{\Sigma}_{w,t}). \tag{3.31}$$

Drawing from the prior distribution is then a matter of propagating the previous estimate $\mathbf{x}_{p,t-1}$ from each particle through the state evolution model (3.26). In this process, the propagation of each particle state includes a sampled realization of the process noise for all of the random variables in the model equations. The resulting distribution of particle states is then a discrete approximation to $p(\mathbf{x}_t|\mathbf{x}_{t-1})$.

The complete bootstrap particle filter algorithm is summarized in algorithm 4.

Algorithm 4: Bootstrap Particle Filter

begin
  // Initialize
  foreach *particle* $p = 1$ *to* N_{part} **do**
    Draw $\mathbf{x}_{p,0}$ from an initial prior distribution;
  end
  for $t \leftarrow 1$ *to* T **do**
    foreach *particle* $p = 1$ *to* N_{part} **do**
      // Correct
      $w_p \leftarrow \mathcal{N}(\mathbf{H}_p \mathbf{s}_p, \mathbf{\Sigma}_{w,p})$;
    end
    $w_p \leftarrow w_p \{\sum_p w_p\}^{-1}$; // Normalize
    $\hat{\mathbf{x}}(t) \leftarrow \sum_p w_p \mathbf{x}_p$; // Estimate
    $\mathbf{x}_p \leftarrow R(w_p, \mathbf{x}_p)$; // Resample
    foreach *particle* $p = 1$ *to* N_{part} **do**
      // Predict
      // Sample $\mathbf{x}_{p,t} \sim p(\mathbf{x}_t | \mathbf{x}_{p,t-1})$
      Propagate $\mathbf{x}_p = f(\mathbf{x}_p, \mathbf{e}_t)$;
    end
  end
end

The bootstrap particle filter provides an estimator for the system and works well provided sufficient particles are used. However, the required number of particles grows exponentially with the number of dimensions in the estimated state. While the AR coefficients are relatively low dimensional, as are all of the parameters, the number of MA coefficients can be very large. To handle this, an alternative method is devised.

3.4.3 Marginalized Particle Filter

Before continuing this discussion, a change is made to the model. The output equation can be written equivalently as (since convolution is commutative):

$$\mathbf{y}_t = \mathbf{H}_t\mathbf{s}_t + \mathbf{w}_t \tag{3.32}$$
$$= \mathbf{S}_t\mathbf{h}_t + \mathbf{w}_t \tag{3.33}$$

where $\mathbf{S}$ is defined to provide the convolution product with $\mathbf{h}$, the column vector of the channel impulse responses

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}^T & \mathbf{0}_{1\times L} & \cdots & \mathbf{0}_{1\times L} \\ \mathbf{0}_{1\times L} & \mathbf{s}^T & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{1\times L} \\ \mathbf{0}_{1\times L} & \cdots & \mathbf{0}_{1\times L} & \mathbf{s}^T \end{bmatrix} \qquad \mathbf{h} = \begin{bmatrix} \mathbf{h}^{(1)} \\ \mathbf{h}^{(2)} \\ \vdots \\ \mathbf{h}^{(N)} \end{bmatrix}. \tag{3.34}$$

Returning to the design of the estimator, up to this point the substructure of the model has not been exploited. The state variable $\mathbf{x}$ can be partitioned into two portions:

$$\mathbf{h} \quad \text{and} \quad \mathbf{z} = \{\mathbf{s}, \mathbf{a}, \phi_v, \sigma_v, \phi_w^{(1)}, \cdots, \phi_w^{(N)}, \sigma_w^{(1)} \cdots, \sigma_w^{(N)}\}. \tag{3.35}$$

Likewise the random vector $\mathbf{e}$ can be partitioned into

$$\mathbf{u}_h \quad \text{and} \quad \mathbf{r} = \{v, u_{\phi w}, u_{\phi v}, \mathbf{u}_a\}. \tag{3.36}$$

Then the system equations can be partitioned as follows (for ease of notation, the matrices $\mathbf{S}$ and $\mathbf{\Sigma}_v$ are redefined to be functions of the state variables):

$$\mathbf{z}_{t+1} = g(\mathbf{z}_t, \mathbf{r}_t) \tag{3.37}$$
$$\mathbf{h}_{t+1} = c_h\mathbf{h}_t + \mathbf{u}_{h,t}, \qquad \mathbf{u}_{h,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_h) \tag{3.38}$$
$$\mathbf{y}_t = \mathbf{S}(\mathbf{z}_t)\mathbf{h}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_w(\mathbf{z}_t)). \tag{3.39}$$

Thus, conditional on $\mathbf{z}$ (the nonlinear state variables), equations (3.38) and (3.39) form a linear Gaussian system that can be solved with the Kalman filter. The nonlinear portion continues to be estimated using a particle filter. This approach is known as a marginalized particle filter or Rao-Blackwellized particle filter [39][16]. The method reduces the dimensionality of the space

that must be sampled with the particle filter, improving the estimate.

The derivation of the method for the blind deconvolution system is described below and stems directly from [39].

Prediction Beginning with an initial estimate for the nonlinear state at time t, sampled by each particle p, $\mathbf{z}_{p,t}$, and linear state $\mathbf{h}_{p,t}$, with covariance $\mathbf{P}_{p,t}$. The one time step ahead prediction of the nonlinear state can be sampled by drawing from $p(\mathbf{z}_{t+1}|\mathbf{z}_t)$, which is equivalent to evaluating equation (3.37) including a realization of the process noise $\mathbf{r}_t$ for each particle:

$$\mathbf{z}_{p,t+1|t} = g(\mathbf{z}_{p,t}, \mathbf{r}_t). \tag{3.40}$$

This is an identical process to the bootstrap particle filter, but with only the nonlinear portion of the state. The linear portion of the state, can be analytically estimated for each particle as

$$\mathbf{h}_{p,t+1|t} = c_h \mathbf{h}_{p,t} \tag{3.41}$$

with covariance

$$\mathbf{P}_{p,t+1|t} = c_h^2 \mathbf{P}_{p,t} + \mathbf{\Sigma}_h. \tag{3.42}$$

Correction Similarly, the correction portion of the estimator can be calculated as part Kalman filter and part particle filter. Because the prior density was sampled in the prediction step, given a new observation $\mathbf{y}_{t+1}$, the particle weight calculation, from equation (3.30), is

$$w_p = p(\mathbf{y}_{t+1}|\mathbf{z}_{p,t+1|t}, \mathbf{h}_{p,t+1|t}) \tag{3.43}$$

where $p(\mathbf{y}_{t+1}|\mathbf{z}_{p,t+1|t}, \mathbf{h}_{p,t+1|t})$ is a Gaussian distribution with mean

$$\mathbf{S}(\mathbf{z}_{p,t+1|t})\mathbf{h}_{p,t+1|t} \tag{3.44}$$

and covariance

$$\mathbf{S}(\mathbf{z}_{p,t+1|t})\mathbf{P}_{p,t+1|t}\mathbf{S}(\mathbf{z}_{p,t+1|t})^T + \mathbf{\Sigma}_w(\mathbf{z}_{p,t+1|t}). \tag{3.45}$$

The mean and covariance stem directly from the propagation of the linear prediction $\mathbf{h}_{p,t+1|t}$ through the observation equation, conditional on the nonlinear prediction $\mathbf{z}_{p,t+1|t}$. The linear portion is estimated with the Kalman filter, first

calculating the innovation covariance and the Kalman gain

$$\mathbf{E} = \mathbf{S}(\mathbf{z}_{p,t+1|t})\mathbf{P}_{p,t+1|t}\mathbf{S}(\mathbf{z}_{p,t+1|t})^T + \mathbf{\Sigma}_w(\mathbf{z}_{p,t+1|t}) \tag{3.46}$$

$$K = \mathbf{P}_{p,t+1|t}\mathbf{S}(\mathbf{z}_{p,t+1|t})^T\mathbf{E}^{-1}. \tag{3.47}$$

Then the particle estimate of the linear state is

$$\mathbf{h}_{p,t+1|t+1} = \mathbf{h}_{p,t+1|t} + K(\mathbf{y} - \mathbf{S}(\mathbf{z}_{p,t+1|t})\mathbf{h}_{p,t+1|t}) \tag{3.48}$$

with covariance

$$\mathbf{P}_{p,t+1|t+1} = \mathbf{P}_{p,t+1|t} - K\mathbf{S}(\mathbf{z}_{p,t+1|t})\mathbf{P}_{p,t+1|t}. \tag{3.49}$$

For MMSE estimation, the estimate of the linear portion of the state is calculated using the normal particle filter weighted sum method

$$\hat{\mathbf{h}}_{t+1|t+1} = \sum_{p=1}^{N_{part}} w_p \mathbf{h}_{p,t+1|t}. \tag{3.50}$$

By marginalizing out the linear portion, the state estimate requires fewer particles to accurately sample the state distribution.

Amplitude Ambiguity

The amplitude ambiguity of blind deconvolution cannot be resolved explicitly, in that the true source amplitude cannot be known without knowing the gain of the channel. However, it is possible to resolve the ambiguity such that relative changes in the source amplitude are estimated properly. This is done by constraining the gain of the channel. The exact gain is difficult to judge and varies with frequency, but it is approximately proportional to the norm of the channel impulse response.

In this algorithm, the composite channel vector $\mathbf{h}$ is constrained to have a L^2 norm of 1. The 2-norm was chosen on the basis of producing the highest quality results of the alternative L^1 and L^∞ norms. This results in inserting a normalization step following the Kalman prediction step of the algorithm:

$$\mathbf{h}_t = \mathbf{h}_t \|\mathbf{h}_t\|_2^{-1}. \tag{3.51}$$

The complete marginalized particle filter algorithm for this system is described in algorithm 5.

3.4.4 Subspace Projection

To further improve the quality of the estimate, knowledge of a subspace known to contain the system impulse response can be used. The impulse response estimate produced by the Kalman filter can be projected onto this subspace. The blind deconvolution method described in [49] can provide such a subspace.

Subspace Method for Blind Deconvolution

The authors in [49] discuss a method of blind deconvolution based on solving a cross relation between received signals. In the noise free case we have

$$y^{(i)} = h^{(i)} * s \tag{3.52}$$

$$y^{(j)} = h^{(j)} * s \tag{3.53}$$

Convolving the both sides of the first equation with $h^{(j)}$ yields

$$\begin{aligned} y^{(i)} * h^{(j)} &= (h^{(i)} * s) * h^{(j)} \\ y^{(i)} * h^{(j)} &= h^{(i)} * (s * h^{(j)}) \\ y^{(i)} * h^{(j)} &= h^{(i)} * y^{(j)} \\ y^{(i)} * h^{(j)} - h^{(i)} * y^{(j)} &= 0 \end{aligned} \tag{3.54}$$

This relationship can be expressed in matrix form. Note that the following is differs slightly from [49], but is equivalent. Define the matrix $\mathbf{Y}_{(L)}^{(i)}$ to be the $(T-L-1) \times L$ matrix

$$\mathbf{Y}_{(L)}^{(i)} = \begin{bmatrix} y_L^{(i)} & y_{L-1}^{(i)} & \cdots & y_2^{(i)} & y_1^{(i)} \\ y_{L+1}^{(i)} & y_L^{(i)} & \cdots & y_3^{(i)} & y_2^{(i)} \\ \vdots & \vdots & & \vdots & \vdots \\ y_{t-L}^{(i)} & y_{t-L-1}^{(i)} & \cdots & y_{t+1}^{(i)} & y_t^{(i)} \\ \vdots & \vdots & & \vdots & \vdots \\ y_T^{(i)} & y_T^{(i)} - 1 & \cdots & y_{T-L+2}^{(i)} & y_{T-L+1}^{(i)} \end{bmatrix}, \tag{3.55}$$

Algorithm 5: Marginalized Particle Filter

begin

  // Initialize

  foreach *particle* $p = 1$ *to* P **do**

    $\mathbf{z}_{p,0} \sim p(\mathbf{z}_0)$;

    $\mathbf{h}_{p,0} \sim p(\mathbf{h}_0)$;

  end

  for $t = 1$ *to* T **do**

    foreach *particle* $p = 1$ *to* N_{part} **do**

      // Correct - Particle Filter

      // calculate $p(\mathbf{y}_{t+1}|\mathbf{z}_{p,t+1|t}, \mathbf{h}_{p,t+1|t})$

      $w_p \leftarrow p(\mathbf{y}) \quad \mathbf{y} \rightarrow \mathcal{N}(\mathbf{S}(\mathbf{z}_p)\mathbf{h}_p, \mathbf{S}(\mathbf{z}_p)\mathbf{P}_p\mathbf{S}(\mathbf{z}_p)^T + \mathbf{\Sigma}_w(\mathbf{z}_p))$;

      $w_p \leftarrow w_p\{\sum_p w_p\}^{-1}$; // Normalize

      // Correct - Kalman Filter

      $\mathbf{E} \leftarrow \mathbf{S}(\mathbf{z}_p)\mathbf{P}_p\mathbf{S}(\mathbf{z}_p)^T + \mathbf{\Sigma}_w$; // Innovation Covariance

      $K \leftarrow \mathbf{P}_p\mathbf{S}(\mathbf{z}_p)^T\mathbf{E}^{-1}$; // Kalman Gain

      $\mathbf{P}_p \leftarrow \mathbf{P}_p - K\mathbf{S}(\mathbf{z}_p)\mathbf{P}_p$; // Covariance Estimate

      $\mathbf{h}_p \leftarrow \mathbf{h}_p + K(\mathbf{y} - \mathbf{S}(\mathbf{z}_p)\mathbf{h}_p)$; // State Estimate

    end

    $\hat{\mathbf{s}}_t \leftarrow \sum_p w_p\mathbf{s}_p$; // Estimate

    $\mathbf{z}_p \leftarrow R(w_p, \mathbf{z}_p)$; // Resample

    foreach *particle* $p = 1$ *to* N_{part} **do**

      // Predict - Particle Filter

      // propagate $\mathbf{z}_{p,t}$ through $p(\mathbf{z}_{t+1}|\mathbf{z}_{p,t})$

      $\mathbf{z}_p \sim g(\mathbf{z}_p, \mathbf{r})$;

      // Predict - Kalman Filter

      $\mathbf{h}_p \leftarrow c_h\mathbf{h}_p$; // State Prediction

      $\mathbf{P}_p \leftarrow c_h^2\mathbf{P}_p + \mathbf{\Sigma}_h$; // State Covariance Prediction

      $\mathbf{h}_p \leftarrow \mathbf{h}_p\|\mathbf{h}_p\|_2^{-1}$; // Amplitude Constraint

    end

  end

end

then equation (3.54) can be written in the form

$$\begin{bmatrix} \mathbf{Y}_{(L)}^{(i)} & -\mathbf{Y}_{(L)}^{(j)} \end{bmatrix} \begin{bmatrix} \mathbf{h}^{(j)} \\ \mathbf{h}^{(i)} \end{bmatrix} = \mathbf{0}_{2L\times 1}. \tag{3.56}$$

Thus, the impulse response vector is the nullspace of the matrix in equation (3.56). Like any blind channel estimate, this is within a constant of proportionality, as discussed in section 2.1.

Furthermore, the relationship can be extended to simultaneously express the cross relation for all combinations of receiver pairs. In this general case, the matrix equation is formed as follows:

$$\begin{bmatrix} \mathbf{Y}_{(L)}^{(2)} & -\mathbf{Y}_{(L)}^{(1)} & \mathbf{0}_{L\times 1} & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} \\ \mathbf{Y}_{(L)}^{(3)} & \mathbf{0}_{L\times 1} & -\mathbf{Y}_{(L)}^{(1)} & \mathbf{0}_{L\times 1} & \cdots & \vdots \\ \vdots & \vdots & & \ddots & & \mathbf{0}_{L\times 1} \\ \mathbf{Y}_{(L)}^{(N)} & \mathbf{0}_{L\times 1} & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} & -\mathbf{Y}_{(L)}^{(1)} \\ \\ \mathbf{0}_{L\times 1} & \mathbf{Y}_{(L)}^{(3)} & -\mathbf{Y}_{(L)}^{(2)} & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} \\ \mathbf{0}_{L\times 1} & \mathbf{Y}_{(L)}^{(4)} & \mathbf{0}_{L\times 1} & -\mathbf{Y}_{(L)}^{(2)} & \ddots & \vdots \\ \vdots & \vdots & & \ddots & & \mathbf{0}_{L\times 1} \\ \mathbf{0}_{L\times 1} & \mathbf{Y}_{(L)}^{(N)} & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} & -\mathbf{Y}_{(L)}^{(2)} \\ \vdots & & & & & \vdots \\ \mathbf{0}_{L\times 1} & \cdots & & \mathbf{0}_{L\times 1} & \mathbf{Y}_{(L)}^{(N)} & -\mathbf{Y}_{(L)}^{(N-1)} \end{bmatrix} \begin{bmatrix} \mathbf{h}^{(1)} \\ \mathbf{h}^{(2)} \\ \vdots \\ \mathbf{h}^{(N)} \end{bmatrix} = \mathbf{0}_{\frac{(N^2-N)(T-L-1)}{2}\times 1} \tag{3.57}$$

or concisely

$$\dot{\mathbf{Y}}\dot{\mathbf{h}} = \mathbf{0}. \tag{3.58}$$

The impulse response vector is then the nullspace of this large $\frac{1}{2}(N^2 - N)(T - L - 1) \times NL$ matrix $\dot{\mathbf{Y}}$. The solution can be obtained by solving $\min_{\dot{\mathbf{h}}} \|\dot{\mathbf{Y}}\dot{\mathbf{h}}\|$ subject to some constraint on $\|\dot{\mathbf{h}}\|$ that prevents the trivial solution $\dot{\mathbf{h}} = \mathbf{0}$.

A convenient and computationally efficient solution to this equation, specifically

$$\min_{\dot{\mathbf{h}}} \|\dot{\mathbf{Y}}\dot{\mathbf{h}}\|_2 \quad \text{s.t. } \|\dot{\mathbf{h}}\|_2 = 1 \tag{3.59}$$

can be obtained by the singular value decomposition (SVD).

$$\dot{\mathbf{Y}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{3.60}$$

$$\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, ..., \sigma_{NL}), \qquad \text{where } \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{NL}. \tag{3.61}$$

Here $\sigma_1, \sigma_2, ..., \sigma_L$ are the singular values, and $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices. Then the last j columns of $\mathbf{V}$, corresponding to the singular values $\sigma_{NL-j+1} = \cdots = \sigma_{NL-1} = \sigma_{NL} = 0$ form an orthonormal basis for the nullspace of $\dot{\mathbf{Y}}$.

The method would ordinarily continue by using the estimated impulse response to form Bezout identity based inverse filters and subsequently recovering $\mathbf{s}$ as described in section 2.1.1.

Noise Free - No Common Zeros In an noise free environment, a unique solution is guaranteed provided that the channels meet the criterion of having no common zeros and the requirement on the complexity of the source is met (see section 2.1). Then the nullspace has a well defined dimension of one. This is evident upon calculating the SVD in that one singular value will be zero. In this scenario the method works well and $\dot{\mathbf{h}}$ is the last column of $\mathbf{V}$. This is pictured in figure 3.9, only one singular value is zero. The right graph in the figure provides a means of observing the contribution that the columns of $\mathbf{V}$ corresponding to the smallest j singular values make to $\dot{\mathbf{h}}$. The plot is of the projection residual:

$$\frac{\|\dot{\mathbf{h}} - \mathbf{P}\dot{\mathbf{h}}\|}{\|\dot{\mathbf{h}}\|} \tag{3.62}$$

where $\mathbf{P}$ is formed from the last j columns of $\mathbf{V}$, for varying values of j. Of interest is the smallest value of j for which the residual is negligible, i.e. identifying the j columns of $\mathbf{V}$ which form a subspace containing $\dot{\mathbf{h}}$. In this case, $j = 1$, i.e. the last column of $\mathbf{V}$ defines a subspace containing $\dot{\mathbf{h}}$. This will be referred to as the well conditioned case.

However, in the presence of additive noise, common zeros or if the system is otherwise degenerate, the nullspace becomes ill defined as described below.

Common Zeros or Insufficient Source Diversity Multiple singular values will be zero if the channels have common zeros or if the source fails to meet the source complexity requirement discussed in section 2.1.2. Then the

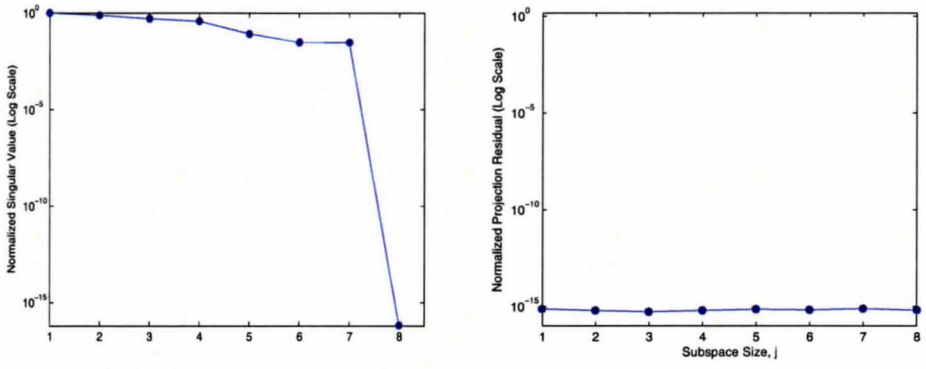

Figure 3.9: Singular values of $\dot{\mathbf{Y}}$ for a well conditioned, noise free case, with two channels, each of length 4 (total of 8 singular values) (left). Normalized projection residual: $\frac{\|\dot{\mathbf{h}}-\mathbf{P}\dot{\mathbf{h}}\|}{\|\dot{\mathbf{h}}\|}$, where $\mathbf{P}$ is formed from the last j columns of $\mathbf{V}$ (right).

impulse response vector will be in the larger nullspace, but it will not be possible to uniquely identify $\dot{\mathbf{h}}$. That is, if $\sigma_{NL-j+1} = \cdots = \sigma_{NL-1} = \sigma_{NL} = 0$, and $\mathbf{v}(i)$ is the i-th column of $\mathbf{V}$ then

$$\alpha_1 \mathbf{v}(NL-j+1) + \cdots + \alpha_{j-1}\mathbf{v}(NL-1) + \alpha_j \mathbf{v}(NL) = \dot{\mathbf{h}} \tag{3.63}$$

and the contribution coefficients $\alpha_1 \cdots \alpha_j$ can not be determined. A case with a single common zero is shown in 3.10. The singular values clearly show two near zero singular values, and the projection residuals show that the last two columns of $\mathbf{V}$ form a basis for a subspace containing the channel response.

Noise In the case where there noise is present, the smallest singular values will be proportional to the observation noise variance. In this case, there will be no zero singular values, but rather a multiplicity of small singular values with similar values related to the standard deviation of the noise.

This is scenario is pictured in figure 3.11. The singular values are shown with and without noise. The case where noise is added shows that the formerly zero singular value is difficult to distinguish from the neighboring three. Moreover, it is difficult to identify a set of column vectors from $\mathbf{V}$ that form a basis for a nullspace containing the impulse response. Thus, moderate amounts of

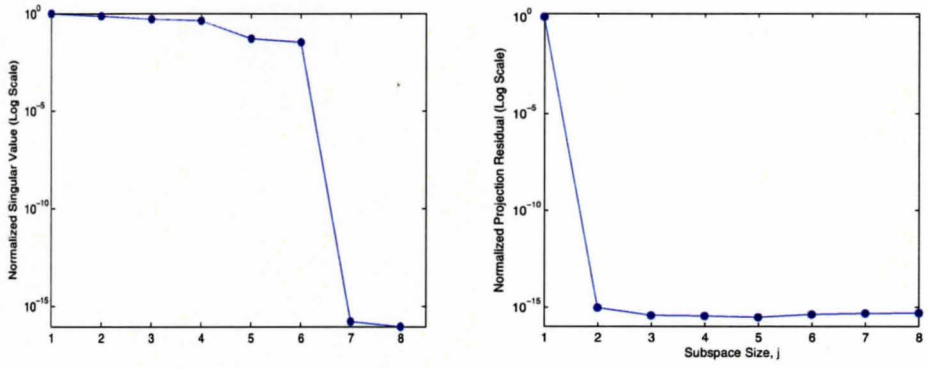

Figure 3.10: Singular values of $\dot{\mathbf{Y}}$ with common zeros, noise free case, with two channels, each of length 4 (total of 8 singular values) (left). Normalized projection residual: $\frac{\|\dot{\mathbf{h}}-\mathbf{P}\dot{\mathbf{h}}\|}{\|\dot{\mathbf{h}}\|}$, where $\mathbf{P}$ is formed from the last j columns of $\mathbf{V}$ (right).

noise will make the nullspace difficult to identify and the column of $\mathbf{V}$ corresponding to the smallest singular value will not be equal to the channel vector.

These two cases define two modes of degeneracy relative to the case where the nullspace has a dimension of 1 and is well defined. In the degenerate case the nullspace or approximate nullspace of $\dot{\mathbf{Y}}$ has more than one dimension and is defined by multiple basis vectors.

Identification of Approximate Nullspace

At this point, the method in [49] fails as there is no way to identify the true nullspace. However a larger approximate nullspace can still be identified, corresponding to singular vectors associated with a selected number of the smallest singular values. The span of these singular vectors defines a subspace $\mathcal{S}$ which will contain the correct solution for the impulse response vector. The dimension of $\mathcal{S}$ is typically substantially smaller than the NL, the dimension the original particle filter estimate. Thus, an estimate can be improved by projecting into $\mathcal{S}$, effectively making the estimate variance zero in directions orthogonal to the subspace. A discussion of this in [42] shows that a constraint projected estimate is guaranteed to produce an error less than or equal to an unconstrained estimate.

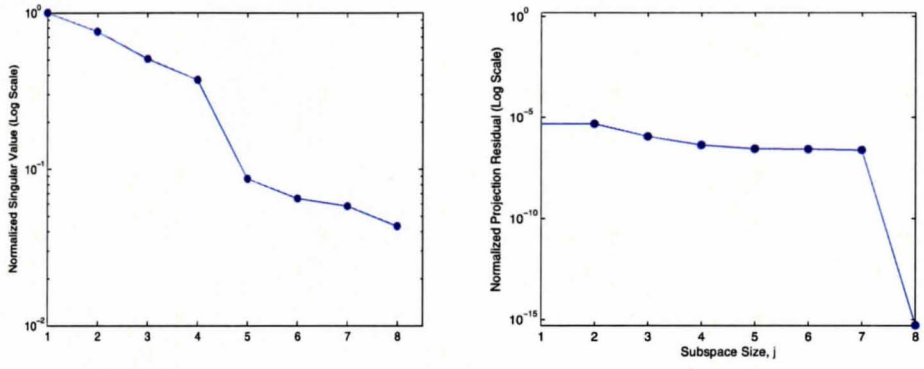

Figure 3.11: Singular values of $\dot{\mathbf{Y}}$ in the presence of noise, with two channels, each of length 4 (total of 8 singular values) (left). Normalized projection residual: $\frac{\|\dot{\mathbf{h}}-\mathbf{P}\dot{\mathbf{h}}\|}{\|\dot{\mathbf{h}}\|}$, where $\mathbf{P}$ is formed from the last j columns of $\mathbf{V}$ (right).

The subspace is chosen based on the smallest singular values obtained from the SVD. An appropriate size (number of dimensions) of this subspace must be chosen to ensure that $\mathbf{h}$ is in the subspace.

This can be done on the basis of the singular values of $\dot{\mathbf{Y}}$ from equation (3.60). In low noise cases, it suffices to inspect the singular values to identify the smallest singular values. For example, in the case shown in figure 3.10 there are clearly two near zero singular values, thus it is appropriate to choose the two columns of $\mathbf{V}$ corresponding to those two singular values as the basis for the subspace.

In cases with significant noise or ones where the smallest singular values are otherwise hard to distinguish, a conservative estimate can be used. In the extreme example shown in figure 3.11, it would be appropriate to include the entire space spanned by the columns of $\mathbf{V}$ since the magnitudes of the singular values are essentially indistinguishable.

Projection

With the subspace identified, the estimated solution can be projected onto the subspace. Orthogonal projection of a vector onto a subspace described by a number of vectors is accomplished by multiplying the vector by a projection matrix, formed as described below.

Consider a set of vectors $\mathcal{A} = \{\mathbf{a}_1, \cdots, \mathbf{a}_N\}$that form a basis for a vector

space. Form a matrix $\mathbf{A}$ whose columns are the basis vectors

$$\mathbf{A} = \left[\mathbf{a}_1, \cdots, \mathbf{a}_N\right]. \tag{3.64}$$

Then the orthogonal projection matrix for the subspace spanned by $\mathcal{A}$ is defined as

$$\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T. \tag{3.65}$$

In the case where the basis vectors are orthonormal, this reduces to

$$\mathbf{P} = \mathbf{A}\mathbf{A}^T. \tag{3.66}$$

Then

$$\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x} \tag{3.67}$$

is the orthogonal projection of $\mathbf{x}$ onto the subspace spanned by $\mathcal{A}$.

Application of Projection to Marginalized Particle Filter

Appying this to the estimator developed up to this point is relatively straight forward. The subspace is extracted from $\dot{\mathbf{Y}}$ before the particle filter and the resulting estimated nullspace is used to form a projection matrix.

In the marginalized particle filter, algorithm 5, the value of $\mathbf{h}_{t+1}$ given by the Kalman filter for each particle is projected onto the known subspace from the SVD. If the estimated subspace consists of the last j columns of $\mathbf{V}$, then form $\mathbf{V}_{NL-j+1:NL}$, as the matrix of orthonormal basis vectors for the subspace. The projection matrix is then

$$\mathbf{P} = \mathbf{V}_{NL-j+1:NL}\mathbf{V}^T_{NL-j+1:NL}, \tag{3.68}$$

and the projection step is

$$\mathbf{h}_{t+1|t} = \mathbf{P}\mathbf{h}_{t+1|t}. \tag{3.69}$$

This method of constraining a Kalman filter to improve estimates is explored in detail, along with more complex constraints, in [42] and [43].

3.4.5 Computational Complexity

The asymptotic computational complexity of the proposed estimator is dominated by the following factors.

The first is calculation of the weight update in the marginalized particle filter algorithm. This involves calculating the a probability from an N-dimensional multivariate Gaussian. Calculating a multivariate Gaussian probability involves Cholesky decomposition of the covariance matrix as the dominant factor in the computational complexity, an $O(N^3)$ operation. The second factor is the complexity of testing the roots of the AR process for stability. This is accomplished by factoring the process polynomial, which is in turn done through the calculating the eigenvalues of the companion matrix, an $O(K^3)$ operation.

Each of the two calculations discussed above must be completed for each particle, thus the complexity of the marginalized particle filter portion of the estimator, for each time step, can be written as:

$$O(N_{part}(N^3 + K^3)). \tag{3.70}$$

The number of particles in the preceding equation can be approximately expressed in terms of the other dimensions of the problem. Although the relationship is not exact due to a high level of dependence between the state variables, an upper bound on the number of particles needed grows exponentially with the total number of dimensions in the state variables that must be sampled. In this estimator the number of sampled dimensions is the size of $\mathbf{z}$: $\max(K, M) + K + 2N + 2$. Thus, an upper bound on the asymptotic computational complexity of the estimator for each time step is

$$O(\alpha^{\max(K,M)+K+2N+2}(N^3 + K^3)), \tag{3.71}$$

for some unknown constant α.

The singular value decomposition in the projection algorithm contributes an additive $O(M^3)$ term to the complexity, but it is only calculated once at the beginning of the algorithm. Accordingly, this term is not significant in the asymptotic complexity of the overall estimator.

In this chapter a marginalized particle filter algorithm was developed for the blind deconvolution problem. It incorporated a classical method of blind deconvolution with the intent of making a more robust estimator.

In the next chapter, details of the initialization of the algorithm will be discussed along with a discussion of experimental results using the estimator.

Chapter 4

Application Details and Experimental Results

This chapter describes the application of the developed algorithm. First, the setup of the algorithm's parameters will be discussed. This will be followed by an analysis of the performance of the algorithm, including tests under the types of degeneracy inherent in acoustic scenarios as discussed in section 1.1.2.

4.1 Initialization and Parameter Settings

The algorithm has a number of control parameters that must be configured for good operation. Additionally, the initial states of

Parameters

- c_a, c_h - The coefficients affecting whether the previous estimate of the AR or MA coefficients (respectively) is scaled. In [13], 0.9999 is suggested. However, there is nothing to indicate that anything other than 1 should be used. The expectation is that, once converged, the random walk process should be on average stationary. Using a value other than 1 would tend to push the estimate in a particular direction and potentially bias the estimate.
- σ_a - The variance of the step size for the AR coefficients guides how quickly the estimate converges. It is also inversely proportional to the final accuracy of the estimator.

The first factor determining this step is the rate of change of the AR process being estimated. This can be estimated from observing similar cases and is assumed known. In the case of a slowly varying or stationary AR process a value of 0.01 was used. Empirically, it was determined that a value approximately equal to 1% of the typical AR coefficient value provided good results. In cases of typical speech explored, AR coefficients typically ranged from -1 to 1. Sensitivity to this parameter was mild and any value within 0.05 and 0.005 provided comparable results.

- $\mathbf{\Sigma}_h$ - The covariance matrix for the step size of the channel MA coefficients has similar properties to those discussed for σ_a above. Although it is a full covariance matrix, it is difficult to make assumptions or determine experimentally whether there is any consistent correlation between changes in coefficients at any given time step. Thus $\mathbf{\Sigma}_h$ was assigned to be a diagonal matrix. The diagonal elements are chosen to be proportional to the expected shape of the impulse response: constant in the synthetic trials presented here and exponentially decaying when used with acoustic impulse responses.

- $\sigma_{\phi v}$ - The covaraince of the step size of the log of the driving noise. This is configured experimentally to match the typical amplitude changes in the source, and the desired adaptation rate for the driving noise estimate. In the cases tested, this was set to 0.1.

- $\sigma_{\phi wi}$ - Similar to the previous parameter, the covariances of the observation noise was set experimentally based on the expected rate of change and the desired adaptation rate. For all channels, this was set to 0.2.

Any of the random walk step size variances influence the steady state accuracy of the estimator and the ability to track non-stationary elements of the model. Small step size variances converge to a smaller steady state error, but are unable to adapt to fast changes in the associated state variables. Conversely, large step size variances can track fast changes, result in a high level of steady state variance. This tradeoff must be considered when choosing these parameters.

Initial Values

As discussed in section 2.4.2, the particle states must be initialized from an appropriate initial distribution $p(x_0)$. This distribution reflects any prior information and any physical constraints. The initial distribution for the elements of the state is described below. Most elements are initialized with a non-informative prior [13]. This is a broad distribution that only reflects the physical constraints, but is otherwise representative of the fact that there is no prior knowledge available.

- **a** - For the AR coefficients, a non-informative prior was used representing the approximate distribution of coefficients in speech samples tested. A Gaussian distribution with mean 0 and variance 1 was used for all coefficients.

- ϕ_v - The initial distribution of the log of the driving noise covariance is approximately proportional to the magnitude of the source. Thus, this can be known within an order of magnitude form observing the output. A broad Gaussian prior with variance 5 is used, with a mean of 10. This is approximately the center of the results for a wide range of tests conducted.

- **s** - For the source, a number of options were tested: a non-informative prior, using the initial observations, and simply setting the initial values to a vector of zeros. It was found that there was no measurable difference between the options. Initializing the source to zeros is used in the examples below.

- **h** - The initial channel estimate was a non-informative prior. Each coefficient was drawn from a Gaussian distribution with variance 0.2 and mean of zero. This represented the approximate typical scale of the channel coefficients in the examples.

- ϕ_v - In a real environment, the observation noise can be estimated from during pauses in the speech. It is assumed that an estimate is available, thus a Gaussian distribution centered at the true value, but with a variance of 2 is used.

4.2 Post Processing

The amplitude ambiguity, although constrained by the algorithm, will still be present relative to the original source signal. This can be resolved by normalizing the mean of the absolute value of the estimate to the mean of the absolute value of the source. This mean is calculated after the estimator has settled to a steady state, in the examples below, over the last 100 samples of the estimate. This is the method used in [13] to normalize for the amplitude ambiguity.

All source estimation error figures presented in the following sections are mean square error in the case of ensemble calculations, and absolute error in the case of single runs. To facilitate comparison between different source amplitudes, the error is normalized relative to the RMS amplitude of the source signal. For the absolute error, this is calculated as

$$err_t = \frac{|\hat{s}_t - s_t|}{\sqrt{\frac{1}{T}\sum_{t=1}^{T} s_t^2}}. \tag{4.1}$$

For error data presented from single runs, a 10 sample window running average is shown to improve readability by showing the performance without spurious spikes in the results.

4.3 Tests

In this section, the results of tests conducted on the estimator are described. First the basic performance of the estimator will be established, then a number of difficult cases will be shown to demonstrate the robustness of the algorithm.

4.3.1 Basic Test

Baseline Test

The first test is designed to test the recovery of only the AR process from the channel outputs, effectively to test whether the algorithm can perform multichannel deconvolution. The purpose is to validate that the algorithm works before subjecting it to more strenuous tests.

A synthetic AR process of length 4 is mixed with two channels of length 4, with no common zeros. The position of the poles and zeros is depicted in figure 4.12. Except moderate observation noise, the system is free from any kind of potential degeneracy. From the singular values of the matrix $\dot{\mathbf{Y}}$, shown in figure 4.13, it is evident that one singular value is substantially smaller than all others, implying a clearly defined one dimensional nullspace. Thus the channel response can be estimated well with the subspace method described in section 3.4.4, and the size of the projection space is one dimension. This leaves everything in the estimator fully constrained, except the AR coefficients and the power of the driving noise for the AR process and observation noise. This test will serve as the benchmark for comparison to see how degenerate situations affect the quality of the estimate.

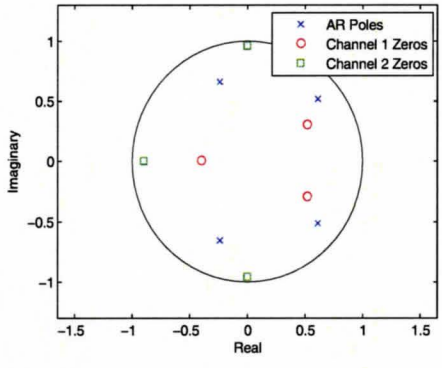

Figure 4.12: Pole and zero locations of the AR process and channels for the base case experiment.

Results for 15dB average signal to noise ratio, using 1000 particles are shown in figure 4.14. The results presented are from 100 trials of the estimator, of which 90 are presented, and the remaining 10 failed to track the source, for reasons described in the next paragraph. The mean square error of the source estimate converges to -42dB. It should be noted each trial had a substantially different learning curve and the composite curve only represents the average. Some runs converged almost instantly, while others took up to approximately 600 samples to reach steady-state. This is the cause of the shape of the mean learning curve displayed.

After the estimator converged, the AR process is well estimated, the estimated roots of the AR process coincide well with the true roots. This is shown

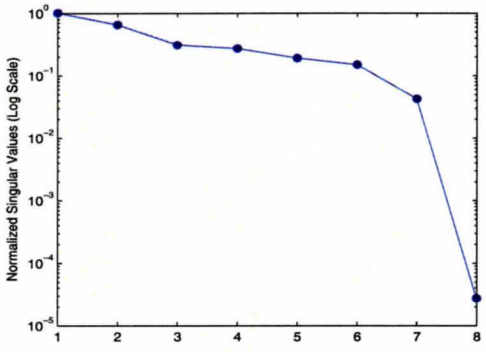

Figure 4.13: Normalized singular values of $\dot{\mathbf{Y}}$ for the basic test case.

in figure 4.15.

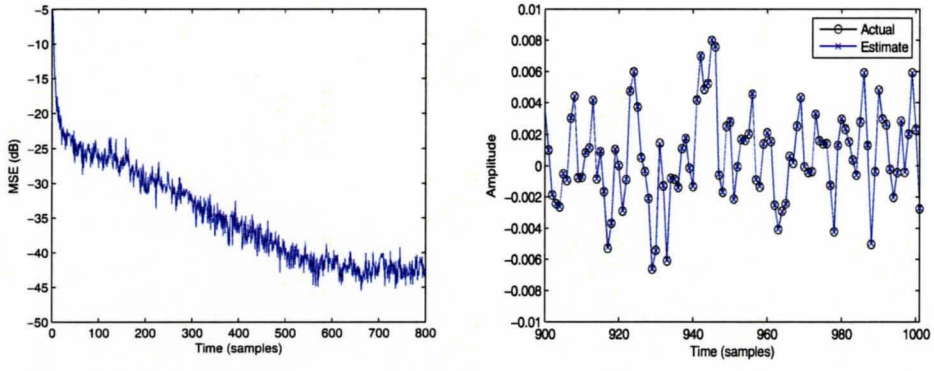

Figure 4.14: A typical case at 15dB SNR observation noise. Adaptation curve converging to -42dB MSE (left) and closeup of a single run showing the original signal and estimate (right).

Two issues became evident during the testing. First, the estimator would become numerically unstable with greater probability as the system observation noise level became lower. This would manifest itself as all particle weights approaching zero, leading to normalization of the particle weights failing.

The cause of the failure is the narrow likelihood of any state relative to the spread of the particle density. As the observation noise decreases, it takes more particles to cover the given range of the state. The outcome is similar to the effect of using a particle filter without resampling, only in this case

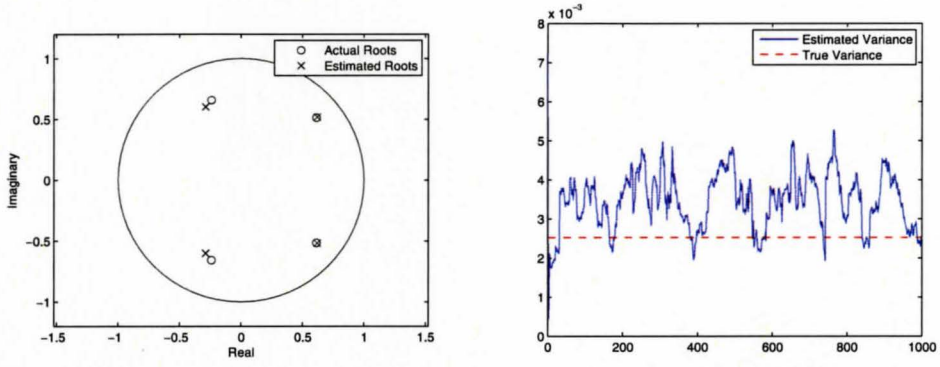

Figure 4.15: Actual and estimated roots of the source (left), and actual and estimated AR process driving noise variance (right).

resampling was not sufficient to prevent degeneracy. The only solution was to increase the number of particles. Using 1000 particles, the estimator was only usable to an observation signal to noise ratio of approximately 30dB.

Secondly, a number of the runs of the estimator did not track all of the zeros of the AR model. The mode of failure is almost exclusively manifested as two of the four poles being identified accurately, but the remaining two would be manifested as two poles along the real axis, rather than their true location. This would tend to suggest that the convergence is not towards a unique global minimum, but rather that local minima exist and can cause problems for the estimator.

At 15dB SNR, with 1000 particles, the two modes of failure occurred in 20% of runs.

Performance Comparison

In figure 4.16 the projection constrained method is compared to a marginalized particle filter without projection constraints and to the cross relation method on which the projection is based. The figure presents a single run of each estimator under the same conditions as the previous experiment. For clarity, the data is presented after passing through a 10 sample window running average.

The proposed estimator is a significant improvement over the -22dB error which results from using a marginalized particle filter alone, without nullspace projection. However, the estimator did not improve upon the cross relation

method as expected, which produced an error level of only -73dB.

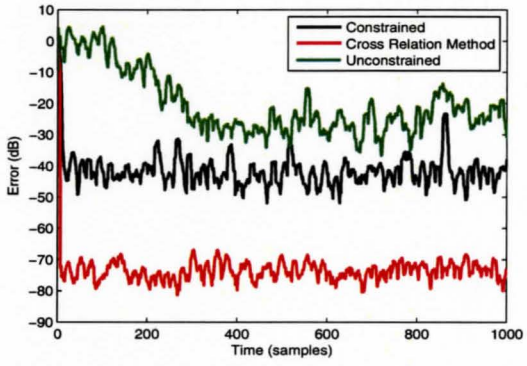

Figure 4.16: Error performance comparison of different blind deconvolution methods.

Error performance in absolute terms was not explored. A Cramer-Rao lower bound for the variance of a general nonlinear estimator for the filtering problem is discussed in [13]. The proposed estimation method utilizes a filtering estimator: the marginalized particle filter, and a non-filtering estimator component: the cross relation subspace algorithm, which has a substantially different lower bound. Accordingly, it is difficult to identify if the estimator is reaching a theoretical bound for the steady state variance.

Algorithm Speed

The algorithm is written efficiently, however, it is inherently slow, due in part to the exponential growth computational complexity described in the previous chapter. In the conditions of the previous test, approximately 10 samples per second were processed. This makes the algorithm infeasible for real time implementation, however, it may serve some value as an offline post processing step.

4.3.2 Tests Under Degenerate Conditions

The estimator is designed to provide good blind deconvolution performance in degenerate conditions. This section will explore how the algorithm performs in cases that exhibit the conditions of acoustic blind dereverberation described in section 1.1.2.

Non Minimum Phase

The algorithm does not explicitly invert the FIR channels, as such, it does not have any issues with non minimum phase channels. Direct estimation of the source is an alternative to Bezout identity based inversion of non minimum phase channels. A sample case illustrating the scenario is depicted in figure 4.17, note that the error graph in the figure uses a length 10 window running average filter for clarity. The test conditions are the same as those in section 4.3.1, but a number of poles of the channels have been moved outside the unit circle. The learning curve shows good performance, converging to -35dB MSE, similar to the minimum phase case, demonstrating that non minimum phase channels have no significant effect on the estimate.

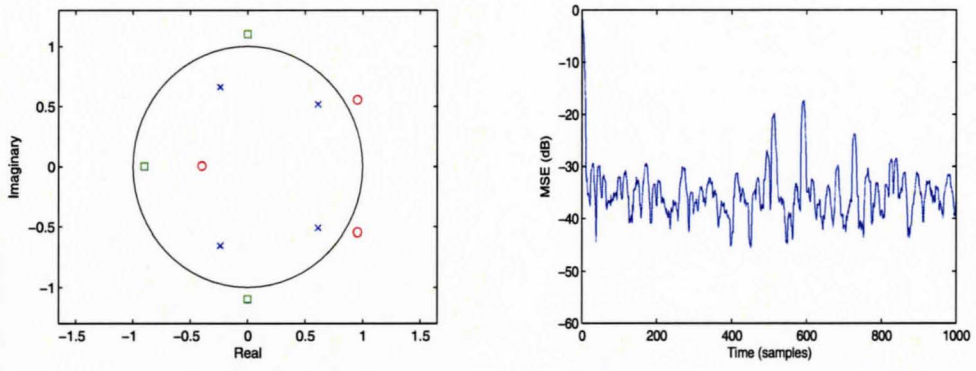

Figure 4.17: Non minimum phase channels. The poles of the channels (left) and the learning curve for a typical run of estimator converging to -35dB MSE (right).

Incorrectly Estimated Length

As discussed in section 1.1.2, one of the requirements of working with acoustic impulse responses is that the estimator must degrade gracefully with an incorrectly estimated channel length. In this experiment, the channel length is overestimated. The true channels have a length of 4, but the length used by the estimator is 6. The cross relation based estimator used for obtaining the projection space fails in this case. However, one can observe that the projection space provided by this algorithm can include the true channel response (zero padded to appropriate length) by extending the dimension of the

subspace. The residual of the projection of the zero extended true channel response vector onto the approximate nullspace of increasing size as per equation (3.62) is shown in figure 4.18. It shows that if the projection space is set to 3 dimensions, the error in the projection is limited to the noise floor. Thus the projection method can be used in cases where the channel length may be overestimated. In this case, the singular values of $\dot{\mathbf{Y}}$ clearly show that the length is overestimated, however if the situation were not as clear, a conservative estimate could be used which would yield similar results.

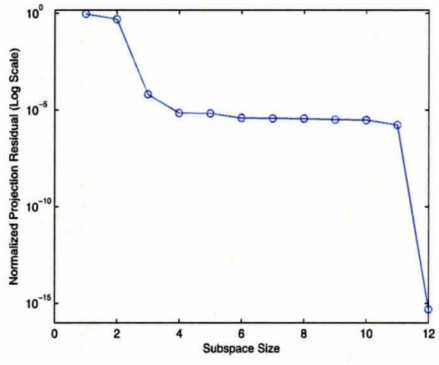

Figure 4.18: Overestimated channel length: projection residual for varying number of dimensions of the projection space.

Figure 4.19 shows that with the channel length overestimated, the proposed algorithm provides substantially better performance than the cross relation approach. The error figure uses a length 10 window running average filter for clarity. The projection constrained marginalized particle filter estimator converges to -12dB MSE, which tracks the source reasonably well. In contrast, the cross relation estimate is largely meaningless. Thus it can be said that the proposed algorithm is robust to incorrectly estimated channel length because it degrades gracefully in such a case.

Near Common Zeros

As discussed in section 2.1.2, a necessary criterion for multichannel blind deconvolution is that the channels have no common zeros. Not meeting this criterion causes deterministic blind deconvolution methods to fail. However, near common zeros are one of the challenges presented by acoustic environments.

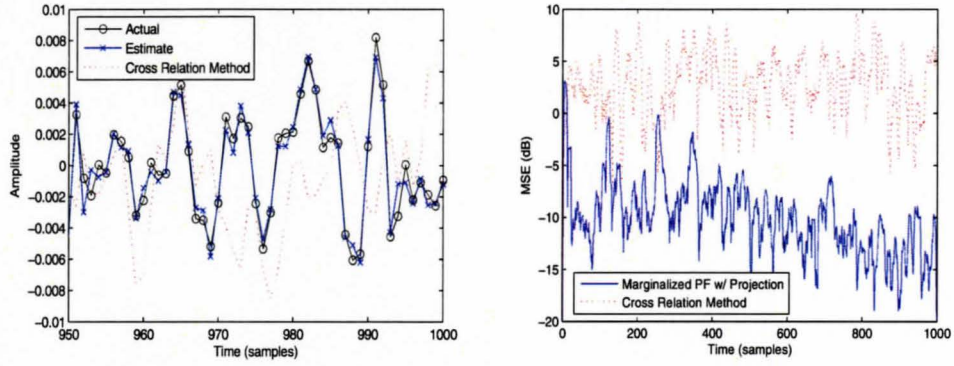

Figure 4.19: Overestimated channel length test results.

In this section the effect of near common zeros on the proposed estimator is explored.

The test conditions are the same as those in the previous section. The zeros of the channels will be moved to test how the error performance changes as zeros are brought together. Two of the zeros of one of the channels are rotated towards zeros of the other channel as pictured in figure 4.20.

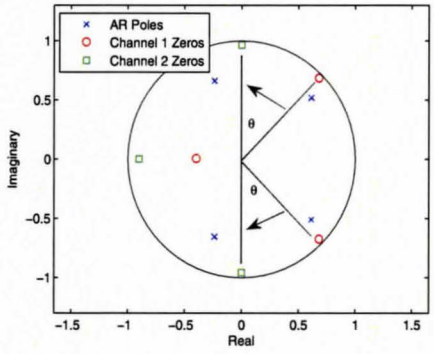

Figure 4.20: Approaching common zeros.

The results of the error once converged, versus the angle between the poles, θ, is shown in figure 4.21. The cross relation method decays logarithmically as the angle between the zeros approaches zero while the proposed method decays much more gradually. The log plot shows the results as θ approached zero. As the angle gets small, the proposed method shows better error performance than

the cross relation method. Additionally, extending the size of the projection space improves the estimate slightly.

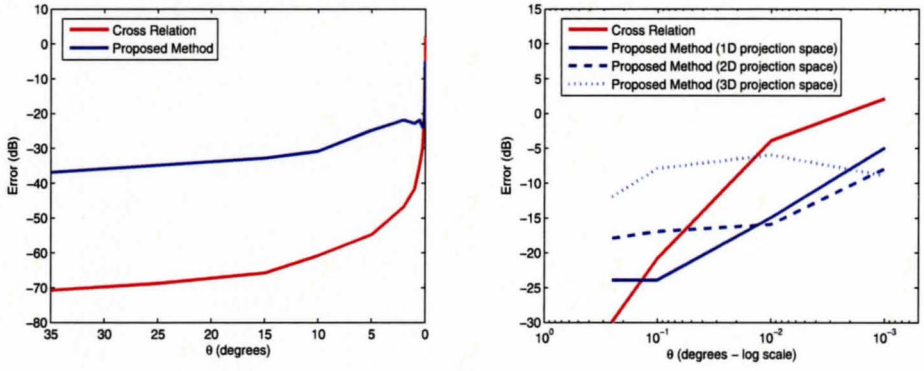

Figure 4.21: Near common zeros results. Linear scale, compared to cross relation method (left). Log scale for small values of θ, proposed method with 1D, 2D and 3D projection subspace, compared to cross relation method (right).

Thus the proposed can handle near common zeros. There is one exception, if the common zero were to be near or overlapping a pole of the AR process, it would be impossible to identify the effects of that pole and the estimate would be relatively poor.

Time Varying Parameters

To test if the estimator can track the time varying parameters of speech, the same channels were used as in the base case example, but a sample of speech [19] was used as the source. The length of the AR process was increased to 8 to accommodate the speech. Results for 50 runs, shown in figure 4.22, shows the MSE converging to an average of -41dB, comparable to the performance of the base.

To evaluate the effects of changing the channels, the same speech was used, but the zeros for both channels were changed mid way through the test. The size of the projection subspace to use was determined by looking at the singular values of $\dot{\mathbf{Y}}$. These are shown in figure 4.23. Based on this, graph, two singular values are substantially smaller than the others, implying that the corresponding two dimensional projection subspace be used.

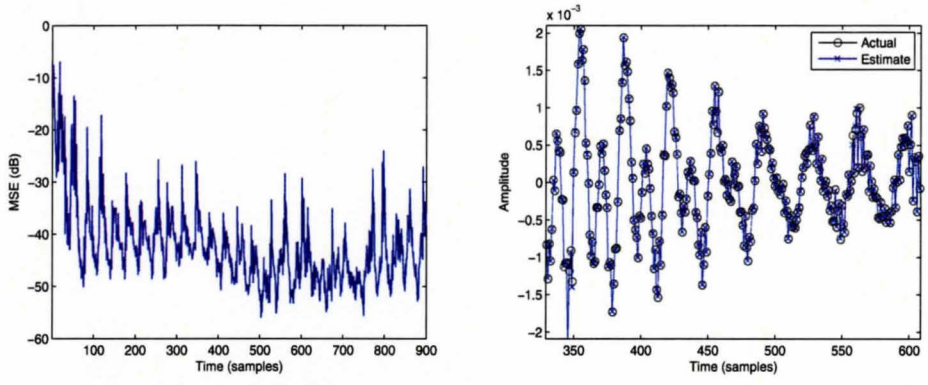

Figure 4.22: Speech estimate at 15dB SNR observation noise. Adaptation curve converging to -41dB MSE (left) and closeup of a single run showing the original signal and estimate (right).

The results are shown in figure 4.24. They have been filtered with a length 10 window running average for clarity. The figure shows that the proposed method performs significantly better than the cross relation method. The marginalized particle filter with projection constraints produced an mean error of-24dB, compared to -13dB for the cross relation method.

Overall the proposed estimator degrades gracefully in the presence of the challenges posed by an acoustic environment. It is an improvement on the conventional marginalized particle filter in terms of error performance and an improvement on the cross relation method in terms of robustness.

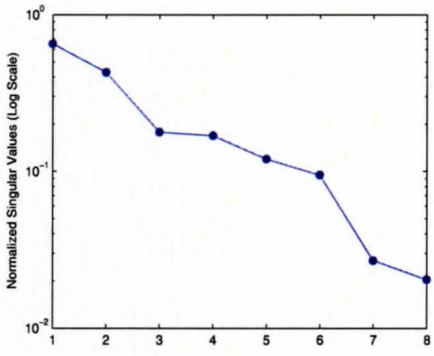

Figure 4.23: Singular values of $\dot{\mathbf{Y}}$ for a small change in channel coefficients.

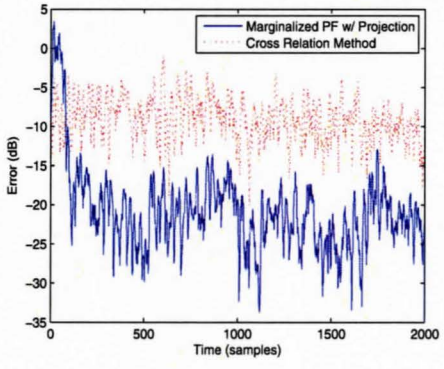

Figure 4.24: Adaptation curve for a estimating a speech source in changing channels.

Chapter 5

Summary

5.1 Contribution

In this thesis a multichannel deconvolution method was developed. The blind deconvolution problem was formulated as a joint state and parameter estimation problem and subsequently transformed to a recursive state estimation problem. A solution was then formulated on the basis of a marginalized particle filter. The method included integration of a conventional blind deconvolution method to produce a hybrid. The cross relation method was run on the data initially, and the results were used to generate a subspace onto which the particle filter solution was projected. The result was a method that outperforms a purely particle filter approach and in some cases the classical cross relation approach of [49].

5.2 Conclusions

The proposed method performed well on a number of test cases representative of the acoustic environments. The following list outlines how the estimator handles the challenges discussed in section 1.1.2.

1. **Long length** - This was the only challenge that was not explored. Subband approaches may be possible, but may be limited by the slow speed of the estimator.

2. **Non-minimum phase** - The estimator did not have any issues with

non-minimum phase channels since it does not rely on directly inverting the channels.

3. **Ill defined length** - In the experiments conducted, the estimator decayed gracefully in the case of overestimated channel length.

4. **Noise** - High noise levels proved to not be a significant issue for the estimator. The algorithm worked well in the tested cases at moderate SNR.

5. **Possibility of common zeros or near common zeros** - In tests involving common zeros the algorithm provided a useful estimate and an improvement over the classical cross relation approach.

6. **Possibly non-stationary** - The algorithm was able to track changing source and channel parameters.

Despite this good performance, the proposed method did not outperform the cross relation method in non-degenerate cases. After the estimator had adapted, the steady state error was still considerably higher than that provided by the reference cross relation method. Additionally, the computational complexity makes the method impractical for audio in the foreseeable future.

The marginalized particle filter can however be viewed as a good solution to a general nonlinear estimation problem. In areas such as industrial control and finance, these estimation methods may be usable despite their slow speed.

5.3 Future Research

As a result of researching blind deconvolution methods, a number of areas for potential exploration were identified.

Separating the Decaying Tail

One of the original intents of this research was to explore modeling the effect of the decaying tail of the acoustic impulse response separately from modeling it as conventional convolution. The following model was proposed for the observation as an alternative to equation (3.17):

$$\mathbf{y}_t = \mathbf{H}\mathbf{s}_t + \mathbf{\Theta}_t + \mathbf{w}_t. \tag{5.1}$$

In this model the impulse responses in $\mathbf{H}$ is expected to be formed from a truncated set of impulse responses. Then $\mathbf{\Theta}_t$ represents the contribution of the decaying tail to the observation.

It is hypothesized that $\mathbf{\Theta}_t$ can be modeled statistically, by estimating the distribution

$$p(\mathbf{\Theta}_t|\mathbf{\Theta}_{t-1}, \cdots, \mathbf{\Theta}_{t-k}, \mathbf{s}_t, \cdots, \mathbf{s}_{t-k}) \tag{5.2}$$

from a training set of impulse responses and sources. Then $\mathbf{\Theta}_t$ can be estimated along with $\mathbf{s}_t$ in a blind deconvolution estimator. Thus the effects of the impulse response tail can be captured separate of the deconvolution element. This may resolve some of the blind deconvolution difficulties imposed by the long decaying tail of acoustic impulse responses.

This approach was integrated with the marginalized particle filter developed in this thesis. However, the results did not prove fruitful in this framework. This is in part due to the extra dimensionality imposed by $\mathbf{\Theta}$, which has a negative effect on the performance of the particle filter. Futhermore, the distribution of $\mathbf{\Theta}_t$ that was estimated from training data was a heavy tailed multivariate Laplace distribution, which is difficult to sample meaningfully since the samples are very widely distributed. This difficulty was caused by the large amplitude changes in speech. When amplitude compressed speech was used (normalized by a low pass envelope), the distribution could be reasonably modeled by multivariate Gaussian distribution.

The method may still have merit outside of particle filtering. However, there was no opportunity to study this further.

Modeling the Estimation Error

A variation on modeling the decaying tail as discussed above is to explicitly model the error in the solution from an arbitrary blind deconvolution algorithm. The suggestion is that the estimation error of s_t consists of an uncorrelated portion and a correlated portion:

$$\hat{s}_t = s_t + \theta_t + e_t, \tag{5.3}$$

where θ_t is correlated to $\hat{s}_t$ and past instances of θ, and e_t is uncorrelated. If the distribution of θ_t can be modeled as $p(\theta_t|\theta_{t-1}, \cdots, \theta_{t-k}, \hat{s}_t, \cdots, \hat{s}_{t-k})$ on the basis of a training set, then θ_t can be estimated from $\hat{s}_t$. This can then be used to improve the quality of the estimate by subtracting the estimated θ directly

or by shaping a whitening filter to the estimate, to eliminate the correlation. Uncorrelated noise sounds better to a listener than correlated noise, this is one of the significant problems with acoustic reverberation.

This method was not tried within the scope of this research, however it may have some merit as a post processing step for a wide range of blind deconvolution algorithms.

Reconsideration of the L^2 Norm

Throughout blind deconvolution literature, there are numerous places where over constrained systems of linear equations are solved

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|, \tag{5.4}$$

often with a simple constraint on the solution to resolve a one dimensional ambiguity

$$\text{s.t. } \|\mathbf{x}\| = 1. \tag{5.5}$$

This can be seen in this thesis in solving for the nullspace, equation 3.59; and solving for the inverse filters in the Bezout identity, equation (A.7).

Almost universally, this formulated in terms of minimizing the L^2 norm, for reasons of simple analytical tractability. The unconstrained solution can be solved by Moore-Penrose pseudoinverse and the constrained solution by SVD (as in equation (3.60)). This may provide fast analytical solutions, but whether this provides good perceived sound quality in acoustic blind dereverberation is not an explored area. In [49], the general solution is noted, but then the final algorithm is formulated for the L^2 norm.

Using the L^1 and L^∞ norms may provide improved sound quality and solutions can still be arrived at reasonably quickly through convex optimization [9]. Moreover, analytical solutions exist to certain problems, most notably $\min \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ s.t. $\|\mathbf{x}\|_1 = 1$. This was explored in the process of researching this thesis, when working with the method of [49] it was noted to produce better sounding results in simulated acoustic scenarios.

Ultimately in acoustic scenarios the choice of the norm or other aspects of the algorithm must be explored in not just analytical terms, but also in terms of the perceived sound quality.

Appendix A

Bezout Identiy Inverse Filters

The Bezout identity allows the perfect inversion of a system composed of two or more FIR channels through the use of an equal number of FIR filters. Given channel responses, expressed as polynomials

$$h^{(1)}(z), h^{(2)}(z), \cdots, h^{(N)}(z) \tag{A.1}$$

of length $L^{(1)}, L^{(2)}, \cdots, L^{(N)}$ respectively. Then there exists a set of filter polynomials

$$p^{(1)}(z), p^{(2)}(z), \cdots, p^{(N)}(z) \tag{A.2}$$

of length $L^{(1)} - 1, L^{(2)} - 1, \cdots, L^{(N)} - 1$ respectively such that

$$h^{(1)}(z)p^{(1)}(z) + h^{(2)}(z)p^{(2)}(z) + \cdots + h^{(N)}(z)p^{(N)}(z) = 1. \tag{A.3}$$

This is true if and only if $h^{(1)}(z), h^{(2)}(z), \cdots, h^{(N)}(z)$ are relatively coprime, they have no common zeros [27]. The lengths of $p^{(1)}(z), p^{(2)}(z), \cdots, p^{(N)}(z)$ can be longer, but then the solution is no longer unique.

The inverting set $p^{(1)}(z), p^{(2)}(z), \cdots, p^{(N)}(z)$ can be found by expressing A.3 as convolution in vector matrix form. For each channel i, form a $2L-1 \times L-1$

convolution matrix from the coefficients of $h^{(i)}(z)$:

$$\mathbf{H}^{(i)} = \begin{bmatrix} h_1^{(i)} & 0 & 0 & \cdots & 0 \\ h_2^{(i)} & h_1^{(i)} & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 0 \\ h_{L^{(i)}-1}^{(i)} & \cdots & \cdots & h_2^{(i)} & h_1^{(i)} \\ h_{L^{(i)}}^{(i)} & \cdots & \cdots & h_3^{(i)} & h_2^{(i)} \\ 0 & h_{L^{(i)}}^{(i)} & & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & h_{L^{(i)}}^{(i)} & h_{L^{(i)}-1}^{(i)} \\ 0 & \cdots & 0 & 0 & h_{L^{(i)}}^{(i)} \end{bmatrix}. \tag{A.4}$$

Then equation (A.3) can be written

$$\begin{bmatrix} \mathbf{H}^{(1)} & \mathbf{H}^{(2)} & \cdots & \mathbf{H}^{(N)} \end{bmatrix} \begin{bmatrix} p_1^{(1)} \\ \vdots \\ p_{L^{(1)}}^{(1)} \\ \vdots \\ p_1^{(N)} \\ \vdots \\ p_{L^{(N)}}^{(N)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{A.5}$$

or concisely as

$$\mathbf{Hp} = \begin{bmatrix} 1 \\ \mathbf{0}_{2L-2\times 1} \end{bmatrix}. \tag{A.6}$$

The inverting filters can then be identified by solving the linear equation.

Note In practice, the $h^{(i)}(z)$ has been estimated and is not known exactly. In this case, it has been suggested that acoustically more pleasing results can be obtained by moving the 1 in the right hand vector of equation (A.5) further down the vector and otherwise filling with zeros. Effectively, any of the elementary vectors $\mathbf{e}$. There is no known research about this, but enough

anecdotal evidence has been shown that it warrants mentioning.

If $h^{(i)}(z)$ has been estimated multiple times, equation (A.6) can be extended to an over constrained system by including several estimates of $h^{(i)}(z)$ in $\mathbf{H}$ sequentially. The solution can then come about through solution of the minimization problem

$$\min_{\mathbf{p}} \|\mathbf{Hp} - \mathbf{e}\|. \tag{A.7}$$

SIMO Inversion With the filters $p^{(1)}(z), p^{(2)}(z), \cdots, p^{(N)}(z)$ identified, the SIMO system can be inverted using the inverting set of filters using the linear system depicted in figure A.25.

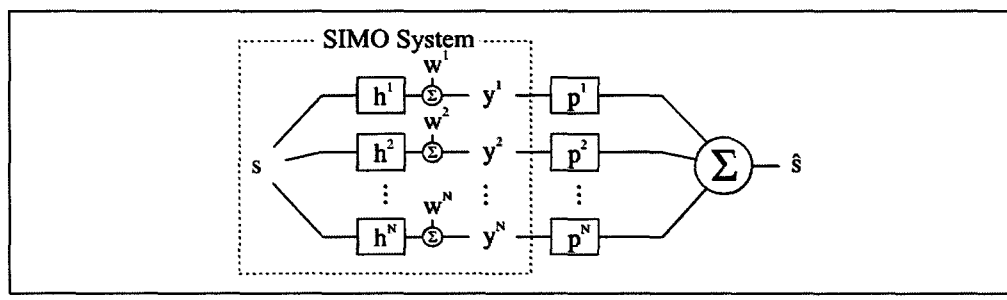

Figure A.25: Bezout identity based inversion of a SIMO system.

Bibliography

[1] K. Abed-Meraim, W. Qiu, and Y. Hua, "Blind system identification," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1310–1322, Aug 1997.

[2] S. Amari, S. Douglas, A. Cichocki, and H. Yang, "Multichannel blind deconvolution and equalization using the natural gradient," *Signal Processing Advances in Wireless Communications, 1997 First IEEE Signal Processing Workshop on*, pp. 101–104, Apr 1997.

[3] C. Andrieu and A. Doucet, "Online expectation-maximization type algorithms for parameter estimation in general state space models," *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP '03). Proceedings.*, vol. 6, pp. VI–69–72 vol.6, April 2003.

[4] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[5] T. Bakir, "Blind adaptive dereverberation of speech signals using a microphone array," Ph.D. dissertation, Georgia Institute of Technology, 2004.

[6] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation.* New York, NY: John Wiley & Sons, Inc., 2002.

[7] C. Berzuini, N. Best, W. Gilks, and C. Larizza, "Dynamic conditional independence models and markov chain Monte Carlo methods," *Journal of the American Statistical Association*, vol. 92, no. 440, pp. 1403–1412, 1997.

[8] M. Bolic, P. Djuric, and S. Hong, "Resampling algorithms and architectures for distributed particle filters," *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2442–2450, July 2005.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, NYA: Cambridge University Press, 2004.

[10] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," Adaptive Systems Lab, McMaster University, Tech. Rep., 2003.

[11] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications.* New York, NY: John Wiley & Sons, Inc., 2002.

[12] D. Crisan, "Particle filters - a theoretical perspective," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York, NY: Springer–Verlag, 2001.

[13] M. Daly, "Marginalized particle filters for blind system identification," Master's thesis, McMaster University, 2004.

[14] J. Deller, J. Proakis, and J. Hansen, *Discrete Time Processing of Speech Signals.* Upper Saddle River, NJ: Prentice Hall PTR, 1993.

[15] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York, NY: Springer–Verlag, 2001.

[16] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence.* San Francisco, CA: Morgan Kaufmann Publishers Inc., 2000, pp. 176–183.

[17] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208(12), July 2000.

[18] A. Doucet and V. Tadic, "Parameter estimation in general state-space models using particle methods," *Annals of the Institute of Statistical Mathematics*, vol. 55, no. 2, pp. 409–422, June 2003.

[19] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus cdrom," 1993.

[20] S. Godsill, A. Doucet, and M. West, "Maximum a posteriori sequence estimation using Monte Carlo particle filters," *Annals of the Institute of Statistical Mathematics*, vol. 53, no. 1, pp. 82–96, March 2001.

[21] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music.* New York, NY: John Wiley & Sons, Inc., 1999.

[22] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach.* Hoboken, New Jersey: John Wiley & Sons, Inc., 2004.

[23] A. Haug, "A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian processes," MITRE Corporation, McLean, VA, Tech. Rep., January 2005.

[24] M. Hodgson and E. Nosal, "Experimental evaluation of radiosity for room sound-field prediction," *The Journal of the Acoustical Society of America*, vol. 120, no. 2, pp. 808–819, 2006.

[25] J. Hol, T. Schon, and F. Gustafsson, "On resampling algorithms for particle filters," *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pp. 79–82, Sept. 2006.

[26] S. Julier and J. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference on Signal Processing, Sensor Fusion, and Target Recognition VI. Proccedings.*, I. Kadar, Ed., vol. 3068, July 1997, pp. 182–193.

[27] T. Kailath, *Linear Systems.* Englewood Cliffs, NJ: Prentice–Hall, 1980.

[28] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 3545, March 1960.

[29] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[30] ——, "A self-organizing state-space model," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1203–1215, 1998.

[31] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *Signal Processing Magazine, IEEE*, vol. 13, no. 3, pp. 43–64, May 1996.

[32] H. Liu, G. Xu, L. Tong, and T. Kailath, "Recent developments in blind channel equalization: From cyclostationarity to subspaces," *Signal Processing*, vol. 50, no. 1-2, pp. 83–99, April 1996.

[33] H. Luo and Y. Li, "The application of blind channel identification techniques to prestack seismic deconvolution," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 2082–2089, Oct 1998.

[34] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, no. 2, pp. 145–152, Feb 1988.

[35] E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," *Signal Processing, IEEE Transactions on*, vol. 43, no. 2, pp. 516–525, Feb 1995.

[36] C. Musso, N. Oudjane, and F. L. Gland, "Improving regularized particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York, NY: Springer–Verlag, 2001.

[37] H. Pudder, "Noise reduction with Kalman-filters for hands-free car phones based on parametric spectral speech and noise estimates," in *Topics in Acoustic Echo and Noise Control.* Berlin: Springer, 2006, pp. 385–427, online: http://www.springerlink.com/content/g7q15t2641587r7g.

[38] J. Reilly, M. Wilbur, M. Seibert, and N. Ahmadvand, "The complex subband decomposition and its application to the decimation of large adaptive filtering problems," *IEEE Trans. Signal Processing*, vol. 50, no. 11, pp. 2730–2743, Nov 2002.

[39] T. Schon, F. Gustafsson, and P. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2279–2289, July 2005.

[40] E. Serpedin and G. Giannakis, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE proceedings. Radar and signal processing Part F.*, vol. 140, no. 2, pp. 107–113, Apr 1993.

[41] ——, "A simple proof of a known blind channel identifiability result," *IEEE Trans. Signal Processing*, vol. 47, no. 2, pp. 591–593, Feb 1999.

[42] D. Simon and T. Chia, "Kalman filtering with state equality constraints," *IEEE Trans. Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128–136, Jan 2002.

[43] D. Simon and D. Simon, "Aircraft turbofan engine health estimation using constrained Kalman filtering," *Journal of Engineering for Gas Turbines and Power*, vol. 127, no. 2, pp. 323–328, 2005.

[44] L. Tong and S. Perreau, "Multichannel blind identification: from subspace to maximum likelihood methods," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1951–1968, Oct 1998.

[45] S. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction.* New York, NY: John Wiley & Sons, 2006.

[46] J. Vermaak, C. Andrieu, A. Doucet, and S. Godsill, "Particle methods for Bayesian modeling and enhancement of speech signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 3, pp. 173–185, Mar 2002.

[47] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *Adaptive Systems for Signal Processing, Communications, and Control (AS-SPCC) Symposium 2000. Proceedings.*, pp. 153–158, 2000.

[48] G. Welch and G. Bishop, "An introduction to the Kalman filter." [Online]. Available: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[49] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Trans. Signal Processing*, vol. 43, no. 12, pp. 2982–2993, Dec 1995.

[50] D. Yee, "Sequential Monte Carlo methods and their applications," Master's thesis, McMaster University, 2005.