Advances in Wideband Array Signal Processing Using Numerical Bayesian Methods

ADVANCES IN WIDEBAND ARRAY SIGNAL PROCESSING USING NUMERICAL BAYESIAN METHODS

BY

WILLIAM NG

SEPTEMBER 2003

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

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2003) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario

TITLE:	Advances in Wideband Array Signal Processing Using Numerical Bayesian Methods
AUTHOR:	William Ng M. Eng. (Electrical Engineering) McMaster University, Ontario, Canada
SUPERVISOR:	Dr. James P. Reilly, and Dr. Thia Kirubarajan
NUMBER OF PAGES:	xxi, 213

To my beloved parents

Abstract

This thesis focuses on joint model order detection and estimation of the parameters of interest, with applications to narrowband and wideband array signal processing in both *off-line* and *on-line* contexts. A novel data model that is capable of handling both narrowband and wideband cases with the use of an interpolation function and signal samples is proposed. In the off-line mode, Markov Chain Monte Carlo methods are applied to obtain a numerical approximation of the joint posterior distribution of the parameters under the condition that they have stationary distribution functions. On the other hand, if the distribution functions are nonstationary, the on-line approach is used. That approach employs a sequential implementation of Monte Carlo methods, applied to probabilistic dynamic systems.

Four inter-related problems were addressed in the course of this thesis.

1. A new data structure based on interpolation functions and signal samples to approximate wideband signals was developed. This data model, after appropriate transformation, has similar features found in the conventional narrowband data model. Furthermore, as the novel data model is developed for the wideband scenario, it can also address the narrowband scenario without change of structure or parameters. This novel data model is the basis on which the MCMC and the SMC approaches solve the array signal processing problems developed in the subsequent chapters.

2. The first algorithm presents an advanced approach using sequential MC methods to beamforming for narrowband signals in white noise with unknown variance. Traditionally, beamforming techniques assume that the number of sources is given and the

signal of interest (or target) is stationary within an observation period. However, in reality these two assumptions are commonly violated. The former assumption can be dealt with by jointly estimating the number of sources, whereas the latter severely limits the usefulness of conventional beamforming techniques when the target is indeed moving. In the case where the sources are moving, tracking the incident angles of the sources are required, and the accuracy of such tracking significantly affects the performance of signal separation and recovery, which is the objective of beamforming. The proposed method is capable of recursively estimating the time-varying number of sources as well as incident angles of the sources as new data arrive such that the signal amplitudes can be separated and restored in an on-line fashion.

3. The second algorithm presents an application of MCMC methods for the joint detection and estimation problem for the wideband scenario in white noise with unknown variance. In general, compared to the narrowband scenario, it is more difficult and cumbersome to solve this array signal processing problem in the wideband context. Conventional approaches tend to solve this problem in the frequency domain, and as such require a considerable amount of data to sustain accuracy, which imposes a large computational burden for these approaches. Furthermore, these approaches employ separate algorithms like AIC and MDL to estimate the number of sources. In contrast, the proposed method utilizes the reversible jump MCMC technique that simultaneously detects the number of sources and estimates the parameter of interest within the same algorithm. The proposed method is applied to the novel data model mentioned earlier and solves the problem in the time domain, which significantly reduces the requirement for a large number of data samples.

4. The final algorithm is an extension of the off-line approach to wideband array signal processing problem using sequential MC methods. Most conventional array signal processing approaches are developed under the assumption that the sources are stationary in direction of arrival. If this assumption is invalid, the solutions from these approaches become suboptimal and their performance is significantly degraded. When

sources are nonstationary, tracking the motions of the sources is needed, but in wideband scenarios the same problem becomes more difficult and cumbersome than in narrowband scenario because the methods for wideband scenarios usually require a considerable amount of data for processing. The proposed algorithm focuses on the sequential implementation of particle filters for probabilistic dynamic systems. This algorithm is applied to the modified novel data structure mentioned earlier in white noise with unknown variance for recursive estimation of the motions of the sources as new data arrive. A systematic statistical testing procedure is used to keep track of the number of sources.

Acknowledgements

The last four years have been both very challenging and very enjoyable; I had the opportunity to work on a very interesting topic about which I had no prior knowledge, but for which the likelihood of success was quite well defined, making for a very promising posterior situation.

I wish to thank Dr. Jim Reilly and Dr. Thia Kirubarajan for co-supervising and supporting my work. Their enthusiasm, approachability and patience were essential to the completion of this thesis. I am also very grateful to Dr. Alex Gershman and Dr. Tim Davidson for their most insightful comments and discussions.

I especially want to express my gratitude to Dr. J.R. Larocque for his experience in MCMC and sequential MC methods. Further thanks are also due to the ECE department and staff, particularly Ms. Cheryl Gies, who solved all of my non-research related problems.

The God and my parents are a great source of support and inspiration! Last but not least, I would like to thank my mother for her uninterrupted love and encouragement which played an instrumental role in my completing this project.

To all you involved in this chapter of my life - THANK YOU!

The research reported in this thesis has been made possible by the financial support from the National Sciences and Engineering Research Council of Canada, the Canadian Wireless Telecommunications Association, the ECE Department of McMaster University, CITO (Communications and Information Technology Ontario) and Mitel Corporation.

Notation and Acronyms

Symbol	Definition
x	Scalar
$\boldsymbol{x}$	Vector
$\boldsymbol{Y}$	Matrix
$\boldsymbol{Y}^T$	Matrix transpose
$\boldsymbol{Y}^H$	Hermitian transpose
$\lvert\boldsymbol{Y}\rvert$	Determinant
$\mathrm{Tr}(\boldsymbol{Y})$	Trace
Array Processing	
M	Number of array elements
L	Number of taps in the interpolation function
k	Model order (Number of sources)
N_t	Number of snapshots (observations)
$\boldsymbol{\phi}$	Physical directions of arrival
$\boldsymbol{S}(\boldsymbol{\phi})$	Steering matrix
$\boldsymbol{s}(\boldsymbol{\phi})$	One column of the steering matrix
$\boldsymbol{\tau}$	Inter-sensor delays
$\tilde{\boldsymbol{H}}(\boldsymbol{\tau})$	Interpolation matrix
$\tilde{\boldsymbol{h}}(\boldsymbol{\tau})$	One column of the interpolation matrix
ω_0	Carrier frequency in rad/s
f	Carrier frequency in Hz

Δ	Spacing (for a uniform linear array)
C	Velocity of propagation
$\boldsymbol{R}_{yy}$	Covariance matrix of the observations $\boldsymbol{y}$
λ	Eigenvalue
$\boldsymbol{e}_l$	Eigenvector corresponding to the lth eigenvalue
$\mathcal{N}$	Noise subspace
$\mathcal{S}$	Signal subspace
Statistics	
N	Number of samples (particles)
$P[A]$	Probability of the event A
$p(x)$ or $p(\boldsymbol{x})$	Probability density function (pdf) of x or $\boldsymbol{x}$
$\boldsymbol{x} \sim p(\boldsymbol{x})$	$\boldsymbol{x}$ is distributed or drawn from $p(\boldsymbol{x})$
$p(\boldsymbol{x}\|\boldsymbol{\mu}; \sigma^2)$	Density of $\boldsymbol{x}$ conditional on $\boldsymbol{\mu}$, with the knowledge of the parameter σ^2
$l(\boldsymbol{x})$	Likelihood function where $\boldsymbol{x}$ is the unknown parameter
$\mathcal{L}(\boldsymbol{x})$	Log-likelihood function, $\log(l(\boldsymbol{x}))$
$\pi(\boldsymbol{x})$	Posterior density of interest $p(\boldsymbol{x}\|\boldsymbol{y})$
$q(\cdot)$	Importance (or proposal) function
r	Acceptance ratio
α	Acceptance probability
$w_t^{(i)}$	The weight of the (i)th particle at time t
Acronyms	
BIS	Bayesian Importance Sampling
iid	Independant and Identically Distributed
IS	Importance Sampling
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
pdf	Probability Density Function

SIS	Sequential Importance Sampling
CRLB	Cramér-Rao Lower Bound
PCRB	Posterior Cramér-Rao Bound
SNR	Signal-to-Noise Ratio
DOA	Direction of Arrival
ISD	Inter-sensor Delay

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Statistical methods for signal processing have a wide range of different applications, such as, for example, radar, sonar, wireless communications, telephony, and geophysics. However, the objectives are often the same, namely, the extraction of parameters of interest from noisy observations.

Array signal processing focuses on signals conveyed by propagating waves. An array of sensors located at distinct spatial locations is deployed to measure a propagating electromagnetic, acosutic, or seismic wavefield. The goals of array signal processing are to combine sensors' outputs *cleverly* so as

- to **characterize** the field by *detecting* the number of sources and *locating* these sources;
- to **track** the *instantaneous positions* of the sources as they move in space; and
- to **enhance** the *quality* of the target sources by spatial filtering the interfering sources and noises.

Depending on the application, the processing might be "batch mode" or "off-line", where the data are collected before processing, or "sequential" or "online", when the algorithm proceeds as the observations are collected. There are many classical methods of either type, addressing each of the objectives. In this chapter, a review of the fundamentals of array

signal processing and the traditional algorithms developed to achieve the aforementioned goals are given.

1.1 Array signal processing

Sensor arrays have been in use for several decades in many practical signal processing applications. An array is used to filter signals in a space-time field by exploiting their spatial and temporal characteristics. This filtering may be expressed in terms of a dependence upon angle or wavenumber.

An array consists of a set of sensors that are located at different points in space with reference to a common reference point. These sensors *listen* to the incoming signals and provide a means of sampling these signals in space. Depending on the sensor characteristics and the propagation path, the source waveforms undergo deterministic and/or random modifications. The sensor outputs are composed of these signal components and additive contaminations such as measurement and thermal noise. The outputs are combined such that target signals from a set of angles are enhanced by a constructive combination and unwanted signals from other angles are rejected by destructive combination.

Sensor array systems can be divided into two classes: *active* and *passive*. In active sensing situations, a known waveform of finite duration is generated, which in turn propagates through a medium and is reflected by some target back to the point of origin. The transmitted signal is usually modified both in amplitude and phase by the target characteristics, which by themselves might be changing with time and its position in space. These disturbances give rise to a random return signal. In the passive context the signal received at the array is self-generated by the target, such as propeller or engine noise from submarines.

Applications for sensor arrays (Johnson, 1982; Van Trees, 2002) include the following areas:

1. **Radar** – Radar is the area in which antenna arrays were first used. Most radar systems are active, and the antenna array is used for both transmission and reception of signals. Radar technologies are used in military applications, including ballistic

missile detection and numerous airbone systems. On the other hand, non-military applications include air traffic control, depth-sounding, impulse radar, etc.

2. **Radio Astronomy** – Unlike radar systems, radio astronomy systems are passive and are used to detect celestial objects and estimate their characteristics. These systems usually employ arrays with very long baselines, extending from hundreds of meters to nearly the diameter of the earth.

3. **Sonar** – Sonar systems can be active or passive. The theory of active sonar systems has much in common with radar, but sonar systems deal with acoustic energy into the water while the radar systems deal with electromagnetic energy in the atmosphere. The main application in sonar systems is the detection and tracking of submarines, and in the fishing industry for detecting schools of fish.

4. **Seismology** – There are two main areas of seismology in which array processing plays an important role. The first area is the detection and location of underground nuclear explosions. The other area is exploration seismology that is to construct an image of the subsurface in which the structure and physical properties are described.

5. **Tomography** – Tomography is the cross-sectional imaging of objects from transmitted or reflected data. The object is illuminated from a number of different directions and data are collected at a receiving array. The cross-sectional image can then be reconstructed from the data. Medical diagnosis and treatment are examples of successful applications of tomography.

6. **Communications** – Antenna arrays are used in many communication systems. Several satellite systems utlize phased arrays in either the earth terminal or space segment for applications like tracking and data relay. Wireless cellular systems also utilize various types of multiple access techniques such as Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), and Global System for Mobile Communications (GSM).

Array processing deals with methods for processing the sensors' output data in the above applications in order to obtain insight into the structure of the waves traversing the array. The practical problems of interest in array signal processing are extracting the desired parameters such as the directions-of-arrival, power levels and crosscorrelations of the signals present in the scene from the available information including the measured data. Often one may also be specifically interested in the actual waveform corresponding to one of these sources, and in that case it is necessary to estimate the actual waveform associated with the desired signal while at the same time supress the other signals.

At times, the desired signal structure might be only known partially, and the objective in that case is to detect its presence in the available noisy data. This situation is often encountered in sonar to detect the presence of the signature of a specific class of submarine. Though the signal structure is known, it may still contain unknown parameters such as angle-of-arrival or random phase.

All these problems fall into one of two categories: detection or estimation of signals. In cases where the angle-of-arrival of the signals are time-varying, instantaneous estimation or *tracking* will be deployed.

In an array signal processing scenario, we have N_t vector observations $\boldsymbol{y}(n)$, each is of dimension M, for $n = 1, \ldots, N_t$ from an array of M sensors, which is illuminated by k_o plane waves incident onto the array from angles $\phi_1, \ldots, \phi_{k_o}$ relative to the normal of the array. The objective is, given the observations, to estimate the parameter k_o and the corresponding directions of arrival. In cases where the directions of arrival and/or the number of unknown sources are nonstatic and time-varying, the estimation and detection problems become a *tracking* problem where both parameters are recursively estimated, given the latest observation.

1.1.1 Array geometries

In the context of array signal processing, several assumptions are necessary to support the development of the algorithms.

- *Plane waves* – The sources are assumed far enough from the array so that the incident signals have planar wave fronts.
- *Narrowband* – Under this assumption, propagation over the length of the array is a function of only the phase of the incident wave. In this case, the amplitude of the incident wave is stationary over the length of time required for a point on the wave to completely traverse the array.
- *Wideband* – Under this assumption, signals occupy a significant frequency band, and hence the propagation delays cannot be represented by phase shifts as with narrowband signals. Here, the waveform changes during the time it interacts with the array.
- *Calibration* – The sensors are assumed calibrated, namely their radiation pattern is assumed known for all ϕ (Tranter *et al.*, 1999).
- *Uncorrelated noise* – The noise samples are assumed uncorrelated with the signals.
- $k_o < M$ – The true number of sources impinging the array must be less than the number of elements composing the array.

When an advancing plane wave passes through a non-dispersive medium (Johnson, 1982), the signal output at any sensor element immersed in that medium can be represented as a time-advanced/delayed version of the signal relative to a reference element. Figure 1.1 depicts an example of a uniform linear array of sensors, where the time delay between two successive sensors when a signal is transversing along the array is

$$\tau = \frac{\Delta}{C} \sin \phi,$$

where Δ denotes the interspacing distance between two successive sensors, and C represents the velocity of propagation. With the absence of noise, let $y_0(t) = a(t)$ denote the signal at the reference element (sensor 0), where $a(t)$ is a plane wave impinging onto the array, and $y_1(t)$ be the output at the second sensor from the reference point in absolute units. Therefore, $y_1(t)$ can be related to $y_0(t)$ as follows

$$y_1(t) = y_0(t - \tau) = a(t - \tau).$$

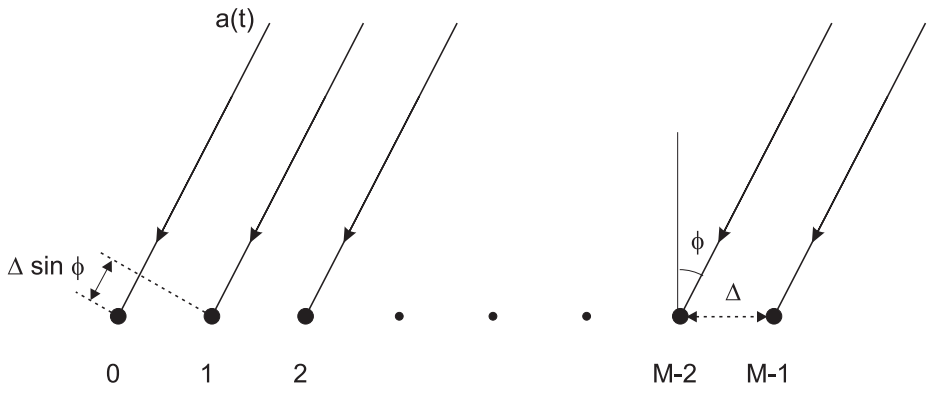

Figure 1.1: Definition of symbols for a linear antenna array scenario.

For all M sensors, the output vector, $\boldsymbol{y}(t)$, also known as the *observation*, can be written as

$$\begin{aligned}\boldsymbol{y}(t) &= [y_0(t), y_1(t), \ldots, y_{M-1}(t)]^T, \\ &= [a(t), a(t-\tau), \ldots, a(t-(M-1)\tau)]^T\end{aligned}$$

In the case where there are k_0 sources, the mth sensor output in response to k_0 signals from distinct angles of arrival, $\phi_k, k = 1, ..., k_0$, can be expressed as

$$y_m(t) = \sum_{k=0}^{k_0-1} a_k(t - m\tau_k),$$

where $a_k(t)$ is the kth source, and τ_k is the corresponding time delay between adjacent sensors, defined as

$$\tau_k = \frac{d}{C}\sin\phi_k.$$

1.1.2 Narrowband Situation

If the signals under consideration are narrowband, that is, the carrier frequency is fairly large compared to the bandwidth of the signal, then the signal can be treated as quasi-static during time intervals of order τ, and $y_m(t)$ can be rewritten as (Johnson, 1982; Buckley and

Griffiths, 1988; Pillai, 1989)

$$y_m(t) = \sum_{k=0}^{k_0-1} a_k(t) \exp\left(-j\frac{m\omega_o d \sin\phi_k}{C}\right),$$
$$= \sum_{k=0}^{k_0-1} a_k(t) \exp\left(-j\frac{m2\pi d \sin\phi_k}{\lambda}\right),$$

where ω_o is the operating frequency and λ is the associated carrier wavelength. As a result, the sensor output vector, $\boldsymbol{y}(t)$, can be then expressed as

$$\boldsymbol{y}(t) = \boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{a}(t),$$

where $\boldsymbol{\phi} \in \mathcal{R}^{k_o}$, and $\boldsymbol{S}(\boldsymbol{\phi}) \in \mathcal{C}^{M\times k_0}$ is known as the steering matrix, the kth column of which is defined as

$$\boldsymbol{s}(\phi_k) = \left[1, \exp\left(-j\frac{2\pi d \sin\phi_k}{\lambda}\right), \ldots, \exp\left(-j\frac{2\pi(M-1) d \sin\phi_k}{\lambda}\right)\right]^T. \tag{1.1}$$

For narrowband signals, the time delay can be approximated by a pure phase delay of the reference signal, and this phase delay depends only on the spacing between the sensors in question, the angle of arrival of the plane wave, and the frequency of the propagating wave.

The structure of the steering matrix varies with the geometry of the array. In particular, when the array elements are arranged in a straight line and are uniformly spaced, we assume $\Delta \leq \frac{\lambda}{2}$, where λ is the wavelength. However, this structure is subject to an ambiguity problem, for it can resolve only one angular component,[1] leading to a cone of uncertainty and right/left ambiguities. To resolve this problem, a circular array can be used. Nevertheless, because of its simplicity, the uniform linear array is commonly used in the literature, and for this reason this thesis adopts this simple structure.

With the consideration of observation noise, the snapshot vector at the nth sampling time n can then be expressed as

$$\boldsymbol{y}(n) = \boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{a}(n) + \boldsymbol{\nu}(n) \quad n = 1, 2, \ldots, N_t, \tag{1.2}$$

[1]For example, in the case of a vertically erected linear array, azimuth angle cannot be resolved, and an ambiguity exists as to whether the wave is incident from the front or back of the array.

where $\boldsymbol{\nu}(n)$ is the observation noise. We assume that the noise is a Gaussian[2] random variable. Various methods have been developed according to this narrowband model to resolve the detection and estimation problems.

1.1.3 Wideband Situation

While the time delays of narrowband signals can be approximated by their respective phase delays, wideband signals require more signal processing prior to applying existing approaches to solve for detection and estimation problems.

Using the *time shifting* property of the Fourier Transform, it is apparent that the mth sensor output can be rewritten as

$$\begin{aligned} Y_m(\omega) &= \mathrm{FT}\left[\sum_{k=0}^{k_0-1} a_k(n - m\tau_k)\right], \\ &= \sum_{k=0}^{k_0-1} A_k(\omega)\exp\left(-jm\omega\tau_k\right), \end{aligned}$$

where FT$[\cdot]$ is the Fourier Transform operator, $A_k(\omega)$ represents the Fourier Transform of $a_k(n)$. Thus the snapshot vector in frequency domain can be written as

$$\boldsymbol{Y}(\omega) = \boldsymbol{S}(\boldsymbol{\tau}, \omega)\boldsymbol{A}(\omega),$$

where $\boldsymbol{S}(\boldsymbol{\tau}, \omega) \in \mathcal{C}^{M\times k_0}$ is known as the location matrix, the kth column of which is defined as

$$\boldsymbol{s}(\tau_k, \omega) = \left[1, \exp\left(-j\omega\tau_k\right), \ldots, \exp\left(-j(M-1)\omega\tau_k\right)\right]^T. \tag{1.3}$$

Structurally, (1.1) and (1.3) are identical, but $\boldsymbol{s}(\tau_k, \omega)$ for $k = 0, ..., k_0 - 1$ is dependent on every ω. The steering matrix and the location matrix are structurally similar, and hence those approaches that were developed for narrowband signals can be applied to the wideband signals in the frequency domain. That is, the received data in the presence of noise is defined as follows

$$\boldsymbol{Y}(\omega) = \boldsymbol{S}(\boldsymbol{\tau}, \omega)\boldsymbol{A}(\omega) + \boldsymbol{\nu}(\omega), \tag{1.4}$$

[2]The definitions of the density functions can be found in Appendix A.

where $\boldsymbol{\nu}(\omega)$ is the Fourier Transform of the observation noise. As in the narrowband scenario, we assume the observation noise is a Gaussian random variable and uncorrelated with the signal sources. In Chapter 2 a novel data model, which is purely real and can handle both narrowband and wideband signal models, will be introduced.

1.2 Traditional array processing

In the field of array signal processing, there have been a variety of techniques to solve for detection and estimation problems for narrowband signals. However, for wideband signals it becomes cumbersome as a huge amount of data is required and more computational efforts are demanded to solve the same problems, even though in both cases the models are similar, where the only difference is in the domain where each model is defined. In this section, typical traditional techniques developed to solve for the detection and estimation problems for both narrowband and wideband case will be reviewed.

1.2.1 Narrowband Situation

For narrowband signals, assuming that the observation noise and the signals are uncorrelated, traditional array processing methods for model selection and direction finding purposes often use the information contained in the covariance matrix of the data

$$\begin{aligned}\boldsymbol{R}_{yy} &\triangleq E\left\{\boldsymbol{y}(n)\boldsymbol{y}^H(n)\right\},\\ &= E\left\{(\boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{a}(n)+\boldsymbol{\nu}(n))(\boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{a}(n)+\boldsymbol{\nu}(n))^H\right\},\\ &= \boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{R}_{aa}\boldsymbol{S}^H(\boldsymbol{\phi})+\boldsymbol{\Sigma}_\nu,\end{aligned}$$

where $\boldsymbol{y}(n)$ is defined in (1.2), $\boldsymbol{R}_{aa}$ is the signal covariance matrix, defined as

$$\boldsymbol{R}_{aa} \triangleq E\left\{\boldsymbol{a}(n)\boldsymbol{a}^H(n)\right\},$$

and $\boldsymbol{\Sigma}_\nu$ is the spatial covariance of the noise process, defined as

$$\boldsymbol{\Sigma}_\nu \triangleq E\left\{\boldsymbol{\nu}(n)\boldsymbol{\nu}^H(n)\right\} = \sigma_\nu^2\boldsymbol{I}_M,$$

where we assume the noise is white and Gaussian. The rank of the matrix $\boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{R}_{aa}\boldsymbol{S}^H(\boldsymbol{\phi})$ in (1.2.1) is the number of sources. In reality, the covariance matrix, $\boldsymbol{R}_{yy}$, is only available through the estimate, given as

$$\widehat{\boldsymbol{R}}_{yy} = \frac{1}{N_t}\sum_{n=1}^{N_t} \boldsymbol{y}(n)\boldsymbol{y}^H(n). \tag{1.5}$$

Using an eigen-decomposition on the sample covariance matrix, $\widehat{\boldsymbol{R}}_{yy}$, it is possible to decompose the space spanned by the eigenvectors of $\widehat{\boldsymbol{R}}_{yy}$ into two disjoint subspaces: the signal subspace, $\widehat{\mathcal{S}}_s$, and the noise subspace, $\widehat{\mathcal{S}}_n$. Given the covariance matrix is only estimated by a finite number of samples, these two subspaces are also estimates. Let the estimated eigenvectors of $\widehat{\boldsymbol{R}}_{yy}$ by the set $[\widehat{\boldsymbol{e}}_0, \ldots, \widehat{\boldsymbol{e}}_{M-1}]$, and $\widehat{\boldsymbol{E}}_S$ and $\widehat{\boldsymbol{E}}_N$ be the approximate signal and noise subspaces, respectively, defined as follows

$$\widehat{\boldsymbol{E}}_S = [\widehat{\boldsymbol{e}}_0, \ldots, \widehat{\boldsymbol{e}}_{k_o-1}],$$
$$\widehat{\boldsymbol{E}}_N = [\widehat{\boldsymbol{e}}_{k_o}, \ldots, \widehat{\boldsymbol{e}}_{M-1}],$$

where

$$\widehat{\boldsymbol{E}}_S^H \widehat{\boldsymbol{E}}_N \approx \boldsymbol{0}.$$

That is, these two sets of vectors are approximately orthogonal. It can be shown that the signal and noise subspaces can be respectively represented in terms of the eigenvectors as

$$\widehat{\mathcal{S}} = \text{span}[\widehat{\boldsymbol{E}}_S] = \text{span}[\widehat{\boldsymbol{e}}_0, \ldots, \widehat{\boldsymbol{e}}_{k_o-1}],$$
$$\widehat{\mathcal{N}} = \text{span}[\widehat{\boldsymbol{E}}_N] = \text{span}[\widehat{\boldsymbol{e}}_{k_o}, \ldots, \widehat{\boldsymbol{e}}_{M-1}].$$

1.2.1.1 Direction finding

Subspace methods With the knowledge of the number of sources, the signal subspace and noise subspace can be estimated with the corresponding eigenvectors. There are three classical methods that can estimate the directions of arrival by exploiting a fundamemtal property between the two sets of eigenvectors $\widehat{\boldsymbol{E}}_S$ and $\widehat{\boldsymbol{E}}_N$. These methods are the Pisarenko pseudo spectrum (Pisarenko, 1973), the *mu*ltiple *s*ignal *c*lassification (MUSIC) algorithm (Schmidt, 1986), and the Root-MUSIC (Barabell, 1983; Rao and Hari, 1989) method.

Given the fact that $\widehat{\boldsymbol{E}}_S^H \widehat{\boldsymbol{E}}_N \approx \mathbf{0}$ such that

$$\widehat{\boldsymbol{e}}_s^H \widehat{\boldsymbol{e}}_n \approx 0, \quad \widehat{\boldsymbol{e}}_s \in \widehat{\boldsymbol{E}}_S, \widehat{\boldsymbol{e}}_n \in \widehat{\boldsymbol{E}}_N,$$

if the signal vector is in the signal DOA's subspace, the magnitude of the projection onto $\widehat{\boldsymbol{E}}_N$ should be small. On the other hand, a projection onto $\widehat{\boldsymbol{E}}_S$ will result in a large magnitude. As a result, the search for signal vectors is equivalent to searching for vectors that are most closely orthogonal to the noise subspace as follows

$$||\boldsymbol{s}(\phi)^H \widehat{\boldsymbol{E}}_N|| \to 0 \quad \text{when } \phi \in \Theta,$$

where $\boldsymbol{s}(\phi)$ is a column of the steering matrix and Θ is the set of the true directions of arrival.

Based on this orthogonality property, the Pisarenko pseudo spectrum (Pisarenko, 1973) projects a steering vector onto a single noise eigenvector when estimating the DOAs. The Pisarenko pseudo spectrum for a particular noise eigenvector is defined as

$$\widehat{P}_{Pis}(\phi) = \frac{1}{\boldsymbol{s}^H(\phi)\widehat{\boldsymbol{e}}_k \widehat{\boldsymbol{e}}_k^H \boldsymbol{s}(\phi)}, \quad k \in [k_o, M-1].$$

It is well known that unless the number of sources is correct, it is not possible to tell which peaks shown in the Pisarenko pseudo spectrum are attributed to the sources and which are spurious.

Unlike the Pisarenko method, the MUSIC algorithm (Schmidt, 1986) involves a projection of a steering vector onto the whole noise subspace. The MUSIC pseudo spectrum is defined as

$$\widehat{P}_{MUSIC}(\phi) = \frac{1}{\boldsymbol{s}^H(\phi)\widehat{\boldsymbol{E}}_N \widehat{\boldsymbol{E}}_N^H \boldsymbol{s}(\phi)}.$$

Unlike the Pisarenko method, even if the number of sources is over estimated, the MUSIC method will perform well.

The Root-MUSIC algorithm (Barabell, 1983; Rao and Hari, 1989), developed specifically for uniform linear arrays (ULAs), is motivated by the fact that

$$\boldsymbol{s}^H(\omega_k)\boldsymbol{e}_j = 0, \quad j = k_o, ..., M-1,$$

where ω_k is a signal frequency, and $\omega_k = \frac{2\pi\Delta}{\lambda}\sin\phi_k$. Define the polynomials using the eigenvectors corresponding to the noise subspace, i.e.,

$$\boldsymbol{e}_j(z) = \frac{1}{\sqrt{M}}\sum_{m=0}^{M-1} e_{jm}z^{-m}, \quad j = k_o, ..., M-1,$$

such that the signal zeros, $z_k = e^{j\omega_k}$, $k = 0, ..., k_o - 1$, are roots of each of the above polynomials. Define another polynomial $\widehat{P}_{RMUSIC}(z)$ as follows

$$\begin{aligned}\widehat{P}_{RMUSIC}(z) &= \sum_{j=k_o}^{M-1} \boldsymbol{e}_j(z)\boldsymbol{e}_j^*(1/z^*),\\ &= \prod_{m=0}^{M-1}(1 - z_m z^{-1})(1 - z_m^* z),\\ &= D(z)D^*(1/z^*),\end{aligned}$$

where $D(z) = \prod_{m=0}^{M-1}(1 - z_m z^{-1})$ can be obtained by a spectral fractorization (Lee, 1960) of $\widehat{P}_{RMUSIC}(z)$ and has its roots inside or on the unit circle. The k_o signal zeros are the roots of $D(z)$ that are closest to the unit circle, i.e. $|z| = 1$, thereby recovering the signal angles ϕ_k, $k = 0, ..., k_o - 1$.

Maximum Likelihood methods When the *a priori* density of a parameter is not known, techniques must be developed that make no presumption about the relative possibilities of parameter values. In the event of DOA estimation, the maximum likelihood (ML) esitmator can be used that maximizes the likelihood function of the observation $\boldsymbol{y}(n)$, given other parameters.

According to the observation model in (1.2) with the noise being a Gaussian random variable, we can define the likelihood function as follows

$$l(\boldsymbol{\phi}; \boldsymbol{y}(n)) \triangleq p(\boldsymbol{y}(n)|\boldsymbol{\phi}) \sim \mathcal{N}(\boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{a}(n), \sigma_\nu^2 \boldsymbol{I}_M),$$

where $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ refers to a normal distribution with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$. Note that this is a function of the parameter $\boldsymbol{\phi}$ and not of $\boldsymbol{y}(n)$. The value of the parameter that maximizes this function is called the *maximum-likelihood* estimate, defined by

$$\widehat{\boldsymbol{\phi}}_{ML} = \arg\max_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} p(\boldsymbol{y}|\boldsymbol{\phi}). \tag{1.6}$$

The ML estimator in (1.6) can be interpreted as a search for a set of $\boldsymbol{\phi}$ such that the sum of the Euclidean distances between the observations $\boldsymbol{y}(n)$ and the estimates $\widehat{\boldsymbol{y}}(n|\boldsymbol{\phi})$ is minimized, i.e.,

$$\widehat{\boldsymbol{\phi}}_{ML} = \arg \min_{\boldsymbol{\phi} \in \boldsymbol{\Phi}} \sum_{n=1}^{N_t} ||\boldsymbol{y}(n) - \widehat{\boldsymbol{y}}(n|\boldsymbol{\phi})||^2. \tag{1.7}$$

The estimates $\widehat{\boldsymbol{y}}(n|\boldsymbol{\phi})$ are obtained by the least-squares estimate of the amplitudes $\boldsymbol{a}(n)$ as follows

$$\widehat{\boldsymbol{y}}(n|\boldsymbol{\phi}) = \boldsymbol{S}(\boldsymbol{\phi})\widehat{\boldsymbol{a}}(n), \tag{1.8}$$

where, according to (1.2), the least-squares estimate $\widehat{\boldsymbol{a}}(n)$ is given by

$$\widehat{\boldsymbol{a}}(n) = \left[\boldsymbol{S}^H(\boldsymbol{\phi})\boldsymbol{S}(\boldsymbol{\phi})\right]^{-1} \boldsymbol{S}^H(\boldsymbol{\phi})\boldsymbol{y}(n). \tag{1.9}$$

Defining a model dependent projector matrix by $\boldsymbol{P}_S(\boldsymbol{\phi})$ as

$$\boldsymbol{P}_S(\boldsymbol{\phi}) = \boldsymbol{S}(\boldsymbol{\phi}) \left[\boldsymbol{S}^H(\boldsymbol{\phi})\boldsymbol{S}(\boldsymbol{\phi})\right]^{-1} \boldsymbol{S}^H(\boldsymbol{\phi}), \tag{1.10}$$

and its orthogonal complement as

$$\boldsymbol{P}_S^{\perp}(\boldsymbol{\phi}) = \boldsymbol{I} - \boldsymbol{P}_S(\boldsymbol{\phi}),$$

we can rewrite the estimate $\widehat{\boldsymbol{y}}(n|\boldsymbol{\phi})$ as

$$\widehat{\boldsymbol{y}}(n|\boldsymbol{\phi}) = \boldsymbol{P}_S(\boldsymbol{\phi})\boldsymbol{y}(n), \tag{1.11}$$

such that the ML estimator in (1.6) becomes

$$\begin{aligned}
\widehat{\phi}_{ML} &= \arg \min_{\phi \in \Phi} \sum_{n=1}^{N_t} ||\boldsymbol{y}(n) - \widehat{\boldsymbol{y}}(n)||^2, \\
&= \arg \min_{\phi \in \Phi} \sum_{n=1}^{N_t} ||(\boldsymbol{I} - \boldsymbol{P}_S(\boldsymbol{\phi}))\boldsymbol{y}(n)||^2, \\
&= \arg \min_{\phi \in \Phi} \operatorname{tr}(\boldsymbol{P}_S^{\perp}(\boldsymbol{\phi})\widehat{\boldsymbol{R}}_{yy}), \\
&= \arg \max_{\phi \in \Phi} \operatorname{tr}(\boldsymbol{P}_S(\boldsymbol{\phi})\widehat{\boldsymbol{R}}_{yy}),
\end{aligned}$$

where $\mathrm{tr}(\cdot)$ is the trace operator, and $\widehat{\boldsymbol{R}}_{yy}$ is defined in (1.5). The same result can also be obtained in a Bayesian context (Reilly, 1981; Haykin, 2000; Wu and Wong, 1994; Viberg *et al.*, 1991).

Since the ML estimator does not take any prior knowledge of the parameters into account, it is expected that the error characteristics of the resulting estimates could be worse than those that can use prior knowledge. Moreover, like many other optimization problems, the ML estimation problem is well known to be difficult, as the function to optimize shows many saddle points and local extrema. Any gradient based method would need a good initialization in order to succeed.

1.2.1.2 Model selection

In the case where the noise is spatially white in (1.2), the eigendecomposition of $\boldsymbol{R}_{yy}$ yields M eigenvalues, $M - k_o$ of which correspond to the noise variance σ_ν^2 and the remaining k_o eigenvalues which are larger correspond to the signals, i.e.

$$\lambda_0 > \lambda_1 > \cdots > \lambda_{k_o-1} > \lambda_{k_o} = \lambda_{k_o+1} = \cdots = \lambda_{M-1} = \sigma_\nu^2.$$

There exist a few techniques using information theoretic criteria to estimate the model order. They are the Akaike information criteria (AIC) (Akaike, 1974), the minimum description length criteria (MDL) (Rissanen, 1978; Wax *et al.*, 1984), and the more recent D-MAP (Djuric, 1996). These approaches determine the best model order, given the data, by maximizing the log-likelihood function of the data over all possible model orders. Each method is composed of two terms – the log-likelihood function and the penalty function of the model order – and has the following general expression

$$L(\widehat{\boldsymbol{\phi}}_{ML}) = -\log f(\boldsymbol{y}|\widehat{\boldsymbol{\phi}}_{ML}) + g(k, M, N_t),$$

where

$$f(\boldsymbol{y}|\widehat{\boldsymbol{\phi}}_{ML}) = \left(\frac{(\prod_{i=k+1}^{M} \widehat{\lambda}_i)^{1/(M-k)}}{\frac{1}{M-k}\sum_{i=k+1}^{M} \widehat{\lambda}_i} \right)^{2(M-k)N_t},$$

$\widehat{\boldsymbol{\phi}}_{ML}$ is the ML estimate of the parameters for a specific model order, $\widehat{\lambda}_i$ are the estimated eigenvalues that are arranged in descending order, and $g(k, M, N_t)$ is a penalty term, which

is a function of the model order, k, the number of sensor elements, M, and the number of snapshots, N_t. The model order is estimated by maximizing $L(\widehat{\boldsymbol{\phi}}_{ML})$. Note that the term in the brackets in the likelihood function $f(\boldsymbol{y}|\widehat{\boldsymbol{\phi}}_{ML})$ is the ratio of the geometric and arithmetic means of the $M-k$ smallest eigenvalues of the estimated covariance matrix $\widehat{\boldsymbol{R}}_{yy}$. If the model order chosen is correct, the geometric and arithmetic means are approximately equal, which also corresponds to the point where the function $L(\widehat{\boldsymbol{\phi}}_{ML})$ is minimum. On the other hand, if the model order is too low, the arithmetic mean exceeds the geometric mean such that $L(\widehat{\boldsymbol{\phi}}_{ML})$ decreases. However, if the model order is too high, the penalty term $g(k, M, N_t)$ dominates and $L(\widehat{\boldsymbol{\phi}}_{ML})$ increases.

The AIC, MDL and D-MAP criteria for estimation of the number of sources can be written respectively as

$$\widehat{k}_{AIC} = \min_k \left\{ -\log f(\boldsymbol{y}|\widehat{\boldsymbol{\phi}}) + 2k(2M-k) \right\},$$

$$\widehat{k}_{MDL} = \min_k \left\{ -\log f(\boldsymbol{y}|\widehat{\boldsymbol{\phi}}) + \frac{k(2M-k)}{2}\log(N_t) \right\},$$

$$\widehat{k}_{DMAP} = \min_k \left\{ -\log f(\boldsymbol{y}|\widehat{\boldsymbol{\phi}}) + \frac{5k}{2}\log(N_t) \right\}.$$

There are a few points worth noting about these approaches. Since the whole concept of these information theoretic criteria relies on the accuracy of the computation of the sample covariance matrix $\widehat{\boldsymbol{R}}_{yy}$ and the associated eigenvalues λ_i, $i = 0, ..., M-1$, the number of snapshots plays a critical role in the entire detection process. In particular, when the SNR is low, more snapshots are required to sustain the necessary accuracy in model order detection. Furthermore, these approaches assume the noise is additive white Gaussian. It has been verified that their performance is very sensitive to this assumption. A performance analysis of these methods in non-white Gaussian noise, for varying degrees of color, is given in Chen (1991). Another alternative in array signal processing is the use of multiple arrays (Wong *et al.*, 1992; Fuchs, 1992; Wu and Wong, 1994; Nagesha and Kay, 1996; Chen *et al.*, 1996).

It is well known that the MDL criterion and the D-MAP yield more consistent estimates of the number of sources than the AIC criterion (Johnson, 1982; Zhao *et al.*, 1987), when

the number of observations is large. The AIC criterion usually yields a slightly larger value for the number of signals regardless of how many observations are used and how high is the SNR.

1.2.2 Wideband Situation

For wideband signals, most of the traditional approaches developed to solve detection and estimation problems convert the data model from the time-domain as defined in (1.2) to the frequency domain as defined in (1.4). Such a conversion is motivated by two reasons: One is that the data model in the frequency domain is structurally similar to that in the time domain, and the other is that all the successful techniques originally developed for narrowband models can be reused in the set of frequency bins to solve the same problems in question, namely detection and estimation.

In practice, a sufficiently long duration of sensor output is observed. Then, the sampled data are divided into groups of N_t snapshots, each containing J samples. In each snapshot, an FT algorithm is used to transform the data into the frequency domain. Thus, N_t sets of transformed data are available where each set contains J frequency samples of the spectrum of the observation vector. As a result, given a set of frequency samples, $\omega_j, j = 0, ..., J-1$, the sample covariance matrix $\widehat{\boldsymbol{R}}_{yy}(\omega_j)$ becomes

$$\begin{aligned}\widehat{\boldsymbol{R}}_{yy}(\omega_j) &= \mathrm{FT}\left[E\left\{\boldsymbol{y}(n)\boldsymbol{y}^H(n)\right\}\right]_{\omega=\omega_j}, \quad n = 1+j, ..., jN_t/J, \\ &= \boldsymbol{S}(\boldsymbol{\phi}, \omega_j)\widehat{\boldsymbol{R}}_{aa}(\omega_j)\boldsymbol{S}^H(\boldsymbol{\phi}, \omega_j) + \boldsymbol{\Sigma}_\nu(\omega_j),\end{aligned} \tag{1.12}$$

where $\widehat{\boldsymbol{R}}_{aa}(\omega_j)$ is the sample covariance matrix for the sources at the jth frequency bin. When J is sufficiently large, the bandwidth the spectrum of the signal occupied in every frequency bin becomes narrow, thereby approximately corresponding to that of a narrowband signal.

1.2.2.1 Direction finding

Focusing technique The Coherent Signal-subspace Method (CSM) (Valaee and Kabal, 1995) is a focusing technique that transforms the signal subspaces spanned by the columns

of $\boldsymbol{S}(\boldsymbol{\phi}, \omega_j)$ in all frequency bins $j = 0, ..., J-1$ and overlaps them in a predefined subspace, known as the focusing subspace (Valaee and Kabal, 1995). Given a predefined focusing frequency ω_F and hence the focusing location matrix $\boldsymbol{S}(\boldsymbol{\phi}, \omega_F)$, the objective of the focusing technique is to find the solutions $\boldsymbol{T}(\omega_j)$ of the equations given by

$$\boldsymbol{T}(\omega_j)\boldsymbol{S}(\boldsymbol{\phi}, \omega_j) = \boldsymbol{S}(\boldsymbol{\phi}, \omega_F), \quad j = 0, ..., J-1.$$

Using the focusing matrices $\boldsymbol{T}(\omega_j)$, the snapshots at different frequency bins can be transformed into the focusing subspace, i.e.,

$$\boldsymbol{y}^{(j)}(n) = \boldsymbol{T}(\omega_j)\boldsymbol{y}(n),$$

and then a set of sample covariance matrices at different frequency bins can be constructed as

$$\widehat{\boldsymbol{R}}_{yy}^{(j)} = \frac{J}{N_t} \sum_{n=1+j}^{jN_t/J} \boldsymbol{y}^{(j)}(n)\boldsymbol{y}^{(j)^H}(n), \quad j = 0, ..., J-1.$$

Eventually, a universal focused sample covariance matrix that can be used for detection and estimation can be obtained as follows

$$\begin{aligned} \widehat{\boldsymbol{R}}_{yy}^{(F)} &= \frac{1}{J} \sum_{j=0}^{J-1} \widehat{\boldsymbol{R}}_{yy}^{(j)}, \\ &= \boldsymbol{S}(\boldsymbol{\phi}, \omega_F)\widehat{\boldsymbol{R}}_{aa}^{(F)}\boldsymbol{S}^H(\boldsymbol{\phi}, \omega_F) + \boldsymbol{R}_{\nu}^{(F)}, \end{aligned} \tag{1.13}$$

where

$$\begin{aligned} \widehat{\boldsymbol{R}}_{aa}^{(F)} &= \frac{1}{J} \sum_{j=0}^{J-1} \widehat{\boldsymbol{R}}_{aa}(\omega_j), \\ \boldsymbol{R}_{\nu}^{(F)} &= \frac{1}{J} \sum_{j=0}^{J-1} \boldsymbol{T}(\omega_j)\boldsymbol{\Sigma}_{\nu}(\omega_j)\boldsymbol{T}^H(\omega_j). \end{aligned}$$

Given that the sample covariance matrix in (1.13) approximately corresponds to that for narrowband signals, it is then possible to apply narrowband methods to the wideband problem.

While the CSM algorithm improves the efficiency of the estimation by condensing the energy in the sub-bands into the focusing signal subspace, it suffers from a few problems

that degrade the overall performance. Firstly, the transformation involving the matrices $\boldsymbol{T}(\omega_j)$ will change the orginal noise structure as well as the SNR levels at the output of the processor. In particular, if the observation noise is Gaussian and white, the transformed noise is no longer white. Secondly, it was pointed out that the method suffers from an asymptotic bias of the peaks in the spatial spectrum. This bias increases with the bandwidth of the sources and the deviation of the focusing points from the true DOAs. Nevertheless, it was shown (Hung and Kaveh, 1988) that if the matrices $\boldsymbol{T}(\omega_j)$ are unitary transfomations, the focusing is lossless.

The Two-sided Correlation Transformation (TCT), which is another focusing technique and uses a similar focusing concept as in the CSM, performs the focusing transformation on the covariance matrix of the sources instead of the location matrix. The transformation matrix at each frequency bin is unitary and minimizes the distance between the focusing subspace and the transformed signal subspace. The TCT and the CSM differ in two areas. Firstly, the transformation of the subspaces using the TCT is performed through a two-sided transformation applied to the source covariance matrix, which can be shown to result in a smaller error. Secondly, given that many high resolution algorithms for DOA estimation are based on the eigen-decomposition of the covariance matrix, the TCT applies the transformation on the source covariance matrix instead of the location matrix.

Similar to the CSM procedures, a predefined focusing covariance matrix is given, and the objective of the TCT is to find all solutions of the transformation matrices that minimize the distance between the focusing covariance matrix and source covariance matrices in all frequency bins, under the contraint that the transformation matrices are kept unitary. Let $\widehat{\boldsymbol{R}}_{aa}^{(F)}$ be the focusing covariance matrix, and $\boldsymbol{U}(\omega_j)$ for $j = 0, ..., J-1$ be the focusing matrices. Then the TCT focusing matrices can be found by the following optimization

$$\begin{aligned} &\min_{\boldsymbol{U}(\omega_j)} ||\widehat{\boldsymbol{U}}^{(F)} - \boldsymbol{U}(\omega_j)\widehat{\boldsymbol{R}}_{aa}(\omega_j)\boldsymbol{U}^H(\omega_j)|| \\ &\text{s.t. } \boldsymbol{U}^H(\omega_j)\boldsymbol{U}(\omega_j) = \boldsymbol{I} \end{aligned},$$

for $j = 0, ..., J-1$. Once the set of focusing matrices is obtained, the transformed covariance

matrices at different frequency bins can be constructed as

$$\widehat{\boldsymbol{R}}_{yy}^{(j)} = \frac{J}{N_t} \sum_{n=1+j}^{jN_t/J} \boldsymbol{y}^{(j)}(n) \boldsymbol{y}^{(j)^H}(n), \quad j = 0, ..., J-1,$$

where

$$\boldsymbol{y}^{(j)}(n) = \boldsymbol{U}(\omega_j)\boldsymbol{y}(n).$$

Finally, a universal focused sample covariance matrix can be constructed as follows

$$\widehat{\boldsymbol{R}}_{yy}^{(F)} = \frac{1}{J} \sum_{j=0}^{J-1} \widehat{\boldsymbol{R}}_{yy}(\omega_j). \tag{1.14}$$

The covariance matrix in (1.14) is approximately equal to that for narrowband signals, and hence can be applied to narrowband methods previously developed for detection and estimation problems when the signals are narrowband.

If the observation noise is Gaussian and white, then a preprocessing step can be taken to reduce the noise components in the covariance matricies. This step requires a low-resolution beamformer to estimate the number of sources and the DOA's of the sources. With the knowledge of the estimated number of sources, performing an eigen-decomposition on the covariance matrices $\widehat{\boldsymbol{R}}_{yy}(\omega_j)$ at the jth frequency bin yields an estimate of the noise power as follows

$$\widehat{\sigma}_\nu^2(\omega_j) = \frac{1}{M-k_0} \sum_{m=k_0}^{M-1} \widehat{\lambda}_m(\widehat{\boldsymbol{R}}_{yy}(\omega_j)),$$

where $\widehat{\lambda}_m(\widehat{\boldsymbol{R}}_{yy}(\omega_j))$ for $m = 0, ..., M-1$ are the estimated eigenvalues of the of $\widehat{\boldsymbol{R}}_{yy}(\omega_j)$ that are arranged in descending order. Therefore, the source covariance matrices can be rewritten as

$$\widehat{\boldsymbol{R}}_{aa}(\omega_j) = \boldsymbol{B}(\omega_j) \left[\widehat{\boldsymbol{R}}_{yy}(\omega_j) - \widehat{\sigma}_\nu^2(\omega_j)\boldsymbol{I} \right] \boldsymbol{B}^H(\omega_j), \quad j = 0, ..., J-1,$$

where

$$\boldsymbol{B}(\omega_j) = \left[\boldsymbol{S}^H(\boldsymbol{\phi}, \omega_j)\boldsymbol{S}(\boldsymbol{\phi}, \omega_j) \right]^{-1} \boldsymbol{S}^H(\boldsymbol{\phi}, \omega_j)$$

As a result, the noise-free, focused sample covariance matrix can be simplified as

$$\widehat{\boldsymbol{R}}_{yy}^{(F)} = \boldsymbol{S}(\boldsymbol{\phi}, \omega_F)\widehat{\boldsymbol{R}}_{aa}^{(F)}\boldsymbol{S}^H(\boldsymbol{\phi}, \omega_F),$$

where

$$\widehat{\boldsymbol{R}}_{aa}^{(F)} = \frac{1}{J}\sum_{j=0}^{J-1} \widehat{\boldsymbol{R}}_{aa}(\omega_j),$$

$$\boldsymbol{S}(\boldsymbol{\phi}, \omega_F) = \boldsymbol{U}(\omega_j)\boldsymbol{S}(\boldsymbol{\phi}, \omega_j), \quad j = 0, ..., J-1$$

Finally, the desired, focused covariance matrix will be applied to appropriate algorithms for detection and estimation. The TCT method has a smaller subspace fitting error than the CSM, and has unbiased estimates of the DOAs, regardless of the bandwidth of the signals.

Both the CSM and the TCT techniques transform the wideband signals into a common subspace, where the transformed signals approximately become narrowband. Since these methods operate in the frequency domain, and the accuracy of the focusing relies on the size of the frequency bins, they require a relatively larger amount of data than other competing methods. Furthermore, it is assumed that an appropriately selected focusing matrix or focusing frequency ω_F will be given in each method, but such a selection could be arbitrary. When an inappropriate focusing matrix is selected, the overall detection and estimation performance is significantly degraded. In particular, when the SNR levels are low, the model order detection by AIC or MDL becomes inconsistent, which adversely affects the eventual performance of DOA estimation.

Asymptotic Maximum-Likelihood Methods The asymptotic maximum-likelihood (AML) methods (Boehme, 1986, 1989; Van Trees, 2002) are an extension of the ML estimators. In particular, when the signals in the estimation problem are wideband and J is sufficiently large and there is no frequency correlation in the sources, the set of frequency spectra $\boldsymbol{Y}(\omega_j)$ for all j can be considered independent so that the AML methods can perform DOA estimation jointly in J frequency bins.

Denote the AML estimate of the DOAs by $\widehat{\boldsymbol{\phi}}_{AML}$. Using the definitions in Section 1.2.2,

we define $\widehat{\boldsymbol{P}}_j$ as the projection matrix for $\omega = \omega_j$ onto the range of $\boldsymbol{S}(\widehat{\boldsymbol{\phi}}_{AML}, \omega_j)$, given by

$$\widehat{\boldsymbol{P}}_j = \boldsymbol{S}(\widehat{\boldsymbol{\phi}}_{AML}, \omega_j)[\boldsymbol{S}(\widehat{\boldsymbol{\phi}}_{AML}, \omega_j)\boldsymbol{S}^H(\widehat{\boldsymbol{\phi}}_{AML}, \omega_j)]^{-1}\boldsymbol{S}^H(\widehat{\boldsymbol{\phi}}_{AML}, \omega_j), \quad j = 0, ..., J-1, \tag{1.15}$$

and $\widehat{\boldsymbol{C}}_{yy}(\omega_j)$ as the sample covariance matrix for the observations at ω_j, given by

$$\widehat{\boldsymbol{C}}_{yy}(\omega_j) = \frac{1}{N_t}\sum_{n=1}^{N_t} \boldsymbol{Y}(\omega_j)\boldsymbol{Y}^H(\omega_j), \quad j = 0, ..., J-1. \tag{1.16}$$

Accordingly, the orthogonal complement of $\widehat{\boldsymbol{P}}_j$ is given by

$$\widehat{\boldsymbol{P}}_j^{\perp} = \boldsymbol{I} - \widehat{\boldsymbol{P}}_j. \tag{1.17}$$

Depending on the knowledge of the observation noise variance σ_ν^2 at ω_j, there are three cases by which the total likelihood function $\mathcal{L}_l(\widehat{\boldsymbol{\phi}}_{AML}), l = 1, 2, 3$, can be defined (Van Trees, 2002), provided the number of sources k_o is available. In each case, the AML estimate $\widehat{\boldsymbol{\phi}}_{AML}$ can be obtained by

$$\widehat{\boldsymbol{\phi}}_{AML} = \arg\max_{\phi \in \boldsymbol{\Phi}} \sum_{j=0}^{J-1} \mathcal{L}_l(\widehat{\boldsymbol{\phi}}_{AML}), \quad l = 1, 2, 3. \tag{1.18}$$

1. When $\eta_j \triangleq \sigma_\nu^2(\omega_j)$, $j = 0, ..., J-1$, is known, the total likelihood function $\mathcal{L}_1(\widehat{\boldsymbol{\phi}}_{ML})$ is defined by

$$\mathcal{L}_1(\widehat{\boldsymbol{\phi}}_{ML}) = \sum_{j=0}^{J-1} \left\{ \frac{-1}{\eta_j} \mathrm{tr}\left[\widehat{\boldsymbol{P}}_j^{\perp}\widehat{\boldsymbol{C}}_{yy}(\omega_j)\right] - \log\det\left[\widehat{\boldsymbol{P}}_j\widehat{\boldsymbol{C}}_{yy}(\omega_j)\widehat{\boldsymbol{P}}_j + \eta_j\widehat{\boldsymbol{P}}_j^{\perp}\right]\right\}. \tag{1.19}$$

2. When $\eta_j \triangleq \sigma_\nu^2(\omega_j)$, $j = 0, ..., J-1$, is unknown, then an estimate of η_j is computed as follows

$$\widehat{\eta_j} = \frac{\mathrm{tr}\left[\widehat{\boldsymbol{P}}_j^{\perp}\widehat{\boldsymbol{C}}_{yy}(\omega_j)\right]}{M - k_o}, \tag{1.20}$$

such that the total likelihood function $\mathcal{L}_2(\widehat{\boldsymbol{\phi}}_{AML})$ is given by

$$\mathcal{L}_2(\widehat{\boldsymbol{\phi}}_{AML}) = -\varphi(\widehat{\boldsymbol{\phi}}_{AML}) - (M - k_o)\sum_{j=0}^{J-1} \log\left\{\mathrm{tr}\left[\widehat{\boldsymbol{P}}_j^{\perp}\widehat{\boldsymbol{C}}_{yy}(\omega_j)\right]\right\}, \tag{1.21}$$

where

$$\varphi(\widehat{\boldsymbol{\phi}}_{AML}) = \sum_{j=0}^{J-1} \log\det\left[\widehat{\boldsymbol{P}}_j \widehat{\boldsymbol{C}}_{yy}(\omega_j)\widehat{\boldsymbol{P}}_j + \widehat{\boldsymbol{P}}_j^{\perp}\right]. \tag{1.22}$$

3. When $\eta_j \triangleq \sigma_\nu^2(\omega_j) = \sigma_\nu^2$, $j = 0, ..., J-1$, and σ_ν^2 is unknown, then an estimate of η_j is computed as follows

$$\widehat{\eta} = \frac{\sum_{j=0}^{J-1} \mathrm{tr}\left[\widehat{\boldsymbol{P}}_j^{\perp}\widehat{\boldsymbol{C}}_{yy}(\omega_j)\right]}{J(M-k_o)} \tag{1.23}$$

such that the total likelihood function $\mathcal{L}_3(\widehat{\boldsymbol{\phi}}_{AML})$ is given by

$$\mathcal{L}_3(\widehat{\boldsymbol{\phi}}_{AML}) = -\varphi(\widehat{\boldsymbol{\phi}}_{AML}) - J(M-k_o)\log\sum_{j=0}^{J-1}\mathrm{tr}\left[\widehat{\boldsymbol{P}}_j^{\perp}\widehat{\boldsymbol{C}}_{yy}(\omega_j)\right]. \tag{1.24}$$

Parameter estimation by maximum likelihood in the frequency domain is a possibility, but the global search in the parameter space of interest is computationally complex and generally not used in practice. In particular, to sustain a high accuracy in the estimation, the number of frequency bins J and the number of snapshots N_t need to be large, adding more computational burden to the algorithm. Furthermore, the knowledge of the model and the initial estimates play a critical role in the performance of the method. The estimates can be unreliable if the model mismatches or the initial estimates are not precise enough. More detailed discussion of the AML estimator for wideband signals can be found in (Doron and Weiss, 1992; Doron *et al.*, 1993).

1.3 The Bayesian approach

In the context of parameter estimation, the more information about the parameters that is available and can be used, the better the estimation performance. The maximum-likelihood methods described earlier do not take any prior knowledge of the parameters into account, so the error characteristics of the resulting estimates is expected to be worse than those that can use it. To address this problem, Bayesian methods, an alternative approach to the

estimation problem that ultilizes the prior distributions of the parameters, can be adopted. The Bayesian approach to statistics provides a theoretical and practical framework with which to view many statistical problems. Bayesian methods have become increasingly popular because they provide solutions to many previously intractable problems. The Bayesian framework gives analysts the ability to use prior knowledge, and the option of marginalizing with respect to the parameters of interest, for evaluation of confidence regions, etc.

In spite of their inherent benefits, the Bayesian methods originally were not widely adopted and implemented in practice. First, the models and the resulting posterior distributions so formed can be highly non-linear and complex such that analytic optimization or integration is prohibitive. To solve any optimization or integration problems, numerical methods must be employed. Second, as numerical approximation of these optimizations or integrations can be adopted, the associated computational complexity of these methods is usually high. In particular, in the days when the computer power was low and expensive, the numerical approach of Bayesian methods were seldom adopted. Third, relying on the use of the Bayes' theorem, the Bayesian methods always lead to controversial debates on the issue of inclusion and selection of the prior distributions of the parameters.

Since afforable computer power has been available over the last decade, the Bayesian approach to statistical problems has become increasingly popular. This has been brought about by the adoption and development of Markov chain Monte Carlo (MCMC) methods. Proposed in the early fifties and originated in the statistical physics literature, the MCMC methods were first ignored due to their intense computational requirements. However, with the ever growing power of personal computers, they have just recently matured into very powerful algorithms, providing a potential solution to difficult problems, when no other solution would exist.

MCMC is essentially Monte Carlo numerical integration using Markov chains. The MCMC process consists of drawing samples according to an arbitrary probability density function, by running a cleverly constructed Markov chain for a long time. These samples are then used to construct a histogram to approximate the desired probability density function as well as to numerically evaluate statistical inference on parameters. In cases where the

parameters of interest are stationary, MCMC methods have demonstrated their profound effect in parameter estimation problems. When the stationarity assumption is not valid, i.e., where the parameters evolve over time, the objective of the estimation problem is to recursively estimate the probability density functions associated with a dynamic system as new data become available. As ordinary MCMC methods cannot handle this so-called *tracking* problem, sequential Monte Carlo methods otherwise known as *particle filters*, which rely on a recursive update equation for a set of weights to approximate the time-varying posterior distribution functions of the parameters, are adopted.

In summary, MCMC and sequential MC methods are capable of solving problems that are intractable using conventional methods. These problems include

- *Optimization*. The objective is to locate the extrema of the function. A complicated optimization procedure is replaced by a simple location searching approach.
- *Integration*. Marginalizing parameters would require complicated non-linear and very often prohibitive analytical integrations. Numerical marginalization of a parameter is simplified to summing bins of the histogram.
- *Simulation*. In some applications, the objective is simply to draw samples from an arbitrary density function, which is only possible in general when this function is either standard or integrable. MCMC methods have proved powerful in this context.

In Chapter 2, we will provide a detailed description of the MCMC and the sequential MC approaches and some popular sampling procedures.

1.3.1 Bayes' theorem

In the Bayesian approach, one starts with the prior distribution function of the parameter from which one can obtain its posterior distribution function using Bayes' formula. Denote the parameter of interest by x and the corresponding prior distribution function by $p(x)$. Likewise, we define y as an observation variable and $p(y)$ as the corresponding prior distribution function. The prior distribution functions contain any information regarding the

variables. Then the joint probability density function $p(y, x)$ is given by

$$p(y, x) = p(y|x)p(x) = p(x|y)p(y), \tag{1.25}$$

where $p(y|x)$ is the likelihood function and $p(x|y)$ is the posterior distribution function of x.

Bayes' formula is defined as the conditional distribution of the parameter x, given the observation y, which is the posterior distribution function of x, as follows

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{c}, \tag{1.26}$$

where $c = p(y)$ is the normalization constant, which is independent of x. Depending on whether the probability density function is continuous or discrete, the normalization constant can be evaluated as follows

$$c = \begin{cases} \int p(y|\phi)p(x)d\phi & \text{continuous,} \\ \sum p(y|\phi)p(x) & \text{discrete.} \end{cases}$$

In an estimation problem, we are only interested in the location of the extrema in the probability density function but not their actual values, so the value of c can be ignored. In addition, in many practical cases it is not possible to evaluate c analytically. Therefore, it is sufficient to express the posterior distribution $p(x|y)$

$$p(x|y) \propto p(y|x)p(x). \tag{1.27}$$

The value of the parameter that maximizes the posterior distribution $p(x|y)$ is referred to as the *Maximum A Posteriori* estimate (MAP), that is,

$$\widehat{x}_{MAP} = \arg\max_{x \in X} p(x|y) = \arg\max_{x \in X} p(y|x)p(x). \tag{1.28}$$

Note that the ML estimator is a special case of the Bayesian MAP estimator with complete prior ignorance. Moreover, other statistical inferences, including the mean, confidence interval, mode, etc, can now be obtained from the posterior distribution, which was not possible in the maximum-likelihood framework. Furthermore, the Bayesian approach admits elimination of nuisance parameters by marginalization.

1.3.2 Prior distributions

When observations from previous experiments are available, the posterior distribution of the parameters for these experiments could be used as a prior distribution for the current experiment. For example, some physical characteristics of the system might preclude some values of the parameters.

The notion of prior distributions has been a point of debate for decades between rigorous statisticians adhering to the classical philosophy, where the maximum-likelihood estimates are obtained with more or less well defined risk functions (e.g. mean-square or uniform), and the Bayesian analysts, that use prior distributions to include prior knowledge.

This debate seems to be fading away, now that the impact of the prior distributions on the end results is better understood. Also, when prior distributions are used, more inference can be obtained from the posterior distribution than from the maximum-likelihood estimate alone. For example, evaluation of confidence intervals of the MAP estimate, and marginalization of the posterior distribution for a parameter of interest are only possible in the Bayesian context.

1.3.2.1 Proper and improper priors

As critics often argue for statistical inference, including MAP estimation, confidence intervals, etc., to exist, the prior distributions $p(x)$ must be proper, i.e., summable or integrable to unity. Even though improper priors might be useful in certain applications, MCMC methods are more sensitive and require a proper prior distribution.

1.3.2.2 Non-informative prior

A noninformative prior is a function which is used in place of a subjective prior distribution when little or no prior information is available. The term "noninformative" is used to connote a lack of subjective belief used in formulating such a prior. However, one can think of a noninformative prior as simply being a function that is formally used in place of a subjective prior distribution, for the purpose of accomplishing some goal. In this case, such

a prior distribution must be non-discriminating with respect to the likelihood function, to "let the data talk" (Box and Tiao, 1973).

1.3.2.3 Jeffrey's prior

Unlike other rules based on *the principle of insufficient reason* that the priors are variant to transformation, Jeffrey's prior (Jeffreys, 1961) is a method of generating noninformative priors which are invariant to transformations of the parameter vector. This method considers the prior distribution for a set of parameters to be proportional to the square root of the determinant of the Fisher information matrix (Box and Tiao, 1973). This approach proposes a prior distribution that is *data translated*, and is given as

$$p(x) \propto |\mathcal{J}(x)|^{1/2},$$

where

$$\mathcal{J}(x) = -\mathrm{E}_{y|x}\left[\frac{\partial^2 \log p(y|x)}{\partial x^2}\right].$$

This is the general form of the non-informative prior distribution.

1.3.3 Ockham's razor

Ockham's Razor is the principle proposed by William of Ockham in the fourteenth century – "Pluralitas non est ponenda sine neccesitate," which translates as "entities should not be multiplied unnecessarily." That is, a simpler model is always selected, as a compromise of complexity and data fit, when the model order is to be jointly estimated along with other parameters.

In the Bayesian framework, this principle favours models of lower dimensions and models of higher complexity are penalized because of the influence of the prior distribution on the model. Similar to the behaviour of an information theoretic criterion (such as AIC or MDL), the effect of Ockham's razor penalizes higher-order models by reducing the weight of the prior distributions more quickly as model order increases.

This is the supporting principle explaining the convergence of the MCMC methods toward the correct model order, as we discuss in latter chapters.

1.4 Traditional tracking

The methods discussed earlier were developed with the assumption that the parameters of interest are static within the observation period. In cases when the static assumption is violated, the parameters evolve over time according to a dynamic system, and conventional *offline* methods are no longer suitable to provide a recursive estimation of these parameters. Therefore, *online* approaches that can recursively estimate the time-varying parameters, given the latest observation, are of interest.

In the field of array signal processing, online tracking algorithms intend to achieve two goals: tracking the motion and locating the incident sources. In this context, the motions and location of the sources can be governed by a dynamic system, described by a *state-space* model, given as follows

$$\boldsymbol{x}(n+1) = \boldsymbol{F}(n, \boldsymbol{x}(n)) + \boldsymbol{v}(n), \tag{1.29}$$

where $\boldsymbol{x}(n) \in \mathcal{C}^{k_o}$ is the state vector and $\boldsymbol{v}(n) \in \mathcal{C}^{k_o}$ is the process noise, defined as zero mean white-noise processes whose covariance matrices is $\boldsymbol{\Sigma}_v$. The term $\boldsymbol{F}(n, \boldsymbol{x}(n))$, which is a time-variant state transistion function relating the state of system from time n to $n+1$, can be a linear or nonlinear function. The observation equation is given as follows

$$\boldsymbol{y}(n) = \boldsymbol{C}(n, \boldsymbol{x}(n)) + \boldsymbol{w}(n), \tag{1.30}$$

where $\boldsymbol{y}(n) \in \mathcal{C}^M$ is an observation vector and $\boldsymbol{w}(n) \in \mathcal{C}^M$, is the observation noise, defined as a zero-mean process whose covariance matrix is $\boldsymbol{\Sigma}_w$. The term $\boldsymbol{C}(n, \boldsymbol{x}(n))$ is a time-varying measurement function that can be linear or nonlinear.

In the field of array signal processing, there have been a variety of techniques to solve this recursive estimation problem. In particular, when the state transition function and the measurement function are both linear with the assumption that both process and observation noises are independent and white Gaussian, the Kalman filter provides the optimal solution to the estimation problem. If such linearity condition is violated, other suboptimal approaches will be adopted.

1.4.1 Kalman filtering approach

Kalman Filter When the state-transition function and the measurement function are both linear, the Kalman filter (KF) can be applied to the tracking problem, where the state and the observation equations can be rewritten as follows

$$\boldsymbol{x}(n+1) = \boldsymbol{F}(n+1,n)\boldsymbol{x}(n) + \boldsymbol{v}(n),$$
$$\boldsymbol{y}(n) = \boldsymbol{C}(n)\boldsymbol{x}(n) + \boldsymbol{w}(n),$$

where $\boldsymbol{F}(n+1,n)$ and $\boldsymbol{C}(n)$ are the known state-transition and measurement matrices, respectively. For a linear system in Gaussian and white noise, the KF is the optimal minimum mean-squared error (MMSE) state estimator and its implementation is well suited for a digital computer.

A key property of the KF is that it leads to minimization of the trace of the filtered state error correlation matrix $\boldsymbol{K}(n)$, defined as

$$\boldsymbol{K}(n) = E[(\boldsymbol{x}(n) - \widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_n))(\boldsymbol{x}(n) - \widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_n))^H],$$

where $\widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_n)$ is the predicted estimate of the state vector. This, in turn, means that the KF is the linear minimum variance estimator of the state vector $\boldsymbol{x}(n)$.

When the number of sources k_o, $\boldsymbol{F}(n+1,n)$, $\boldsymbol{C}(n)$, $\boldsymbol{\Sigma}_v$ and $\boldsymbol{\Sigma}_w$ are given, the KF can provide estimates on $\widehat{\boldsymbol{x}}(n)$ recursively, for $M > k_0$, as follows

Kalman Filter

- Input vector process

  Observations: $\boldsymbol{\mathcal{Y}}_n = [\boldsymbol{y}(1), \boldsymbol{y}(2), ..., \boldsymbol{y}(n)]$

- Description of parameters

  $\widehat{\boldsymbol{x}}(n+1|\boldsymbol{\mathcal{Y}}_n)$ – Predicted estimate of the state vector at time $n+1$, given $\boldsymbol{\mathcal{Y}}_n$

  $\widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_n)$ – Filtered estimate of the state vector at time n, given $\boldsymbol{\mathcal{Y}}_n$

  $\boldsymbol{G}(n)$ – Kalman gain at time n

  $\boldsymbol{\alpha}(n)$ – Innovations vector at time n

  $\boldsymbol{K}(n+1, n)$ – Correlation matrix of the error in $\widehat{\boldsymbol{x}}(n+1|\boldsymbol{\mathcal{Y}}_n)$

  $\boldsymbol{K}(n)$ – Correlation matrix of the error in $\widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_n)$

- Initialization

  1. The initial predicted state estimate can be initialized as $\widehat{\boldsymbol{x}}(1|\boldsymbol{\mathcal{Y}}_0) = E[\boldsymbol{x}(1)]$.
  2. The correlation matrix can be initialized as $\boldsymbol{K}(1,0) = E[(\boldsymbol{x}(1) - E[\boldsymbol{x}(1)])(\boldsymbol{x}(1) - E[\boldsymbol{x}(1)])^H]$.

  In case when $E[\boldsymbol{x}(1)] = \boldsymbol{0}$, we have

  $$\widehat{\boldsymbol{x}}(1|\boldsymbol{\mathcal{Y}}_0) = \boldsymbol{0}, \quad \boldsymbol{K}(1,0) = E[\boldsymbol{x}(1)\boldsymbol{x}^H(1)].$$

- Computation: For $t = 1, 2, 3, ...$

$$\begin{aligned}
\boldsymbol{G}(n) &= \boldsymbol{F}(n+1, n)\boldsymbol{K}(n, n-1)\boldsymbol{C}^H(n)\left[\boldsymbol{C}(n)\boldsymbol{K}(n, n-1)\boldsymbol{C}^H(n) + \boldsymbol{\Sigma}_w\right]^{-1} \\
\boldsymbol{\alpha}(n) &= \boldsymbol{y}(n) - \boldsymbol{C}(n)\widehat{\boldsymbol{x}}(n) \\
\widehat{\boldsymbol{x}}(n+1|\boldsymbol{\mathcal{Y}}_n) &= \boldsymbol{F}(n+1, n)\widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_{n-1}) + \boldsymbol{G}(n)\boldsymbol{\alpha}(n) \\
\boldsymbol{K}(n) &= \boldsymbol{K}(n, n-1) - \boldsymbol{F}(n, n+1)\boldsymbol{G}(n)\boldsymbol{C}(n)\boldsymbol{K}(n, n-1) \\
\boldsymbol{K}(n+1, n) &= \boldsymbol{F}(n+1, n)\boldsymbol{K}(n)\boldsymbol{F}^H(n+1, n) + \boldsymbol{\Sigma}_v
\end{aligned}$$

■

The KF has computational complexity of $\mathcal{O}(M^3)$ (Li *et al.*, 2001).

Extended Kalman Filter The KF approach is optimum in the linear model case. In the nonlinear case, a linearization procedure on the nonlinear system is taken to extend the use of Kalman filtering. The resulting suboptimal filter is known as the extended Kalman filter (EKF).

The linearization can be done by a first-order Taylor approximation on the nonlinear functionals $\boldsymbol{F}(n, \boldsymbol{x}(n))$ and $\boldsymbol{C}(n, \boldsymbol{x}(n))$ around $\widehat{\boldsymbol{x}}(n|\mathcal{Y}_n)$ and $\widehat{\boldsymbol{x}}(n|\mathcal{Y}_n)$, respectively, as follows

$$\begin{aligned}\boldsymbol{F}(n, \boldsymbol{x}(n)) &\approx \boldsymbol{F}(n, \widehat{\boldsymbol{x}}(n|\mathcal{Y}_n)) + \boldsymbol{F}(n+1, n)\left[\boldsymbol{x}(n) - \widehat{\boldsymbol{x}}(n|\mathcal{Y}_n)\right], \\ \boldsymbol{C}(n, \boldsymbol{x}(n)) &\approx \boldsymbol{C}(n, \widehat{\boldsymbol{x}}(n|\mathcal{Y}_{n-1})) + \boldsymbol{C}(n)\left[\boldsymbol{x}(n) - \widehat{\boldsymbol{x}}(n|\mathcal{Y}_{n-1})\right],\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{F}(n+1, n) &= \frac{\partial \boldsymbol{F}(n, \boldsymbol{x})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\widehat{\boldsymbol{x}}(n|\mathcal{Y}_n)}, \\ \boldsymbol{C}(n) &= \frac{\partial \boldsymbol{C}(n, \boldsymbol{x})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\widehat{\boldsymbol{x}}(n|\mathcal{Y}_{n-1})}.\end{aligned}$$

As a result, two sets of transformed linear equations can be expressed as follows

$$\begin{aligned}\boldsymbol{x}(n+1) &= \boldsymbol{F}(n+1, n)\boldsymbol{x}(n) + \boldsymbol{v}(n) + \boldsymbol{r}(n), \\ \bar{\boldsymbol{y}}(n) &= \boldsymbol{C}(n)\boldsymbol{x}(n) + \boldsymbol{w}(n),\end{aligned}$$

where the quantities $\boldsymbol{r}(n)$ and $\bar{\boldsymbol{y}}(n)$ are given as follows, respectively,

$$\begin{aligned}\boldsymbol{r}(n) &= \boldsymbol{F}(n, \widehat{\boldsymbol{x}}(n|\mathcal{Y}_n)) - \boldsymbol{F}(n+1, n)\widehat{\boldsymbol{x}}(n|\mathcal{Y}_n), \\ \bar{\boldsymbol{y}}(n) &= \boldsymbol{y}(n) - [\boldsymbol{C}(n, \widehat{\boldsymbol{x}}(n|\mathcal{Y}_{n-1})) - \boldsymbol{C}(n)\widehat{\boldsymbol{x}}(n|\mathcal{Y}_{n-1})].\end{aligned}$$

One can realize that after the nonlinear system has been linearized, the KF approach can be applied to the transformed equations as if the problem were linear.

At every iteration, the linearized matrices $\boldsymbol{F}(n+1, n)$ and $\boldsymbol{C}(n)$ are computed. Given this information and the noise covariance matrices $\boldsymbol{\Sigma}_v$ and $\boldsymbol{\Sigma}_w$, the EKF algorithm is summarized as follows

Extended Kalman Filter

- Input vector process

  Observations: $\boldsymbol{\mathcal{Y}}_n = [\boldsymbol{y}(1), \boldsymbol{y}(2), ..., \boldsymbol{y}(n)]$

- Initialization

  1. The initial predicted state estimate can be initialized as $\widehat{\boldsymbol{x}}(1|\boldsymbol{\mathcal{Y}}_0) = E[\boldsymbol{x}(1)]$.
  2. The correlation matrix can be initialized as $\boldsymbol{K}(1,0) = E[(\boldsymbol{x}(1) - E[\boldsymbol{x}(1)])(\boldsymbol{x}(1) - E[\boldsymbol{x}(1)])^H]$.

  In case when $E[\boldsymbol{x}(1)] = \boldsymbol{0}$, we have

$$\widehat{\boldsymbol{x}}(1|\boldsymbol{\mathcal{Y}}_0) = \boldsymbol{0},$$
$$\boldsymbol{K}(1,0) = E[\boldsymbol{x}(1)\boldsymbol{x}^H(1)].$$

- Computation: For $t = 1, 2, 3, ...$

$$\begin{aligned}
\boldsymbol{G}(n) &= \boldsymbol{K}(n, n-1)\boldsymbol{C}^H(n)\left[\boldsymbol{C}(n)\boldsymbol{K}(n, n-1)\boldsymbol{C}^H(n) + \boldsymbol{\Sigma}_w\right]^{-1} \\
\boldsymbol{\alpha}(n) &= \boldsymbol{y}(n) - \boldsymbol{C}(n, \widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_{n-1})) \\
\widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_n) &= \widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_{n-1}) + \boldsymbol{G}(n)\boldsymbol{\alpha}(n) \\
\widehat{\boldsymbol{x}}(n+1|\boldsymbol{\mathcal{Y}}_n) &= \boldsymbol{F}(n, \widehat{\boldsymbol{x}}(n|\boldsymbol{\mathcal{Y}}_{n-1})) \\
\boldsymbol{K}(n) &= [\boldsymbol{I} - \boldsymbol{G}(n)\boldsymbol{C}(n)]\,\boldsymbol{K}(n, n-1) \\
\boldsymbol{K}(n+1, n) &= \boldsymbol{F}(n+1, n)\boldsymbol{K}(n)\boldsymbol{F}^H(n+1, n) + \boldsymbol{\Sigma}_v
\end{aligned}$$

■

Unlike the KF approach, the EKF approach is suboptimal, because of the nonlinearity of the system and the linearization step taken in the transformation. The linearization procedure in the state equations and/or the observation equation has the potential to introduce unmodeled errors that violate some basic assumptions in the KF approach. It is also found

that the EKF also has a tendency to diverge during the course of estimation. Indeed, there is no guarantee that a higher order linearization can compensate for errors introduced in the estimation process.

In the context of DOA tracking, the state and the observation equations become

$$\boldsymbol{\phi}(n+1) = \boldsymbol{F}(n+1, \boldsymbol{\phi}(n)) + \boldsymbol{v}(n),$$
$$\boldsymbol{y}(n) = \boldsymbol{S}(\boldsymbol{\phi}(n))\boldsymbol{a}(n) + \boldsymbol{w}(n),$$

where $\boldsymbol{S}(\boldsymbol{\phi}(n))$ is the steering matrix defined as before, with the exception that the nonstatic $\boldsymbol{\phi}(n)$ replaces the static $\boldsymbol{\phi}$. The state-transition function $\boldsymbol{F}(n+1, \boldsymbol{\phi}(n))$ determines the motion of the sources, and once that is given the state $\boldsymbol{\phi}(n)$ can be tracked using the EKF approach described earlier as new observations $\boldsymbol{y}(n)$ arrive.

1.4.2 Linear polynomial approximation beamformer

The linear polynomial approximation (LPA) beamformer (Katkovnik and Gershman, 2002, 2000) is a two dimensional optimization method that tracks the motions of the sources, modelled by a first-order approximation, so that beamformer can track their desired look direction. Using the first-order Taylor series expansion, the evolution of the DOAs can be expressed as

$$\boldsymbol{\phi}(n + iT_s) \approx \boldsymbol{\phi}(n) + \boldsymbol{\phi}'(n) iT_s,$$

where $\boldsymbol{\phi}'(n)$ is the velocity of the DOAs and T_s is the sampling interval. If the length of the observation window L is sufficiently long, the approximation in $\boldsymbol{\phi}(n + kT_s)$ can be replaced by the equals sign. Denote the parameter vector by $\boldsymbol{\theta} = [\boldsymbol{\phi}(n), \boldsymbol{\phi}'(n)]$. The objective of the LPA beamformer is to estimate $\boldsymbol{\theta}$ using the weighted least squares approach.

Consider the single source case. The estimation of $\boldsymbol{\theta}_k$ for $k \in [0, k_o - 1]$ is identical to the simple model fit, defined as follows

$$\boldsymbol{\theta}_k = \arg \min_{\boldsymbol{\theta}_k \in \boldsymbol{\Theta}} \sum_i w_h(iT_s) ||\boldsymbol{y}(n + iT_s) - \boldsymbol{s}(\boldsymbol{\theta}_k) a_k(n + iT_s)||^2,$$

where $w_h(iT_s)$ is the window function, $a_k(n)$ is the amplitude of the kth signal, and $\boldsymbol{s}(\boldsymbol{\theta}_k)$ is the steering vector for the kth signal. Since this objective function is a linear function, it is straight forward to extend this optimization problem from a single source to a multiple sources by a direct superposition of all k_o sources.

The LPA beamformer approach is a two-dimensional search over the axes of the parameter vector $\boldsymbol{\theta}_k$. One axis corresponds to the angular domain that one will search in the conventional beamformer, and the other one corresponds to the angular velocity domain where the velocity is being tracked. Therefore, one can realize that to sustain high accuracy in the estimation of $\boldsymbol{\theta}_k$, the number of points in these two domains must be large, which adds to the computational burden of the algorithm. Furthermore, this method is sensitive to the model selection, i.e., the first-order approximation in $\boldsymbol{\phi}(n+iT_s)$, as well as the length of the window, where the parameters are assumed static within this window. As a result, there is a trade-off between a quick response to the moving sources and accuracy in the estimation.

1.5 Scope of the thesis

This thesis focuses on new approaches to various classical problems in array signal processing, using modern numerical Bayesian methods. We show how the *Markov Chain Monte Carlo* (MCMC) and the *Sequential Monte Carlo* (SMC) methods present new outlooks and offer many advantages to problems in the field. Most conventional array signal processing approaches require the knowledge of model order to estimate other parameters. As such knowledge either is assumed known or has been previously obtained separately. In reality model order is amongst the parameters of interest and should therefore be estimated jointly. Very few methods tackle this joint problem of detection of the model order and estimation of the parameters, particularly in wideband scenario when the techniques require a huge amount of data and demand substantial processing efforts.

In this thesis, we present a novel data model to represent wideband signals that bears a similar data structure to the conventional narrowband data model, after appropriate

transformation. The MCMC and the SMC methods are then applied to this model to solve the joint detection and estimation problem in the array signal processing context. In addition, using SMC techniques, we also present an advanced approach to beamforming for narrowband signals.

Four problems were addressed in the course of this thesis:

1. A new data structure based on interpolation functions and signal samples to approximate wideband signals was developed. This data model, after appropriate transformation, has similar features found in the conventional narrowband data model. Furthermore, as the novel data model is developed for the wideband scenario, it can also address the narrowband scenario without change of structure or parameters. This novel data model is the basis on which the MCMC and the SMC approaches solve the array signal processing problems that are developed in the subsequent chapters.

2. The first algorithm presents an advanced approach using sequential MC methods to beamforming for narrowband signals in white noise with unknown variance. Traditionally, beamforming techniques assume that the number of sources is given and the signal of interest or target is static within an observation period. However, in reality these two assumptions are commonly violated. The former assumption can be dealt with by jointly estimating the number of sources, whereas the latter severely limits the usefulness of conventional beamforming techniques when the target is indeed moving. In the case when the sources are moving, tracking the incident angles of the sources is required, and the accuracy of such tracking significantly affects the performance of signal separation and recovery, which is the objective of beamforming. The proposed method is capable of recursively estimating the time-varying number of sources as well as incident angles of the sources as new data arrive such that the signal amplitudes can be separated and restored in an on-line fashion.

3. The second algorithm presents an application of MCMC methods for the joint detection and estimation problem for the wideband scenario in white noise with unknown variance. Compared to the narrowband scenario, it is more difficult and cumbersome

to solve this array signal processing problem in the wideband context, because more data and computational efforts are required. Conventional approaches tend to solve this problem in the frequency domain, and as such require a considerable amount of data to sustain accuracy, which imposes a huge computational burden for these approaches. Furthermore, these approaches employ separate algorithms like AIC and MDL to estimate the number of sources. In contrast, the proposed method utilizes the reversible jump MCMC technique that simultaneously detects the number of sources and estimates the parameter of interest within the same algorithm. The proposed method is applied to the novel data model mentioned earlier and solves the problem in the time domain. This approach significantly reduces the requirement for the large quantity of data.

4. The final algorithm is an extension of the off-line method of point 3 above to online wideband array signal processing, using sequential MC or particle filtering methods. Most conventional array signal processing approaches are developed under the assumption that the sources are static. If this assumption is invalid, the solutions from these approches become suboptimal and are significantly degraded. When the sources are nonstatic, tracking the motions of the sources is needed, but in wideband scenario this problem becomes more difficult than in narrowband scenario. This algorithm is applied to the modified novel data structure mentioned earlier in white noise with unknown variance for recursive estimation of the motions of the sources as new data arrive. A rational statistical testing procedure is used to keep track of the number of sources.

Even though these numerical Bayesian approaches suffer from some practical difficulties and are very computationally intense, they have the following merits:

- *Flexibility* – Once the samples are available, the integration and optimization objectives are very easily achieved.

- *Convergence* – In optimization, the probability of convergence to the global optimum

is high.

- *Adaptive* – For the sequential MC algorithm, there is no requirement of quasi-stationarity.
- *Performance* – They inherit the properties of Bayesian methods, which generally satisfy the Cramér-Rao lower bound with equality for static scenario and the posterior Cramér-Rao bound for dynamic scenario.

1.6 Outline of thesis

The thesis is divided into six main chapters.

- Chapter 1 is intended as an introduction of the knowledge required to understand the subsequent chapters and to help put the work in perspective.

- Chapter 2 introduces the fundamental theory supporting Markov Chain Monte Carlo methods and the sampling theory supporting particle filters.

- Chapter 3 presents a novel data model that can accomodate both narrowband and wideband signals. The signals are estimated by an interpolation function and their past signal samples. Since this model has a structure similar to the traditional model for narrowband signals, existing methods, like the Markov chain Monte Carlo methods and the sequential Monte Carlo methods, can then be applied to this model to solve the joint detection and estimation problems in the batch and on-line modes. The data model for static signals will first be derived, and a slight modification of this model to address nonstatic signals follows.

- Chapter 4 presents an application of the sequential Monte Carlo methods as an advanced beamforming method that can recursively estimate the nonstatic DOAs of an unknown number of narrowband signals and restore their respective amplitudes. This algorithm could be applied in a cellular communication scenario that utilizes smart antenna technology, where users can come in and out of the cell, to track the number of users and their angular locations.

- Chapter 5 presents a novel method to jointly detect an unknown number of wideband signals and estimate their inter-sensor delays (ISDs), which leads to DOA estimates of the sources. This method, which uses the novel data model presented in Chapter 3, is based on the Markov chain Monte Carlo method. Computer simulations show that the method can consistently estimate the ISDs within the 95% confidence interval and satisfy the Cramér-Rao lower bound. When compared with other competing methods

which are frequency-domain based, the proposed method requires substantially fewer observations.

- Chapter 6 presents sequential MC methods for recursively estimating the ISDs of an unknown number of wideband sources. This method extends the approach in Chapter 5 from static to nonstatic sources and applies sequential MC methods to the modified novel data model for sequential estimation of the time-varying ISDs as new data arrive. Statistical testing procedures are employed to estimate the number of sources. In addition, a posterior Cramér-Rao bound for discrete processes is utilized to demonstrate the consistency of the approach in sequential estimation.

- Finally, the conclusion and future work sections and appendices complete this thesis.

Chapter 2

Review of Monte Carlo Methods

In this chapter, we review the Monte Carlo methods, including the basic theories of Markov Chain Monte Carlo and different sampling procedures, like Metropolis-Hastings algorithm, Gibbs sampler, and Importance Sampling. Furthermore, techniques for sequential Monte Carlo methods for online tracking are described.

2.1 Introduction

Markov chain Monte Carlo (MCMC) methodology provides enormous scope for realistic statistical modelling. MCMC is essentially Monte Carlo numerical integration using Markov chains. With the Bayesian approach, it is common to integrate the posterior distribution over undesired model parameters, to make inference about other desired model parameters or to make predictions, given the data. Often, it is also necessary to evaluate expectations, a process which also requires integration. As most realistic data models lead to multivariate and highly non-linear posterior distributions that are not analytically workable, numerical methods present attractive alternatives. Monte Carlo integration draws samples from the required distribution by running a cleverly constructed Markov chain, and then forms sample averages to approximate expectations.

2.1.1 Computation of expectations

Monte Carlo sampling methods can be used generate a numerical approximation in the form of a histogram corresponding to an arbitrary distribution of interest. This allows for easy numerical integration, marginalization, and computation of other statistical inferences.

To illustrate the use of this approach, consider a function $f(\boldsymbol{x})$ of $\boldsymbol{x}$. The expected value of the function over a probability distribution function $p(\boldsymbol{x})$ is defined as

$$I_f = E_{p(\boldsymbol{x})}(f(\boldsymbol{x})) = \int p(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.1}$$

This expectation may be difficult or impossible to evaluate analytically. However, assuming that a large number $N \gg 1$ of samples $\left\{\boldsymbol{x}^{(i)}; i = 1, 2, \ldots, N\right\}$ distributed according to $p(\boldsymbol{x})$ are available, the Monte Carlo numerical approximation of the distribution $p(\boldsymbol{x})$ is given by

$$\hat{p}_N(d\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{x}^{(i)}}(d\boldsymbol{x}), \tag{2.2}$$

where $d\boldsymbol{x}$ is a small, finite region surrounding an $\boldsymbol{x}$ of interest and $\delta_{\boldsymbol{x}^{(i)}}$ is an indicator function defined as

$$\delta_{\boldsymbol{x}^{(i)}}(d\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x}^{(i)} \in d\boldsymbol{x}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

As a result, we can approximate the expected value of $f(\boldsymbol{x})$ as follows

$$I_f \approx \hat{I}_{f,N} = \int \hat{p}_N(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} = \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}^{(i)}). \tag{2.4}$$

If the samples $\boldsymbol{x}^{(i)}$ are statistically independent, according to the strong law of large numbers (SLLN) with $N \to +\infty$, the estimates $\hat{I}_{f,N}$ converges to I_f, i.e.,

$$\lim_{N \to \infty} \hat{I}_{f,N} = I_f, \tag{2.5}$$

where such convergence allows the existence of central limit theorems for $\hat{I}_{f,N}$ in the following form

$$\lim_{N \to +\infty} \sqrt{N}(I_{f,N} - I_f) \overset{\sim}{\to} \mathcal{N}(0, \sigma^2), \tag{2.6}$$

where $\sigma^2 < \infty$ and $\overset{\sim}{\rightarrow}$ denotes convergence in distribution.

The advantage of the Monte Carlo sampling method is now clear. One can easily and efficiently estimate I_f and other statistical inferences based on the set of samples $\left\{\boldsymbol{x}^{(i)}; i = 1, 2, \ldots, N\right\}$. In particular, if $p(\boldsymbol{x}) = \pi(\boldsymbol{x})$, which is the posterior distribution function of interest, parameter estimates can be determined by numerical evaluation of the expectation of the posterior distribution, a process which is easily implemented. *Maximum a posteriori* (MAP) estimates are generated by finding the maximum of the histogram. This is implemented by finding the largest element of an array, a process which is much easier than a multi–dimensional search, required by conventional Bayesian methods. With Monte Carlo sampling techniques, it is also straightforward to generate estimates of variances, or estimates of the confidence regions corresponding to parameter estimates.

Generally, it is not possible to draw samples directly from any arbitrary probability density function. In some cases, for standard distributions such as Gaussian, or uniform, many techniques exist to perform this task. Many other methods exist for drawing samples from standard distributions (Ripley, 1987). However, in practical cases of interest, these methods are found to be unsuitable for arbitrary nonstandard distributions.

The challenge is to generate *iid* samples, from any arbitrary multi-variate non-standard probability density function. We now present a variety of methods for accomplishing this task.

2.1.2 Direct sampling

In the event that the constant of proportionality is known, or can at least be evaluated approximately, one classic algorithm is the Accept-Reject procedure. This algorithm assumes that

$$\pi(\boldsymbol{x}) \leq Mq(\boldsymbol{x}) \quad M < \infty \quad \forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}},$$

where $q(\boldsymbol{x})$ is some candidate distribution on the space $\boldsymbol{\mathcal{X}}$ that is easy to sample from, and M is the "blanket" factor. The procedure is described below (Andrieu *et al.*, 1998).

Accept-Reject Algorithm

1. Sample $\boldsymbol{x} \sim q(\boldsymbol{x})$ and $u \sim U_{[0,1]}$.
2. if $u < \frac{\pi(\boldsymbol{x})}{Mq(\boldsymbol{x})}$ then accept $\boldsymbol{x}$, otherwise return to step 1.

■

The set of returned $\boldsymbol{x}$ is then distributed according to $\pi(\boldsymbol{x})$. This procedure suffers from one major drawback: the "blanketing" factor M needs to be estimated, which is not always possible. Also, the probability of accepting a proposed $\boldsymbol{x}$ is less than or equal to $\frac{1}{M}$, making the algorithm inefficient in practice.

2.1.3 Importance sampling

In general, we may not be able to sample variates directly from the distribution of interest because of the difficulties involved in generating random samples with a given distribution. Moreover, in some scenarios, the normalizing constant of the distribution of interest is unknown or cannot be estimated. Then an importance sampling approach (Rubin, 1988) can be adopted.

Denote an *importance function* by $q(\boldsymbol{x})$, which must have the same support as $\pi(\boldsymbol{x})$ and must be easy to draw a large number $N \gg 1$ of statistically independent samples. The histogram of these samples approximates the distribution $q(\boldsymbol{x})$. The *Bayesian importance sampling* approach is then used to generate a numerical approximation to the desired distribution $\pi(\boldsymbol{x})$ by a set of "importance weights" $w(\boldsymbol{x}^{(i)})$ given by

$$w(\boldsymbol{x}^{(i)}) \propto \frac{\pi(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})}, \quad i = 1, ..., N. \tag{2.7}$$

In the event that the desired distribution $\pi(\boldsymbol{x})$ is completely known, the weights would already be normalized.

The Monte Carlo approximations defined in (2.2) and (2.4) can be redefined as

$$\hat{\pi}_N(d\boldsymbol{x}) = \frac{\sum_{i=1}^{N} w(\boldsymbol{x}^{(i)})\delta_{\boldsymbol{x}^{(i)}}(d\boldsymbol{x})}{\sum_{i=1}^{N} w(\boldsymbol{x}^{(i)})}, \tag{2.8}$$

$$= \sum_{i=1}^{N} \tilde{w}(\boldsymbol{x}^{(i)})\delta_{\boldsymbol{x}^{(i)}}(d\boldsymbol{x}), \tag{2.9}$$

where $\tilde{w}(\boldsymbol{x}^{(i)})$ is the normalized importance weight, given by

$$\tilde{w}(\boldsymbol{x}^{(i)}) = \frac{w(\boldsymbol{x}^{(i)})}{\sum_{j=1}^{N} w(\boldsymbol{x}^{(j)})}. \tag{2.10}$$

It therefore follows that

$$\hat{I}_{f,N} = \frac{1}{N}\sum_{i=1}^{N} \tilde{w}(\boldsymbol{x}^{(i)}) f(\boldsymbol{x}^{(i)}). \tag{2.11}$$

In most applications, however, it is impossible to evaluate the total probability of $\pi(\boldsymbol{x})$, which remains known up to a constant and thus the normalization of the weights is required, as shown in (2.10). In effect, this algorithm converts samples from the distribution $q(\boldsymbol{x})$ to the desired distribution $\pi(\boldsymbol{x})$.

Using the approximation in (2.9) for $\pi(\boldsymbol{x})$ with a finite number of samples, the estimate from (2.11) is biased. However, the expectation $\hat{I}_{f,N}$ of any function $f(\boldsymbol{x})$ over $\hat{\pi}_N(\boldsymbol{x})$ asymptotically converges to I_f as $N \to +\infty$, i.e.,

$$\lim_{N\to\infty} \hat{I}_{f,N} = I_f. \tag{2.12}$$

While the importance sampling algorithm is easy to implement and in theory almost any importance sampler can be used, the convergence for the estimate of the integral can be inefficient. The rate of convergence is highly dependent on how closely the importance function $q(\boldsymbol{x})$ resembles the distribution function $\pi(\boldsymbol{x})$ of interest, whereas the rate does not improve considerably with an increased number of function evaluations.

2.2 Markov Chain Monte Carlo Methods

Another strategy to obtain the samples is to observe the states of a Markov chain whose limiting distribution is our distribution $\pi(\boldsymbol{x})$ of interest. Each state of the chain represents a

bin of the histogram. These algorithms have nice properties, such as guaranteed convergence and insensitivity to the initial values, which, in some circumstances, might outweigh the computational burden, that can be significant in some cases.

The supporting theory is summarized in the following sections. The reader is referred to Ruanaidh and Fitzgerald (1996) for a good introductory presentation and to Gilks *et al.* (1998), Robert and Casella (1999), and Gamerman (1997) for a more complete treatment.

2.2.1 Background on Markov chains

A Markov chain is a discrete time stochastic process described in terms of states. Denote the set of possible state values that the process $\boldsymbol{x}_i$ can take by $A = \{S_0, S_1, ..., S_q\}$. When $\boldsymbol{x}_i = S_k$, the Markov chain is said to be in state k.

For a first-order Markov process, the probability of the next state $\boldsymbol{x}_{i+1}$, given all previous values of the process, depends only on the present state $\boldsymbol{x}_i$, i.e.,

$$P[\boldsymbol{x}_{i+1} \in A|\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_i] = P[\boldsymbol{x}_{i+1} \in A|\boldsymbol{x}_i], \tag{2.13}$$

where $P[\cdot|\cdot]$ denotes a conditional probability. Let the one-step transition probability from another state be $P_{k|j}[x_{i+1}]$, given as

$$P_{k|j}[\boldsymbol{x}_{i+1}] = P[\boldsymbol{x}_{i+1} = S_k|\boldsymbol{x}_i = S_j]. \tag{2.14}$$

Assuming that the transition probabilities are stationary over time, it is possible to represent the complete set of these probabilities by a transition matrix as follows

$$\boldsymbol{T} = \begin{bmatrix} P_{1|1} & P_{1|2} & \dots & P_{1|q} \\ P_{2|1} & P_{2|2} & \dots & P_{2|q} \\ & \vdots & & \\ P_{q|1} & P_{q|2} & \dots & P_{q|q} \end{bmatrix}, \tag{2.15}$$

such that we can write a general expression for the probability of $\boldsymbol{x}_i$ being in state S_k from state S_j after n iterations as follows

$$P_{k,j}^{(n)}[\boldsymbol{x}_i] = P[\boldsymbol{x}_i = S_k|\boldsymbol{x}_{i-n} = S_j] = \boldsymbol{T}^n P[\boldsymbol{x}_{i-n}|S_j]. \tag{2.16}$$

If the kernel satisfies certain conditions, the Markov chain will converge toward a limiting distribution π, which is independent of the initial state $\boldsymbol{x}_{n_0}$, as follows

$$\pi = \lim_{n\to\infty} P[\boldsymbol{x}_i] = \lim_{n\to\infty} \boldsymbol{T}^{n-n_0} P[\boldsymbol{x}_{n_0}], \tag{2.17}$$

At that point, the states of the resulting chain are all distributed according to the limiting distribution

$$\boldsymbol{T}\pi = \pi. \tag{2.18}$$

However, how long it takes the chain to reach the equilibrium state depends on a number of factors and, in particular, the number of states that must be discarded at the initial stage, a transient period known as the "burn-in" of the Markov chain. Thus, if the limiting distribution π of a Markov chain is the posterior distribution of interest, the states of the chain become the samples from the distribution.

2.2.2 Properties of Markov chains

Although not all Markov chains have a limiting distribution, many algorithms exist to set up Markov chains that will converge to the desired density function. For these algorithms to perform as intended, some conditions must be satisfied: invariance, reversibility, irreductibility, aperiodicity, and recurrence.

- Invariance – The invariance property means that all states in a Markov chain have reached a limiting distribution and are distributed according to the distribution of (2.18).
- Reversibility – The reversibility means that the probability of a transition of a Markov chain from one state to another is equal to the probability of a transition in the reverse direction. Reversibility is a sufficient condition for the states of the chain to be in their limiting distribution.
- Irreductibility – The irreductibility condition means that from all starting points the Markov chain can reach any non-empty set with positive probability, in some finite number of iterations.

- Aperiodicity – The aperiodicity means that a Markov chain has kernels that do not induce a periodic behaviour in the states.
- Recurrence – The recurrence means that from all starting points all states can be reached infinitely often.

All MCMC algorithms have been designed to satisfy these constraints. Next we will present three of them: the Metropolis-Hastings (M-H) algorithm, the Gibbs sampler, and the M-H one-at-a-time algorithm.

2.2.3 Metropolis-Hastings algorithm

The Metropolis-Hastings (M-H) algorithm (Hastings, 1970) is a very flexible method to provide a random sequence of a sample from a given density. Denote a *candidate function* by $q(\cdot)$ to sample from $\pi(\cdot)$. This candidate function is similar to the importance function previously described, and is chosen to be easy to sample from. One major advantage of this algorithm is that the knowledge of the normalizing constant of the posterior distribution is not required. The posterior distribution is only present in ratios, where this unknown normalizing constant will cancel out, assuming it remains constant.

Assuming that the chain is in state $\boldsymbol{x}$, we can obtain a candidate $\boldsymbol{x}^\star$ for the next state by sampling $q(\cdot)$, which in the general case is conditional on $\boldsymbol{x}$. This candidate will be accepted with probability α defined as

$$\alpha(\boldsymbol{x}^\star, \boldsymbol{x}) = \min\{r(\boldsymbol{x}^\star, \boldsymbol{x}), 1\}, \tag{2.19}$$

with the acceptance ratio r defined as

$$r(\boldsymbol{x}^\star, \boldsymbol{x}) = \frac{\pi(\boldsymbol{x}^\star) q(\boldsymbol{x}|\boldsymbol{x}^\star)}{\pi(\boldsymbol{x}) q(\boldsymbol{x}^\star|\boldsymbol{x})}. \tag{2.20}$$

If the candidate is accepted, the chain takes the new state $\boldsymbol{x}^\star$; otherwise the chain remains at the current state $\boldsymbol{x}$.

Metropolis-Hasting Algorithm

1. Initialization $\boldsymbol{x}^{(0)}$ at iteration 0

2. Iteration i, $i \geq 1$,

   - Sample a candidate $\boldsymbol{x}^{\star} \sim q(\boldsymbol{x}|\boldsymbol{x}^{(i-1)})$.
   - Evaluate the acceptance probability

$$r(\boldsymbol{x}^{\star}, \boldsymbol{x}^{(i-1)}) = \frac{\pi(\boldsymbol{x}^{\star})q(\boldsymbol{x}^{(i-1)}|\boldsymbol{x}^{\star})}{\pi(\boldsymbol{x}^{(i)})q(\boldsymbol{x}^{\star}|\boldsymbol{x}^{(i-1)})}.$$

$$\alpha(\boldsymbol{x}^{\star}, \boldsymbol{x}^{(i-1)}) = \min\{r(\boldsymbol{x}^{\star}, \boldsymbol{x}^{(i-1)}), 1\}$$

   - Sample $u \sim U_{[0,1]}$
   - if $u \leq \alpha(\boldsymbol{x}^{\star}, \boldsymbol{x}^{(i-1)})$
     then the chain takes the state $\boldsymbol{x}^{(i)} = \boldsymbol{x}^{\star}$, otherwise it remains at $\boldsymbol{x}^{(i)} = \boldsymbol{x}^{(i-1)}$.

■

In order to get a well-mixed chain, the candidate function $q(\cdot)$ should allow the chain to explore the entire probability space, but with substantial probability of being accepted. To satisfy the irreductibility and the aperiodicity properties, the M-H algorithm requires that $q(\cdot)$ be continuous and strictly positive on the support of $\pi(\boldsymbol{x})$. It can be shown that the acceptance probability defined in (2.19) with (2.20) guarantees the reversibility requirement.

2.2.4 Gibbs sampler

Being a special case of the M-H algorithm, the Gibbs sampler (Geman and Geman, 1984) allows one to break down the problem of drawing samples from a multivariate density into one of drawing successive samples from densities of smaller dimensionality.

Given a random vector $\boldsymbol{x}$ of length K, the Gibbs sampler samples each parameter, one at the time, according to the full conditional distributions when all the other parameters are

fixed. Let $q(x_k|\boldsymbol{x}_{-k})$, $k = 1, ..., K$ denote the full conditional density of the kth component of the vector $\boldsymbol{x}$, where

$$q(x_k|\boldsymbol{x}_{-k}) = q(x_i|x_1, x_2, ..., x_{k-1}, x_{k+1}, ..., x_K). \tag{2.21}$$

Instead of sampling from a complex K dimensional distribution, the problem is reduced to sampling K times from one dimensional conditional distributions. As soon as a variate is drawn, it is inserted into the full conditional probability density function, and it remains there until the next iteration. For this algorithm to be a viable option, all the full conditional posterior distributions must be available in their analytical form.

Gibbs Sampling Algorithm

1. Initialization $\boldsymbol{x}^{(0)}$ at iteration 0
2. Iteration i, $i \geq 1$,
   - Sample $x_1^{(i)} \sim q(x_1|\boldsymbol{x}_{-1}^{(i)})$
   - Sample $x_2^{(i)} \sim q(x_2|\boldsymbol{x}_{-2}^{(i)})$
   - $\vdots$
   - Sample $x_K^{(i)} \sim q(x_K|\boldsymbol{x}_{-K}^{(i)})$

The rate of convergence of the Gibbs sampler is governed by *the posterior correlations* between the different parameters and the dimensionality of the parameter space. One way to improve the rate of convergence is to jointly sample highly correlated variables by creating partitions. Also, it might be beneficial to randomly vary the order of the components.

2.2.5 Metropolis-Hastings One-at-a-time algorithm

In the case where $\boldsymbol{x}$ is of high dimension, it becomes very difficult to select a good candidate function that would lead to a reasonable acceptance rate and allow the chain to mix. To

address this problem, the *Metropolis-Hasting one-at-the-time* algorithm (Andrieu *et al.*, 1998), in a similar fashion to the Gibbs sampler, samples each component (or partition), conditionally on the other components, using a set of candidate functions. Obviously, this algorithm includes the Gibbs sampler as a special case for which the candidate functions are the full conditional distributions and the candidates are always accepted.

In the M-H one-at-the-time algorithm, only one element $x_k^\star$ for $k = 1, 2, ..., K$ at a time will be sampled from $q(\cdot)$, and this candidate vector $\boldsymbol{x}^\star$, defined as

$$\boldsymbol{x}^\star = [x_1, ..., x_{k-1}, x_k^\star, x_{k+1}, ..., x_K]^T, \tag{2.22}$$

will be accepted as the next state with a probability as in (2.19).

Metropolis-Hasting One-at-the-time Algorithm

1. Initialization $\boldsymbol{x}^{(0)}$ at iteration 0
2. Iteration i, $i \geq 1$
3. Set $\boldsymbol{x}^{(i)} = \boldsymbol{x}^{(i-1)}$, and for $k = 1, ..., K$
   - Sample a candidate $x_k^\star \sim q(x|x^{(i)})$.
   - Set the candidate as $\boldsymbol{x}^\star = [x_1^{(i)}, ..., x_{k-1}^{(i)}, x_k^\star, x_{k+1}^{(i)}, ..., x_K^{(i)}]^T$
   - Evaluate the acceptance probability

$$r(\boldsymbol{x}^\star, \boldsymbol{x}^{(i)}) = \frac{\pi(\boldsymbol{x}^\star) q(\boldsymbol{x}^{(i)}|\boldsymbol{x}^\star)}{\pi(\boldsymbol{x}^{(i)}) q(\boldsymbol{x}^\star|\boldsymbol{x}^{(i)})}.$$

$$\alpha(\boldsymbol{x}^\star, \boldsymbol{x}^{(i)}) = \min\{r(\boldsymbol{x}^\star, \boldsymbol{x}^{(i)}), 1\}$$

   - Sample $u \sim U_{[0,1]}$
   - if $u \leq \alpha(\boldsymbol{x}^\star, \boldsymbol{x}^{(i)})$

     then the chain takes the state $\boldsymbol{x}^{(i)} = \boldsymbol{x}^\star$, otherwise it remains at $\boldsymbol{x}^{(i)}$.

2.2.6 Reversible jump MCMC

This section describes how to build MCMC algorithms for model order detection. The main difficulty for the Markov chain in this case is to be able to jump from one subspace to another while preserving the correct invariant distribution. Green (1995) developed a general framework, known as the reversible jump MCMC, which allows the sampling process to jump between subspaces of different dimensions while preserving the reversibililty condition. As other sampling techniques described earlier in this chapter assume the model order dimension is given or fixed, one could obviously detect the appropriate model order by sampling the subspace corresponding to a range of model orders independently. However, this approach is very inefficient and computationally intensive, since the same effort is allocated to all model orders, even though some models will have a low posterior probability. On the other hand, the reversibile jump MCMC allows for the *joint* detection of the model order and the sampling of the parameters from its posterior distribution. The samples concentrate on models with high posterior probability.

Denote the model order by the discrete variable k, which is considered as one of the parameters of interest, and the whole parameter space by $\bigcup_{k=0}^{k_{max}} k \times \boldsymbol{\Phi_k}$, where $\boldsymbol{\Phi_k}$ is the space of the parameters of the model of order k, and k_{max} is the maximum allowable model order in question. The entire parameter space will be visited by moves designed to preserve the reversibility condition. Similar to the M-H algorithm, at each iteration the reversibile jump MCMC algorithm proposes a candidate from a set of candidate functions that are designed to explore the different subspaces and to change the model order.

Denote a candidate distribution by $q(\cdot)$, and a candidate by $\boldsymbol{x}^\star$ of size $k^\star$. Like the M-H algorithm, at each iteration a candidate $\boldsymbol{x}^\star$ is obtained by sampling $q(\cdot)$ and will be accepted with probability α defined as

$$\alpha((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k)) = \min\{r((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k)), 1\}, \tag{2.23}$$

with the acceptance ratio r defined as

$$r((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k)) = \frac{\pi(\boldsymbol{x}^\star, k^\star) q(\boldsymbol{x}, k)}{\pi(\boldsymbol{x}, k) q(\boldsymbol{x}^\star, k^\star)} \times \boldsymbol{J}((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k)), \tag{2.24}$$

where $\boldsymbol{J}((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k))$ is the Jacobian of the transformation required to reconcile the total probability between spaces of different dimensions so that the reversibility condition is satisfied. As descibed in Godsill (1998), $\boldsymbol{J}((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k))$ is given by

$$\boldsymbol{J}((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k)) = \left| \frac{\partial \boldsymbol{x}^\star}{\partial \boldsymbol{x}} \right|. \tag{2.25}$$

If the candidate $(\boldsymbol{x}^\star, k^\star)$ is accepted, the chain takes the new state; otherwise the chain remains at the current state.

In addition to the requirement of the acceptance probability in (2.23) with (2.24), the candidate functions must be appropriately selected in order to satisfy the reversibility condition. The most widely used candidate functions are the birth/death complementary moves (Andrieu and Doucet, 1999; Andrieu, 1997; Troughton and Godsill, 1997, 1998). When the death move is selected, the algorithm proposes a candidate in the model of lower dimension, as opposed to the birth move for which the algorithm proposes a candidate of higher dimension.

The probabilities for choosing each move are denoted u_k, b_k and d_k, respectively, such that $u_k + b_k + d_k = 1$ for all k. In accordance with Andrieu and Doucet (1999), we choose

$$b_k = c \min\left\{ \frac{p(k+1)}{p(k)}, 1 \right\}, \quad d_{k+1} = c \min\left\{ \frac{p(k)}{p(k+1)}, 1 \right\}, \tag{2.26}$$

where $p(\cdot)$ is the prior distribution of the kth model, and c is a tuning parameter which determines the ratio of update moves to jump moves. We choose $c = 0.5$ so that the probability of a jump is between 0.5 and 1 at each iteration (Green, 1995) . Note that in the case of the birth/death moves, the Jacobian of the transformation in (2.24) is unity (Godsill, 2001). The overall description of the reversible jump MCMC algorithm is determined by the choice of moves at each iteration. This description is summarized as follows.

Reversible Jump MCMC

1. Initialization: set $\mathbf{\Phi}^{(0)} = (\boldsymbol{x}^{(0)}, k^{(0)})$
2. Iteration i,
   - Sample $u \sim U_{[0,1]}$
   - if $(u < b_{k^{(i)}})$ then execute a "birth move" (Section 2.2.6.2),
     - else if $(u < b_{k^{(i)}} + d_{k^{(i)}})$ then execute a "death move" (Section 2.2.6.2),
     - else, execute an update move (Section 2.2.6.1).
3. $i \leftarrow i + 1$, goto step 2

■

2.2.6.1 Update move

When the update move is selected, a candidate of the same dimension is proposed. Here, we assume that the current state of the algorithm is in $\{\mathbf{\Phi}_k, k\}$. When the update move is selected, the algorithm samples only on the space of $\mathbf{\Phi}_k$ for a fixed k. Let $\boldsymbol{x}^\star$ of size k be a candidate drawn from the proposal distribution function $q(\cdot|\cdot)$. The acceptance ratio for an update move is defined using (2.20) as

$$r_{update} = \frac{\pi(\boldsymbol{x}^\star) q(\boldsymbol{x}|\boldsymbol{x}^\star)}{\pi(\boldsymbol{x}) q(\boldsymbol{x}^\star|\boldsymbol{x})}. \tag{2.27}$$

The candidate $\boldsymbol{x}^\star$ is then accepted as the current state with probability

$$\alpha_{update}(\boldsymbol{x}, \boldsymbol{x}^\star) = \min\{1, r_{update}\}. \tag{2.28}$$

2.2.6.2 Birth and death moves

In the birth move, we assume the current state is in $(\boldsymbol{\Phi}_k, \{k\})$ and we wish to determine whether the next state is in $(\boldsymbol{\Phi}_{k+1}, \{k+1\})$. The subscript k now indicates the size of the parameter space. This involves the addition of a element x_c, selected from a candidate distribution, to the existing vector parameter to create

$$\boldsymbol{x}_{k+1}^{\star} = [\boldsymbol{x}_k, x_c]. \tag{2.29}$$

The proposal distribution $q(\boldsymbol{x}_{k+1}^{\star}, k+1|\boldsymbol{x}_k, k)$ for the birth move is therefore

$$q(\boldsymbol{x}_{k+1}^{\star}, k+1|\boldsymbol{x}_k, k) = p(k+1)p(x_c), \tag{2.30}$$

where $p(k+1)$ is the prior distribution function of the model order $k+1$ and $p(x_c)$ is the candidate distribution of an individual parameter x_i. The candidate state $(\boldsymbol{x}_{k+1}^{\star}, k+1)$ is then accepted with probability $\alpha_{birth} = \min\{r_{birth}, 1\}$ with r_{birth} defined by (2.24).

The following block describes the algorithm for the birth move. Suppose that the current state of the Markov chain is $\{k\} \times \boldsymbol{\Phi}_k$, then

---

Birth Move

- Propose a new element x_c and a candidate,

$$\boldsymbol{x}_{k+1}^{(i+1)} = [\boldsymbol{x}_k^{(i)}, x_c]$$

- Evaluate $\alpha_{birth} = \min\{r_{birth}((\boldsymbol{x}^{\star}, k^{\star}), (\boldsymbol{x}, k)), 1\}$ with (2.23).
- Sample $u \sim U_{[0,1]}$.
- if $(u \leq \alpha_{birth})$ then the state of the Markov Chain becomes $(\boldsymbol{x}_{k+1}^{(i+1)}, k+1)$, else it remains at $(\boldsymbol{x}_k^{(i)}, k)$.

In order to maintain the invariant distribution of the reversible jump MCMC algorithm with respect to model order, the Markov chain must be *reversible* with respect to moves across subspaces of different model orders. That is, the probability of moving from model order k to $k+1$ must be equal to that of moving from $k+1$ to k. Therefore we propose a death move in which a source in the current state $(\boldsymbol{x}_{k+1}, k+1)$ is randomly selected to be removed such that the next state becomes $(\boldsymbol{x}_k, k)$ at the next iteration. A sufficient condition for reversibility with respect to model order (Green, 1995) is that the acceptance ratio for the death move be defined as

$$r_{death} = \frac{1}{r_{birth}}, \tag{2.31}$$

and the new candidate of dimension k is accepted with probability as described in (2.24)

$$\alpha_{death} = \min\left\{1, r_{death}\right\}. \tag{2.32}$$

In the death move, we assume the current state is in $(\boldsymbol{\Phi}_{k+1}, \{k+1\})$ and we wish to determine whether the state is in $(\boldsymbol{\Phi}_k, \{k\})$ at the next iteration. This involves the removal of an element of the parameter vector, which is chosen randomly amongst the $(k+1)$ existing elements. The proposal distribution $q(\boldsymbol{x}_k^\star, k|\boldsymbol{x}_{k+1}, k+1)$ for the death move is therefore chosen as

$$q(\boldsymbol{x}_k^\star, k|\boldsymbol{x}_{k+1}, k+1) = p(k) \div \binom{k+1}{1} \propto p(k)\frac{1}{(k+1)}. \tag{2.33}$$

The candidate state $(\boldsymbol{x}_k^\star, k)$ is then accepted with probability α_{death}.

The following block describes the algorithm for the death move. Suppose that the current state of the Markov chain is $\{k+1\} \times \boldsymbol{\Phi}_{k+1}$, then

Death Move

- Select randomly the jth element to form the candidate,

$$\boldsymbol{x}_k^{(i+1)} = [\boldsymbol{x}_{1:(j-1)}^{(i)}, \boldsymbol{x}_{(j+1):(k+1)}^{(i)}]$$

- Evaluate $\alpha_{death} = \min\{r_{death}((\boldsymbol{x}^\star, k^\star), (\boldsymbol{x}, k)), 1\}$ with eq. (2.23).
- Sample $u \sim U_{[0,1]}$.
- if $(u \leq \alpha_{death})$ then the state of the Markov Chain becomes $(\boldsymbol{x}_k^{(i+1)}, k)$, else it remains at $(\boldsymbol{x}_{k+1}^{(i)}, k+1)$.

■

2.3 Sequential Monte Carlo methods

The algorithms presented so far are all off-line methods and they are suitable for situations where the parameters of interest are assumed static throughout the entire observation period and batch processing is used. However, in many practical problems, the parameters of interest as well as their distribution functions are indeed time-varying. Thus one is more interested in online methods, also known as *sequential processing*, which allow recursive estimation of the varying parameters and distribution functions in time, as new data arrive.

As a matter of fact, many problems in applied statistics, statistical signal processing, time series analysis, and econometrics can be posed as a dynamic system where the parameter of interest $\boldsymbol{x}$ evolves in time. The sequence of evolving posterior density functions $\pi(\boldsymbol{x}_n)$ is known as a *probabilistic dynamic system*. This dynamic system usually adopts a state-space model in which a transition equation describes the prior distribution of a hidden Markov process $\{\boldsymbol{x}_n\}$ and an observation equation describes the likelihood of the observations $\{\boldsymbol{y}_n\}$.

Under the Bayesian framework, it is possible to obtain all relevant information about the state $\{\boldsymbol{x}_{1:n}\}$ given the observations $\{\boldsymbol{y}_{1:n}\}$, where the notation $(\cdot)_{1:n}$ indicates all the elements from time 1 to time n, from the posterior distribution $\pi(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$, given by

$$\pi(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) = p(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n | \boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n). \tag{2.34}$$

In particular, one is interested in recursively estimating the marginal distribution $\pi(\boldsymbol{x}_n)$, given by

$$\pi(\boldsymbol{x}_n) = p(\boldsymbol{x}_n | \boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n), \tag{2.35}$$

such that statistical inferences like the posterior mode or mean of the state $\boldsymbol{x}_n$ can be evaluated. However, in only a few exceptional cases where linear Gaussian state-space models are involved, it is difficult, if not impossible, to analytically evaluate these distributions. Therefore, a numerical approach using sequential Monte Carlo methods, also known as *Particle Filters*, is adopted to recursively estimate the time-varying distribution functions by a set of samples (*particles*) distributed according to the $\pi(\boldsymbol{x}_n)$, given the past information and the latest observation.

Such a recursive filter is composed of two stages: prediction and update. The prediction stage uses the state-space model to predict the state posterior distribution function $\pi(\boldsymbol{x}_n)$ from one measurement time to the next. Due to unknown disturbances like state noise, the prediction generally translates, deforms, and spreads the state posterior distribution function. The update operation uses the latest measurement to modify the predicted state posterior distribution function using Bayes theorem, which is the mechanism for updating knowledge about the target state in light of the extra information from new data.

2.3.1 Particle filtering

Denote a set of N particles or samples from time 1 to time t by $\boldsymbol{x}_{1:n}^{(i)}, i = 1, \ldots, N$, distributed according to the joint distribution $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$. Our goal is to update the numerical histogram of the particles to give $p(\boldsymbol{x}_{1:n+1}|\boldsymbol{y}_{1:n+1})$, given the knowledge of $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ and

the new observation $\boldsymbol{y}_{n+1}$, without the expense of resampling all the particles from time 1 to $n+1$.

The joint posterior distribution of all parameters $\boldsymbol{x}$ from time 1 to $n+1$ can be written using Bayes' theorem as

$$\pi(\boldsymbol{x}_{1:n+1}) = \frac{p(\boldsymbol{y}_{1:n+1}|\boldsymbol{x}_{1:n+1})p(\boldsymbol{x}_{1:n+1})}{p(\boldsymbol{y}_{1:n+1})}, \tag{2.36}$$

where the term $p(\boldsymbol{y}_{1:n+1}|\boldsymbol{x}_{1:n+1})$ is referred to as the total likelihood function up to time $n+1$, and the other terms $p(\boldsymbol{x}_{1:n+1})$ and $p(\boldsymbol{y}_{1:n+1})$ are the prior distribution functions of the state $\boldsymbol{x}_{1:n+1}$ and the observations $\boldsymbol{y}_{1:n+1}$, respectively. It can be shown (Doucet, 1998), using the Markov properties of the model and *iid* assumptions on the noise variables, that (2.36) can be written in the recursive, time-update form as (see Appendix B)

$$\pi(\boldsymbol{x}_{1:n+1}) = p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) \times \frac{p(\boldsymbol{y}_{n+1}|\boldsymbol{x}_{n+1})p(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n)}{p(\boldsymbol{y}_{n+1}|\boldsymbol{y}_{1:n})}. \tag{2.37}$$

Even though the form in (2.37) enables a recursive update of the posterior distribution function $\pi(\boldsymbol{x}_{1:n+1})$ up to time $n+1$, given the past information from $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ and the latest information from the remaining terms on the right side of (2.37), it is not very useful in practice for two reasons. Firstly, it is generally not possible to generate samples directly from the posterior distribution. Secondly, the normalizing constant in the denominator of (2.37), i.e., $p(\boldsymbol{y}_{n+1}|\boldsymbol{y}_{1:n})$, and the desired marginal distributions require the evaluation of complex, multi-dimensional integrals, which are generally difficult or impossible to evaluate analytically. Thus the Bayesian importance sampling procedure is adopted in order to avoid these two problems as well as to generate samples and hence importance weights to update the posterior distribution function.

Similar procedures as in the importance sampling procedure described earlier in this chapter can be followed. Denote an appropriately selected importance function by $q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$, and the importance weights by $w_{1:n}^{(i)}$, defined as

$$w_n^{(i)} \propto \frac{\pi(\boldsymbol{x}_n)}{q(\boldsymbol{x}_n|\boldsymbol{y}_n)}, \; i = 1, ..., N, \tag{2.38}$$

where

$$\sum_{i=1}^{N} w_n^{(i)} = 1, \tag{2.39}$$

such that the posterior distribution $\pi(\boldsymbol{x}_{1:n})$ can be approximated as follows

$$\hat{\pi}_N(d\boldsymbol{x}_{1:n}) = \frac{\sum_{i=1}^{N} w^{(i)}(\boldsymbol{x}_{1:n}^{(i)})\delta_{\boldsymbol{x}_{1:n}^{(i)}}(d\boldsymbol{x}_{1:n})}{\sum_{i=1}^{N} w^{(i)}(\boldsymbol{x}_{1:n}^{(i)})}. \tag{2.40}$$

In order to implement the sequential Monte Carlo procedure, the importance function $q(\cdot|\cdot)$ should be chosen so as to satisfy two conditions (Doucet, 1998). Firstly, the importance function must satisfy the following recursivity condition (Doucet, 1998) (see Appendix B)

$$q(\boldsymbol{x}_{1:n+1}|\boldsymbol{y}_{1:n+1}) = q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n+1}). \tag{2.41}$$

Secondly, the importance function in addition to satisfying (2.41) must be chosen such that the variance of the resulting weights is minimized. It can be shown (Doucet *et al.*, 2000) that such an *optimal* importance function is given by

$$q_{optimal}(\cdot) = q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n, \boldsymbol{y}_{n+1}). \tag{2.42}$$

As a result, by substituting (2.37) and (2.41) into (2.38), the recursive update equation for the importance weight at time $n+1$ can be given as

$$w^{(i)}(n+1) = \tilde{w}^{(i)}(n) \times \frac{p(\boldsymbol{y}_{n+1}|\boldsymbol{x}_{n+1}^{(i)})p(\boldsymbol{x}_{n+1}^{(i)}|\boldsymbol{x}_n^{(i)})}{q(\boldsymbol{x}_{n+1}^{(i)}|\boldsymbol{x}_n^{(i)}, \boldsymbol{y}_{n+1})}, \tag{2.43}$$

where $\tilde{w}^{(i)}(n)$ is a set of normalized importance weights, defined as

$$\tilde{w}^{(i)}(n) = \frac{w^{(i)}(n)}{\sum_{j=1}^{N} w^{(j)}(n)}. \tag{2.44}$$

2.3.2 Degeneracy Problem

A common problem with particle filtering is the degeneracy phenomenon, in which all but a very few particles have negligible weights after a few iterations. It has been shown (Andrieu, 1998) that the degeneracy problem cannot be avoided, since the variance of the importance

weights can only increase over time. As a result, a large computational effort is devoted to updating particles whose contribution to the approximation of the target distribution function is almost zero. A suitable measure of degeneracy of the algorithm is the effective sample size $N_{\text{eff}} \leq N$ introduced in Bergman (1999) and Liu and Chen (1999) and defined as

$$N_{\text{eff}} = \frac{N}{1 + \sigma^2_{w_n^{*(i)}}}, \tag{2.45}$$

where $\sigma^2_{w_n^{*(i)}}$ is the variance of the "true weight" $w^{*(i)}(n)$ (Arulampalam *et al.*, 2002), defined as

$$w^{*(i)}(n) = \frac{\pi(\boldsymbol{x}_n^{(i)}|\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{y}_n)}. \tag{2.46}$$

Since the exact value of N_{eff} may not be easily obtained, an estimate of N_{eff} is given by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}^{(i)}(n))^2}, \tag{2.47}$$

where $\tilde{w}^{(i)}(n)$ is the normalized importance weight. When $\hat{N}_{\text{eff}}$ is small, it indicates severe degeneracy. Although brute force can be used to reduce the undesirable effect of degeneracy by using a very large N, it is often impractical. In addition to the selection of an optimal importance function that minimizes the variance of the weights generated, a procedure to resample the particles is adopted to reduce the effect.

2.3.2.1 Resampling

Because of the degeneracy problem, any estimate based on these very few significant particles would show a large variance. Therefore, in addition to the use of the optimal importance function in (2.42), it is necessary to introduce other procedures to improve the recursion of (2.43).

Resampling is an idea to eliminate the trajectories of the weights which have weak normalized importance weights and to multiply trajectories with strong importance weights. The most popular resampling scheme is Sampling Importance Resampling (SIR) that resamples the particles according to their respective importance weights. The SIR procedure

is to generate a new set of particles $\{\boldsymbol{x}_n^{\star(j)}\}, j = 1, ..., N$ by resampling N times from an approximate posterior distribution function $\hat{\pi}_N(\boldsymbol{x}_n)$, given by

$$\hat{\pi}_N(\boldsymbol{x}_n) \approx \sum_{i=1}^{N} \tilde{w}^{(i)}(n)\delta(\boldsymbol{x}_n - \boldsymbol{x}_n^{(i)}), \tag{2.48}$$

such that $\Pr(\boldsymbol{x}_n^{\star(j)} = \boldsymbol{x}_n^{(i)}) = \tilde{w}^{(i)}(n)$. After the particles $\boldsymbol{x}_n^{(i)}, i = 1, ..., N$, have been resampled, they become *iid* samples, and the respective weights become

$$\tilde{w}^{(i)}(n) = \frac{1}{N},\ i = 1, ..., N. \tag{2.49}$$

One can use this resampling scheme whenever a significant degeneracy is observed, i.e., when $\hat{N}_{\text{eff}}$ falls below some threshold $N_{threshold}$. It can be shown (Ripley, 1987; Doucet, 1998) that the resampling can be done very efficiently with order $\mathcal{O}(N)$ operations. Unfortunately, the trajectories with high importance weights are statisically selected many times, limiting the true statistical diversity amongst the particles. This is the classical problem of depletion of samples, with the result that the cloud of particles may eventually collapse to a single particle.

A more efficient approach, proposed by Andrieu and Doucet (Andrieu *et al.*, 1999; Doucet *et al.*, 2000; MacEachern *et al.*, 1999; Gilks and Berzuini, 1998) uses an MCMC step on each particle. At time n, the particles are marginally distributed as $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$. An MCMC engine with a kernel of invariant distribution $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ can be used to generate a new set of particles that will also be distributed according to this posterior distribution. This approach provides a valid way to re-introduce diversity amongst the particles. In addition, the reversible jump MCMC process can be included such that parameter spaces of varying dimension can be explored to jointly perform model order detection and estimation of the other parameters. The following schema describes the sequential importance sampling procedure applied at each time step $n = 1, 2, \ldots$ for tracking changing parameters.

Generic Sequential Importance Sampling for Tracking an Unknown Number of Parameters

Initialize weights $w^{(i)}(0) = \frac{\pi(\boldsymbol{x}(0))}{q(\boldsymbol{x}(0))}$ for $i = 1, ..., N$.

1. The Importance Sampling Step

   - For $i = 1, \ldots, N$, generate the particles from the importance function

   $$\boldsymbol{x}^{(i)}(n) \sim q(\boldsymbol{x}(n)|\boldsymbol{x}^{(i)}(n-1), \boldsymbol{y}(n), k^{(i)}(n)) \tag{2.50}$$

   - For $i = 1, \ldots, N$, evaluate the importance weights $w^{(i)}(n)$ from (2.43)

   $$w^{(i)}(n) = w^{(i)}(n-1) \times \frac{p(\boldsymbol{y}(n)|\boldsymbol{x}^{(i)}(n), k^{(i)}(n))p(\boldsymbol{x}^{(i)}(n)|\boldsymbol{x}^{(i)}(n-1), k^{(i)}(n))}{q(\boldsymbol{x}^{(i)}(n)|\boldsymbol{x}^{(i)}(n-1), \boldsymbol{y}(n), k^{(i)}(n))} \tag{2.51}$$

   - For $i = 1, \ldots, N$, normalize the weights

   $$\tilde{w}^{(i)}(n) = \frac{w^{(i)}(n)}{\sum_{j=1}^{N} w^{(j)}(n)} \tag{2.52}$$

2. The Sampling Importance Resampling of the Particles

   - Sample a vector of index $\boldsymbol{l}$, with pdf described by the weights

   $$P(l(j) = i) = \tilde{w}^{(i)}(n)$$

   - Resample the particles with the index vector

   $$\boldsymbol{x}_{0:t}^{(i)} = \boldsymbol{x}_{0:t}^{(l(i))}$$

   - Re-assign all the weights to $\tilde{w}^{(i)}(n) = \frac{1}{N}$.

3. Proceed with the Reversible Jump MCMC Step to introduce diversity in the particles and to facilitate detection of model order.

■

Chapter 3

Novel Wideband Data Model

This chapter presents a novel model to represent wideband signals received from an array of sensors, using an interpolation function and past signal samples. This model offers a few advantages: 1) all the data are real, which reduces the computational load required for complex data, 2) the model can accomodate both narrowband and wideband signals without changes in structure or parameters, and 3) wideband signals represented by the model have a similiar structure compared to those of narrowband signals. This chapter describes in detail the derivation of this data model, which is the key component in the development of the two new algorithms for wideband array signal processing in the subsequent chapters.

3.1 Introduction

Most algorithms developed to solve problems for wideband signals usually transform the received signals from the time domain to the frequency domain using the Fourier Transform as a pre-processing step. The motivation of this transformation is that the transformed model in the frequency domain is structurally similar to that for narrowband signals in the time domain, as shown in Section 1.2. In addition, if the sources are uncorrelated in frequency as well as uncorrelated with each other, their frequency spectra in a large set of discretized frequency bins can be considered independent and narrowband. As a result, the algorithms described in Chapter 1 for the narrowband case can be reused in the transformed

model for the wideband case.

Given that the time delay parameters are seldom scalar multiples of the sampling rate, resolving signals from different delays is very difficult, unless the sampling rate is increased significantly, so that the error between the inter-sensor delay and the closest sampling instant is reduced sufficiently. In other words, if one develops an algorithm in an attempt to resolve wideband signals in the time domain, oversampling may be required and more sophisticated hardware is also required. Thus these burdens are further reasons why most algorithms for wideband signal processing operate in the frequency domain.

The novel data model to be introduced, however, is developed in the time domain, in which delayed versions of wideband signals are expressed in terms of their respective past samples and an appropriately selected interpolation function. The interpolation function models the received signal by returning a set of weights, which are a function of the desired directions of arrival (DOAs), for combining past samples of the received signal. By fitting this interpolation-based model to the received data, the source DOAs can be estimated. In Chapter 5, we first use the novel data model with the assumption that the DOAs and the number of sources are static throughout the entire observation period, followed in Chapter 6 by an extension of the model to the case when both the DOAs and the number of sources are time-varying.

3.2 Development of the data model for a single source

A uniform linear array with M elements is assumed.[1] For the time being, assume further that 1) we are operating in a noiseless environment, 2) only a single source $s(t)$ is incident onto the array, where the source $s(t)$ may be a wideband signal, whose spectrum is band limited to

$$|f| \in \left[f^l, f^u\right], \quad f^u = f^l + \Delta f, \tag{3.1}$$

[1] For ease of presentation, we consider the uniform linear array case. However, the method can be extended to arbitrary linear geometries.

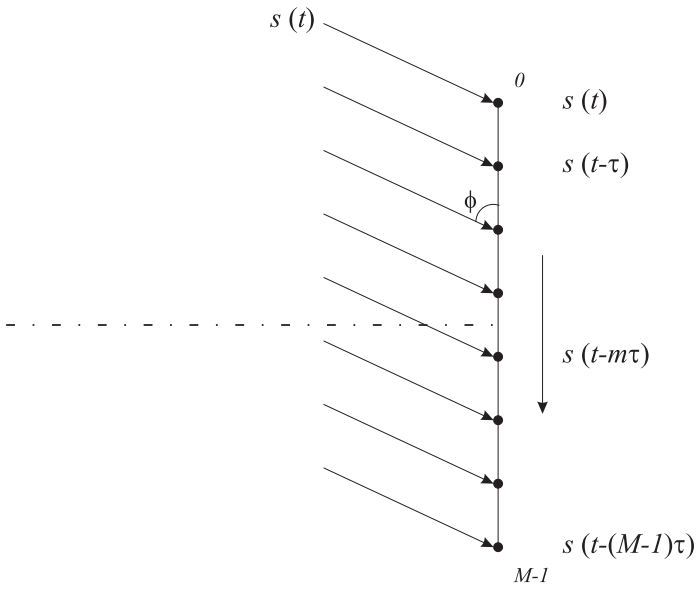

Figure 3.1: A signal $s(t)$ impinging onto the array when the incident angle ϕ is less than $\pi/2$.

where f^l and f^u are the lower and upper frequencies, and Δf is the bandwidth of the signal, and 3) the signal $s(t)$ impinges onto the array at an angle ϕ with respect to the axis of the array, as illustrated in Figure 3.1. Note that the angle ϕ is assumed constant throughout the entire observation period. Denote the inter-sensor delay (ISD) by τ, given by (Johnson, 1982)

$$\tau \triangleq \frac{\Delta}{C} \sin \phi, \tag{3.2}$$

where Δ is the interspacing of the sensors and C is the speed of propagation. Assuming that

$$\Delta \leq \frac{1}{2}\lambda_{min} = \frac{C}{2f_{max}}, \tag{3.3}$$

where f_{max} is the maximum frequency component in the signal, we can write

$$\frac{\Delta}{C} \leq \frac{1}{2f_{max}}. \tag{3.4}$$

Comparing (3.4) to (3.2), we have

$$|\tau| \leq \frac{1}{2f_{max}} = \frac{1}{2f^u} \triangleq T_{max}. \tag{3.5}$$

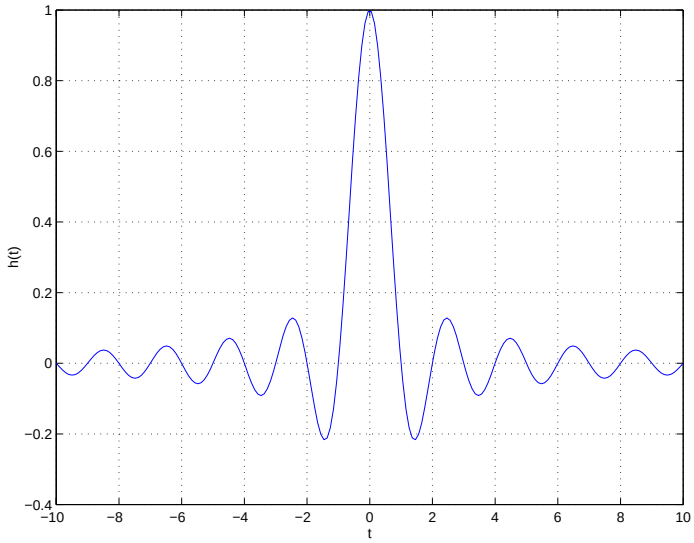

Figure 3.2: Impulse response of an ideal reconstructor.

Therefore, we have $\tau \in [-T_{max}, T_{max}]$.

Given that the wideband signal $s(t)$ is bandlimited, it can be sampled to yield a set of discrete samples, $\{s(lT_s), l = -\infty, ..., \infty\}$, where T_s is the sampling period. With the use of an *ideal reconstructor*, the signal can be reconstructed in terms of these samples as follows

$$s(t) = \sum_{l=-\infty}^{\infty} s(lT_s)\psi(t - lT_s), \tag{3.6}$$

where $\psi(t)$ is the sinc function, given by

$$\psi(t) = \frac{\sin(\pi t/T_s)}{\pi t/T_s}. \tag{3.7}$$

Figure 3.2 shows an example of an ideal reconstructor. Let τ be a delay parameter, defined as in (3.2), and $s(t-\tau)$ be the delayed version of the signal $s(t)$, which can also be expressed in terms of the discrete samples as

$$s(t - \tau) = \sum_{l=-\infty}^{\infty} s(lT_s)\psi(t - \tau - lT_s) \tag{3.8}$$

The shifted versions of $\psi(t - \tau)$ have the identical shape of $\psi(t)$ with the peak at $t = \tau$, as depicted in Figure 3.3. When a snapshot of $s(t - \tau)$ at $t = nT_s$ is considered, we can

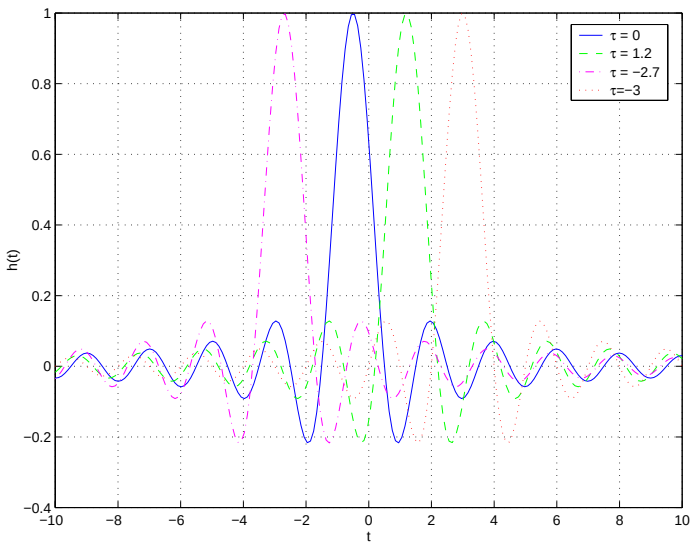

Figure 3.3: Impulse response of an ideal reconstructor with different time shifts.

rewrite $s(t-\tau)$ as follows

$$\begin{aligned} s(t-\tau)_{|t=nT_s} &= \sum_{l=-\infty}^{\infty} s(lT_s)\psi(t-\tau-lT_s)_{|t=nT_s}, \\ s(nT_s-\tau) &= \sum_{l=-\infty}^{\infty} s(lT_s)\psi((n-l)T_s-\tau). \end{aligned} \tag{3.9}$$

By substituting l with $n-l$ in (3.9), and replacing the infinite sum with a finite sum over $2L+1$ samples, we have

$$s(n-\tau) \approx \sum_{l=-L}^{L} \psi_l(\tau)s(n-l), \tag{3.10}$$

where we have adopted the convention that T_s is normalized to unity. In the above,

$$\psi_l(\tau) \triangleq \psi(lT_s-\tau) \equiv \psi(l-\tau). \tag{3.11}$$

The interpolation process in (3.10) is non-causal. This implies that the processing must undergo a delay of L samples. In some cases, this delay is undesirable. We can eliminate this delay at the expense of some additional error in the interpolation process, by replacing

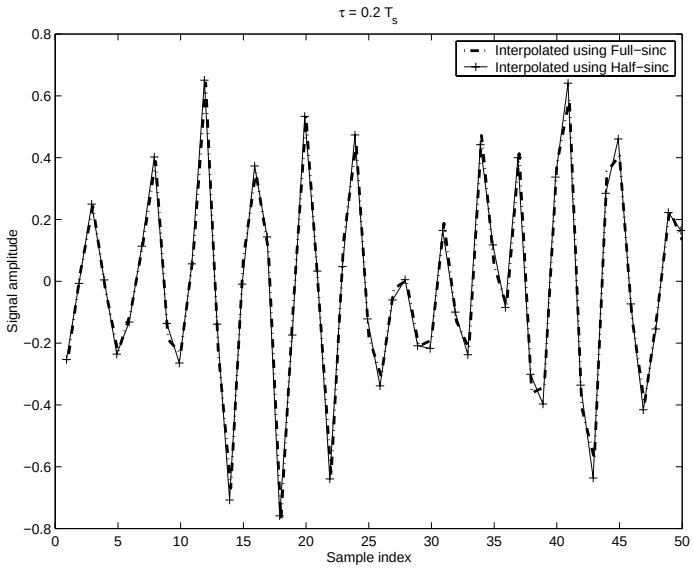

Figure 3.4: A comparison between the interpolation performance between a half-sinc function and a full-sinc function for an AR process when $\tau = 0.2T_s$.

(3.10) with

$$s(n-\tau) \approx \begin{cases} \sum_{l=0}^{L-1} \psi_l(\tau) s(n-l), & \text{if } \tau \geq 0 \\ \sum_{l=-L+1}^{0} \psi_l(\tau) s(n+l), & \text{if } \tau < 0 \end{cases}. \tag{3.12}$$

The additional error introduced by (3.12) in most cases is small. This follows because the interpolation weights, i.e., $\psi_l(\tau)$, excluded from (3.12) are outside the main lobe of $\psi_l(\tau)$ and are therefore small, especially when τ is small or close to T_s. These excluded weights are made smaller by the application of a window function, which is imposed as follows

$$h_l(\tau) = \psi(l-\tau) \times w(l-\tau), \quad l = 0, ..., L-1, \tag{3.13}$$

where, for example, $w(l)$ is the Hamming window, defined as follows

$$w(l) = 0.54 + 0.46 \cos\left(\frac{\pi l}{L-1}\right), \quad 0 \leq l \leq L-1. \tag{3.14}$$

For ease of notation, we replace the approximation in (3.12) with an equality in the sequel.

Figures 3.4 – 3.7 depict a comparison between an AR process interpolated by (3.10) and (3.12) for different values of τ. For easy comparison, the waveforms interpolated by the

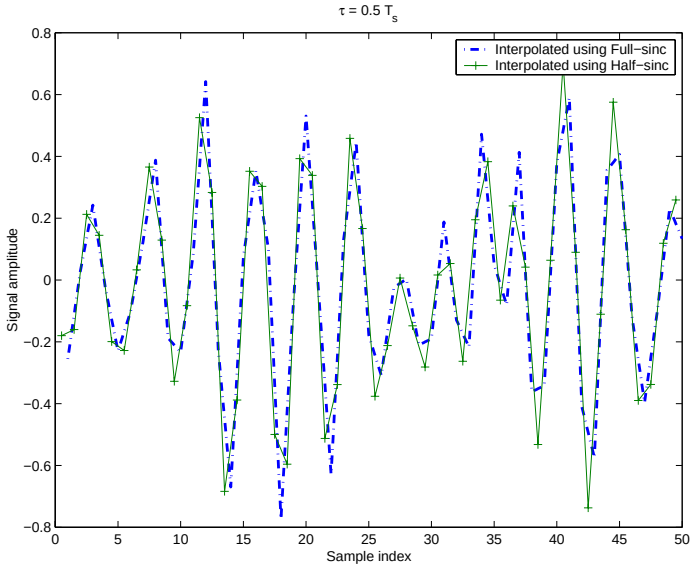

Figure 3.5: A comparison between the interpolation performance between a half-sinc function and a full-sinc function for an AR process when $\tau = 0.5T_s$.

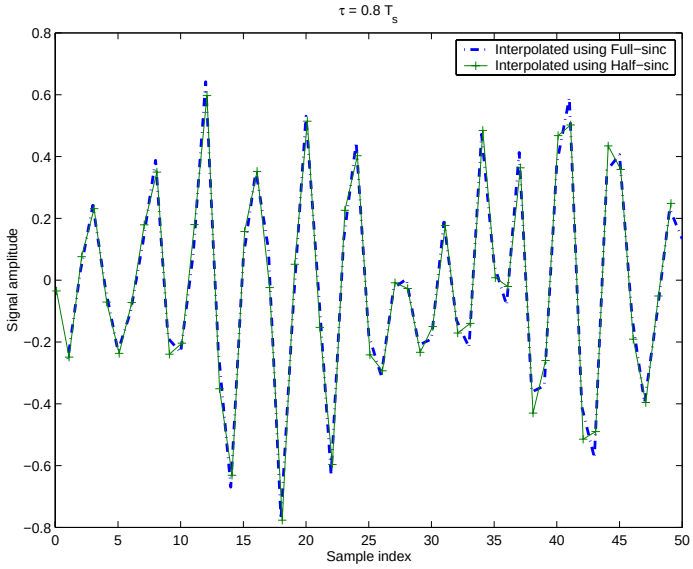

Figure 3.6: A comparison between the interpolation performance between a half-sinc function and a full-sinc function for an AR process when $\tau = 0.8T_s$.

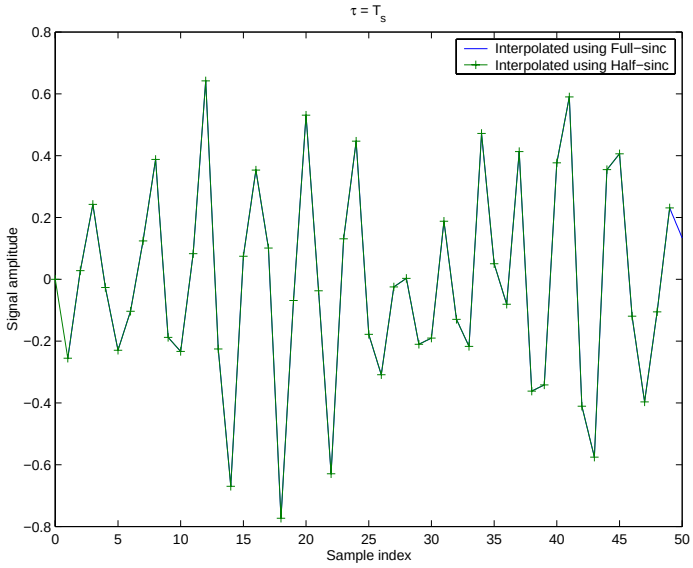

Figure 3.7: A comparison between the interpolation performance between a half-sinc function and a full-sinc function for an AR process when $\tau = T_s$.

half-sinc approach, i.e., (3.12), are shifted by the respective τ's to align with the original waveforms and the waveforms interpolated by the full-sinc approach, i.e., (3.10). When τ is close to 0 or T_s, the waveforms interpolated by the two approaches are indistinguishable, as shown in Figures 3.4 and 3.7. When τ is near $0.5T_s$, the performance of the half-sinc approach begins to degrade, but the waveforms so constructed by the approach are very close to the full-sinc approach. The mean-squared errors between the waveforms constructed by these approaches for different values of τ is listed in Table 3.1. Therefore, the half-sinc approach is adopted in the development of the wideband data model throughout this thesis due to the simplicity and advantage of zero-delay processing. However, in some exceptional circumstances, especially when τ approaches $0.5T_s$, it may be necessary to delay the interpolation function several samples, and thus introduce processing delay, in order to maintain a sufficient level of accuracy.

Accordingly, the wideband signal $s(t)$, delayed by τ, at $t = n$ using its L discrete samples can be written as

$$s(n-\tau) = \sum_{l=0}^{L-1} h_l(\tau) s(n-l), \tag{3.15}$$

τ/T_s	MSE
0	5.6×10^{-8}
0.2	6.1×10^{-7}
0.5	3.1×10^{-4}
0.8	1.9×10^{-6}
1	1.6×10^{-9}

Table 3.1: Mean-squared error between the half-sinc and the full-sinc approaches.

which is the fundamental expression used to construct the data model in this chapter.

Consider the case where ϕ is less than $\pi/2$ with respect to the axis of the array as shown in Figure 3.1. In the noise-free case, according to (3.15), we can approximate the snapshot at the mth sensor as follows

$$s(n - m\tau) = \sum_{l=0}^{L-1} h_l(m\tau)s(n - l), \quad m = 0, \ldots, M - 1. \tag{3.16}$$

Accordingly, we can now express the snapshot vector of the received signal $\boldsymbol{s}_\tau(n) = [s(n), s(n-\tau), \ldots, s(n-(M-1)\tau)]^T$ in (3.16) for $m = 0, 1, ..., M-1$ as follows

$$\begin{bmatrix} s(n) \\ s(n-\tau) \\ \vdots \\ s(n-(M-1)\tau) \end{bmatrix} = \begin{bmatrix} h_0(0) & h_1(0) & \ldots & h_{L-1}(0) \\ h_0(\tau) & h_1(\tau) & \ldots & h_{L-1}(\tau) \\ \vdots & \vdots & \ddots & \vdots \\ h_0((M-1)\tau) & h_1((M-1)\tau) & \ldots & h_{L-1}((M-1)\tau) \end{bmatrix} \begin{bmatrix} s(n) \\ s(n-1) \\ \vdots \\ s(n-L+1) \end{bmatrix} \tag{3.17}$$

which can be written as

$$\boldsymbol{s}_\tau(n) = \boldsymbol{H}(\tau)\boldsymbol{s}(n), \tag{3.18}$$

where $\boldsymbol{s}_\tau(n)$ is defined as the vector on the left-hand side of (3.17), and where $\boldsymbol{s}(n)$ is known as the signal vector

$$\boldsymbol{s}(n) \;=\; [s(n), s(n-1), \ldots, s(n-L+1)]^T, \tag{3.19}$$

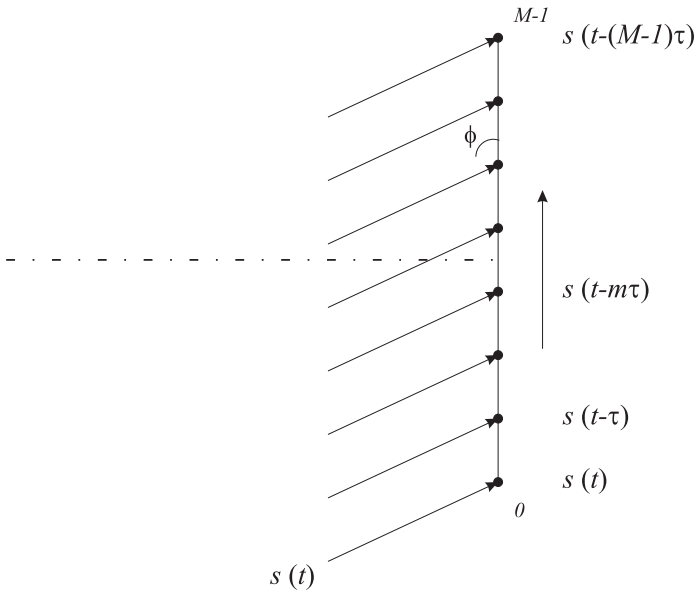

Figure 3.8: A signal $s(t)$ impinging onto the array when the incident angle ϕ is greater than $\pi/2$.

and $\boldsymbol{H}(\tau) \in \mathcal{R}^{M\times L}$ is an *interpolation matrix* for τ and is defined in terms of L column vectors as

$$\boldsymbol{H}(\tau) \triangleq [\boldsymbol{h}_0(\tau), \boldsymbol{h}_1(\tau), \ldots, \boldsymbol{h}_{L-1}(\tau)], \tag{3.20}$$

where $\boldsymbol{h}_l(\tau) \in \mathcal{R}^{M\times 1}$ is the lth column of the interpolation matrix $\boldsymbol{H}(\tau)$, defined as

$$\boldsymbol{h}_l(\tau) \triangleq [h_l(0), h_l(\tau), \ldots, h_l((M-1)\tau)]^T. \tag{3.21}$$

Accordingly, we can rewrite the sensor output vector in the form

$$\boldsymbol{s}_\tau(n) = \boldsymbol{h}_0(\tau)s(n) + \sum_{l=1}^{L-1} \boldsymbol{h}_l(\tau)s(n-l). \tag{3.22}$$

In other words, the sensor output can be expressed as a linear combination of a set of L vectors of $\{\boldsymbol{h}_l(\tau), l = 0, 1, ..., L-1\}$ with the signals $\boldsymbol{s}(n)$ as the associated coefficients.

In cases where the incident angle ϕ of a signal $s(t)$ is greater than $\pi/2$ with respect to the 0th sensor, as illustrated in Fig. 3.8, the $(M-1)$th sensor becomes the first sensor that receives $s(t)$, while the 0th sensor will not receive the signal until $(M-1)\tau$ seconds later.

Following the above derivations for $\phi \leq \pi/2$, we may express the snapshot at the pth sensor output in the following manner

$$s(n - p\tau) = \sum_{l=0}^{L-1} h_l(p\tau)s(n - l), \tag{3.23}$$

where $p = M - 1 - m$. Extending the above for all $p = M - 1, M - 2, ..., 0$, we can write

$$\boldsymbol{s}_\tau(n) = \boldsymbol{E}_M \boldsymbol{H}(\tau)\boldsymbol{s}(n), \tag{3.24}$$

where $\boldsymbol{E}_M \in \mathcal{R}^{M\times M}$ is a *exchange matrix* defined as follows (Golub and Loan, 1993)

$$\boldsymbol{E}_M = \begin{bmatrix} 0 & 0 & \dots & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.25}$$

In other words, if the incident angle ϕ of a signal is greater than $\pi/2$ with respect to the 0th sensor, the expression of the sensor output vector is the same as that defined in (3.18) pre-multiplied by the exchange matrix $\boldsymbol{E}_M$ that reverses the order of rows in the interpolation matrix $\boldsymbol{H}(\tau)$. Therefore, the general expression for a sensor output vector in response to an incoming signal $s(t)$ at an angle ϕ is

$$\boldsymbol{s}_\tau(n) = \tilde{\boldsymbol{H}}(\tau)\boldsymbol{s}(n) = \tilde{\boldsymbol{h}}_0(\tau)s(n) + \sum_{l=1}^{L-1} \tilde{\boldsymbol{h}}_l(\tau)s(n - l) \tag{3.26}$$

where

$$\tilde{\boldsymbol{H}}(\tau) = \begin{cases} \boldsymbol{H}(\tau), & \text{if} \quad \phi \leq \pi/2 \\ \boldsymbol{E}_M\boldsymbol{H}(\tau), & \text{if} \quad \phi > \pi/2 \end{cases}, \qquad \text{where } \tilde{\boldsymbol{H}}(\tau) \in \mathcal{R}^{M\times L}. \tag{3.27}$$

An analogous definition holds for the vector $\tilde{\boldsymbol{h}}_l(\tau)$.

3.3 Development of the data model for multiple sources

The model above can be easily extended to cases where there exist multiple sources $K \geq 1$, where K is constant throughout the entire observation period. The incident angle and delay

parameters now become vectors, i.e., $\boldsymbol{\phi}$ and $\boldsymbol{\tau}$, respectively. They are defined as

$$\boldsymbol{\phi} = [\phi_0, \phi_1, \ldots, \phi_{K-1}]^T, \tag{3.28}$$

$$\boldsymbol{\tau} = [\tau_0, \tau_1, \ldots, \tau_{K-1}]^T. \tag{3.29}$$

To extend the model described earlier to the K-dimensional case, we need to add a subscript k to τ and $s(t)$ such that the snapshot vector in (3.26) for the kth source becomes

$$\boldsymbol{s}_{\tau_k}(n) = \tilde{\boldsymbol{H}}(\tau_k)\boldsymbol{s}_k(n) = \tilde{\boldsymbol{h}}_0(\tau_k)s_k(n) + \sum_{l=1}^{L-1} \tilde{\boldsymbol{h}}_l(\tau_k)s_k(n-l),\ k = 0, \ldots, K-1, \tag{3.30}$$

where

$$\boldsymbol{s}_{\tau_k}(n) = [s_k(n), s_k(n-\tau_k), \ldots, s_k(n-(M-1)\tau_k))]^T. \tag{3.31}$$

Similarly, each wideband signal will have its own upper and lower frequencies, f_k^u and f_k^l, such that T_{max} in the K-dimensional case, according to (3.5), is now defined as

$$T_{max} = \min_{k=0,\ldots,K-1} \left\{ \frac{1}{2f_k^u} \right\}. \tag{3.32}$$

We define the index $\acute{m}$ as follows

$$\acute{m} = \begin{cases} m, & \text{if} \quad \phi_k \leq \pi/2 \\ M-1-m, & \text{if} \quad \phi_k > \pi/2 \end{cases}, \tag{3.33}$$

such that the delayed version $s_k(n - \acute{m}\tau_k)$ of the signal vector $\boldsymbol{s}_{\tau_k}(n)$ may be expressed as

$$s_k(n - \acute{m}\tau_k) = \sum_{l=0}^{L-1} h_l(\acute{m}\tau_k)s_k(n-l), \quad k = 0, \ldots, K-1. \tag{3.34}$$

The $\acute{m}$th sensor output $y_{\acute{m}}(n)$ in the presence of noise now becomes

$$y_{\acute{m}}(n) = \sum_{k=0}^{K-1} s_k(n - \acute{m}\tau_k) + \sigma_w w_{\acute{m}}(n), \tag{3.35}$$

$$= \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} h_l(\acute{m}\tau_k)s_k(n-l) + \sigma_w w_{\acute{m}}(n), \quad \acute{m} = 0, \ldots, M-1, \tag{3.36}$$

where $\boldsymbol{s}_k(n)$ is defined as in (3.19) for the kth source, $w_{\acute{m}}(n)$ is an *iid* Gaussian variable with zero mean and unit variance, and σ_w^2 is the noise variance in the observation.

We now define a signal vector for K sources at time n as

$$\boldsymbol{a}(n) \triangleq [s_0(n), s_1(n), \ldots, s_{K-1}(n)]^T, \tag{3.37}$$

such that (3.36) for $\acute{m} = 0, ..., M-1$ may be re-written in the form

$$\boldsymbol{y}(n) \;\; = \;\; \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n) + \sum_{l=1}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l) + \sigma_w \boldsymbol{w}(n), \tag{3.38}$$

where $\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}) \in \mathcal{R}^{M \times K}$ is defined as

$$\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}) = \left[\tilde{\boldsymbol{h}}_l(\tau_0), \tilde{\boldsymbol{h}}_l(\tau_1), \ldots, \tilde{\boldsymbol{h}}_l(\tau_{K-1})\right]. \tag{3.39}$$

and $\boldsymbol{w}(n) \in \mathcal{R}^M$ is a vector whose elements are $w_m(n)$. The signal vectors $\boldsymbol{a}(n-l)$ for $l = 1, ..., L-1$ in (3.38) are estimated by a proposed algorithm in a manner to be described, and may therefore be considered known. In this vein, we define a vector $\boldsymbol{z}(n)$ as follows

$$\boldsymbol{z}(n) \triangleq \boldsymbol{y}(n) - \sum_{l=1}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l), \tag{3.40}$$

which can be interpreted as the error between the snapshot $\boldsymbol{y}(n)$ and its approximation, based on the past, estimated values of the signals from $\boldsymbol{a}(n-1)$ to $\boldsymbol{a}(n-L+1)$ and the associated columns in the interpolation matrix $\boldsymbol{H}(\boldsymbol{\tau})$. Accordingly, we can rewrite (3.38) as follows

$$\boldsymbol{z}(n) \;\; = \;\; \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n) + \sigma_w \boldsymbol{w}(n), \tag{3.41}$$

which represents the desired form of the model.

3.4 Features of the data model

There are several interesting features regarding the model in (3.41). It is similar to the familiar narrowband model (Johnson, 1982), except that the data is modified and the steering matrix takes the form of an interpolation matrix. The similarities and differences between the traditional narrowband model and the novel model are described as follows:

- Snapshots – For narrowband signals, the snapshot $\boldsymbol{y}(n)$ corresponds to the sensor outputs in the array. For wideband signals, the modified snapshot $\boldsymbol{z}(n)$ represents the estimation error between the sensor output $\boldsymbol{y}(n)$ and its approximation using a linear combination of $L-1$ past samples, where the weights are the columns in the interpolation matrix.

- Arguments – For narrowband signals, the argument is $\boldsymbol{\phi}$, where each element, ϕ_k, is bounded between $-\pi/2$ and $\pi/2$. For wideband signals, the argument is $\boldsymbol{\tau}$, where each element, τ_k, is bounded between $-T_{max}$ and T_{max}.

- Process matrices – For narrowband signals, the process matrix is the steering matrix $\boldsymbol{S}(\boldsymbol{\phi})$ whose elements are complex-valued. For wideband signals, the process matrix, $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})$, is made of the 0th columns of the individual interpolation matrices of the sources, where all elements are purely real.

- The m,kth elements in the process matrices – For narrowband signals, if the inter-sensor distance is taken to be half-wavelength, the m,kth element in $\boldsymbol{S}(\boldsymbol{\phi})$ is governed by an exponential function as $\exp\{-j\frac{m\pi}{2}\phi_k\}$. For wideband signals, the m,kth element in $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})$ can be expressed by $\mathrm{sinc}(m\tau_k)\times w(m\tau_k)$, where $w(\cdot)$ is the smoothing window.

As can be seen, there is a close tie between the traditional narrowband model and the novel model for wideband signals, and this allows the reuse of existing algorithms specifically developed for narrowband signals, like MCMC and particle filters for joint detection and estimation, and tracking problems, respectively, to the wideband case. Table 3.2 summarizes the comparison between these two models.

Further, the model in (3.41) can also accomodate either narrowband or wideband sources, without change of structure or parameters. Also, all quantities in (3.41), including the data, are pure real, unlike previous models in (Valaee and Kabal, 1992, 1995; Wang and Kaveh, 1985; Hung and Kaveh, 1988) which require complex quantities. This latter point leads to significant savings in hardware, since quadrature mixing to IF frequencies is

Description	Narrowband	Wideband
Snapshots	$\boldsymbol{y}(n) = \boldsymbol{S}(\boldsymbol{\phi})\boldsymbol{a}(n) + \sigma_v \boldsymbol{v}(n)$	$\boldsymbol{z}(n) = \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n) + \sigma_v \boldsymbol{v}(n)$
Arguments	$\phi_k \in [-\frac{\pi}{2}, \frac{\pi}{2}]$	$\tau_k \in [-T_{max}, T_{max}]$
Process matrices	$\boldsymbol{S}(\boldsymbol{\phi}) \in \mathcal{C}^{M \times K}$	$\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}) \in \mathcal{R}^{M \times K}$
The m,kth elements of the process matrices	$\exp\left\{\frac{m\pi}{2}\phi_k\right\}$	$\mathrm{sinc}(m\tau_k) \times w(m\tau_k)$

Table 3.2: Comparison between the data models for narrowband and wideband signals.

no longer required, and computational requirements are reduced.

3.5 Development of the data model when the parameters of interest are time-varying

The data model derived earlier assumes that the ISD vector $\boldsymbol{\tau}$ and the model order K are both constant in the entire observation period. As this model is developed for static sources, modifications to the model are needed in order to address the situation when the static assumption is no longer valid, i.e., when the sources are moving and the number of sources is time-varying.

To address the case when the parameters of interest become time-varying, we modify the data model developed for static scenario by introducing the time variable t to the respective parameters. Assume there are $k(t)$ wideband signals, $s_k(t)$, for $k = 0, ..., k(t)-1$, impinging at distinct angles $\phi_k(t)$ onto a uniform linear array of M identical sensors. Further, we assume that these angles do not change significantly over L samples, where L is the number of taps defined earlier. These sources are bandlimited as in (3.1). The corresponding inter-sensor delay $\tau_k(t)$ becomes

$$\tau_k(t) \triangleq \frac{\Delta}{C} \sin \phi_k(t), \tag{3.42}$$

where $\tau_k(t) \in [-T_{max}, T_{max}]$, and

$$T_{max} = \min_{k=0,...,k(t)-1} \left\{ \frac{1}{2f_k^u} \right\}. \tag{3.43}$$

Following the development of the observation vectors for the static scenario, we can define the observation vector $\boldsymbol{y}(t)_{|t=n} \in \mathcal{R}^{M\times 1}$ in response to the $k(t)_{|t=n}$ wideband signals as follows

$$\boldsymbol{y}(n) = \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}(n))\boldsymbol{a}(n-l) + \sigma_w \boldsymbol{w}(n), \tag{3.44}$$

where $\boldsymbol{\tau}(n) \in \mathcal{R}^{k(n)\times 1}$, $\boldsymbol{a}(n) \in \mathcal{R}^{k(n)\times 1}$ and $\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}(n)) \in \mathcal{R}^{M\times k(t)}$ are defined similarly in (3.29), (3.37) and (3.39), respectively, and $\boldsymbol{w}(n) \in \mathcal{R}^{M\times 1}$ is an *iid* Gaussian variable defined as

$$\boldsymbol{w}(n) \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}_M\right), \tag{3.45}$$

and σ_w^2 is the observation noise variance. Finally, we define a vector $\boldsymbol{z}(n)$ as follows

$$\boldsymbol{z}(n) \triangleq \boldsymbol{y}(n) - \sum_{l=1}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}(n))\boldsymbol{a}(n-l), \tag{3.46}$$

which can be interpreted as the error between the observation $\boldsymbol{y}(n)$ and its approximation, based on the past, estimated values of the signals from $\boldsymbol{a}(n-1)$ to $\boldsymbol{a}(n-L+1)$ and the associated columns in the interpolation matrix $\boldsymbol{H}_l(\boldsymbol{\tau}(n))$ for $l = 1, ..., L-1$. Accordingly, we can rewrite (3.44) as follows

$$\boldsymbol{z}(n) = \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}(n))\boldsymbol{a}(n) + \sigma_w \boldsymbol{w}(n), \tag{3.47}$$

which represents the desired form of the model. This model for nonstatic sources also shares the same features as decribed in Section 3.4 with the exception that both ISDs and model order are time-varying in (3.47). Furthermore, this model is fit to address the problem of online tracking for time-varying ISDs $\boldsymbol{\tau}(n)$ and model order $k(n)$ using the sequential Monte Carlo algorithm or particle filters.

Chapter 4

Advanced Beamforming for Narrowband Signals

In this chapter, we present a new alternative to beamforming for extraction of multiple waveforms of desired sources in the presence of interfering signals. Waveform extraction is useful for communications in hostile environments and to aid in classification of targets in radar applications.

Conventional approaches to this problem use a sequence of disjoint procedures for waveform extraction. These include model order detection, direction of arrival (DOA) estimation, DOA tracking, and then finally beamforming. In contrast, the proposed approach combines all these processes jointly. A distinct advantage of the proposed method is the fact that it is effective in highly nonstatic environments, where classical beamforming approaches fail. Unlike modern competing methods, like the LPA-beamformer (Gershman, 1999; Katkovnik and Gershman, 2000), the proposed method is not model sensitive, and does not assume smooth DOA motion within an observation window.

The proposed method is based on sequential Monte Carlo (otherwise known as *particle filtering*) techniques for estimation and tracking of the required DOAs. Once the DOA estimates are available, the desired source waveforms can be extracted using a *maximum a posteriori* (MAP) procedure. Model order detection is obtained using a reversible jump

Markov chain Monte Carlo (RJMCMC) resampling algorithm. Simulation results, which compare performance to the extended Kalman filter and the LPA beamformer, are presented.

4.1 Introduction

The problem of recovering waveforms from multiple sources is of considerable interest in electronic systems. This problem has application in communications in hostile environments, or in radar where the waveforms from the multiple targets are to be used to aid in target classification. Standard approaches to this problem rely on arrays of sensors; first, the number of sources or targets is detected (this is otherwise known as order detection), then the corresponding directions-of-arrival (DOAs) are estimated, and then tracked. Once smoothed or filtered DOA estimates are available, the desired waveforms can then be recovered using beamforming techniques. In this chapter we concentrate specifically on the more difficult case where the sources are highly mobile or not static in angle. In the following portion of this section, we briefly discuss the shortcomings of the order detection, the DOA estimation, the tracking, and the beamforming components as they relate to the waveform extraction problem.

As we see later, for signal recovery to perform well, the DOAs must be properly estimated and tracked. Current DOA estimation techniques (Multiple Signal Classification (MUSIC) (Schmidt, 1986), Maximum Likelihood (ML) (Reilly and Haykin, 1982), Weighted Subspace Fitting (WSF), (Viberg and Ottersen, 1991) etc) do not perform well in the case where the DOAs are not static since these methods depend on covariance matrices averaged over several past observations. Thus, the environment must be homogeneous and static over the observation window, and performance degrades when these conditions are violated. An analogous situation holds with respect to conventional order detection methods (e.g., the AIC and MDL criteria (Wax and Kailath, 1985)).

Conventionally, in DOA tracking applications, the DOA estimation phase precedes the tracking process as a completely disjoint operation. That is, signal processing precedes

state estimation. In particular, tracking using DOA-only measurements, which has been considered extensively in the literature (see Kurien and Washburn (1985); Jauffret and Bar-Shalom (1990); Pattipati *et al.* (1992); Rao *et al.* (1994)), is a challenging problem because of nonlinearity and observability problems.

In the case where the DOAs are not static, DOA estimation and tracking can be approached using *state-space models*. The aim is to estimate the state process using the observations that update the posterior distribution of interest as new observations arrive. Classical methods to obtain approximations to the desired distributions include analytical approximations, such as the extended Kalman filter (Anderson and Moore, 1979), the Gaussian sum filter (Aspach and Sorenson, 1972), and deterministic numerical integration techniques (Bucy and Senne, 1971). The extended Kalman filter (EKF) and Gaussian sum filter are computationally cheap, but fail or diverge in some difficult scenarios. In particular, if the nonlinearities in the state and measurement equations are significant, then the EKF's performance is degraded (Bar-Shalom *et al.*, 2001).

Beamforming (Johnson, 1982; Johnson and Dudgeon, 1993), a form of spatial filtering, is regarded as the ability of an array to capture signals incident from a particular direction. The objective of beamforming is to estimate a desired signal waveform arriving from a known direction, in the presence of noise and interfering signals, or to separate desired signals incident from different spatial locations or directions. The extracted signals can then be used as signatures or features for classification among different objects or targets in radar and sonar applications. Beamforming has also proven very successful in communications in hostile environments.

Typical beamforming approaches were developed assuming one desired source and considering other incident sources as interference. To protect the desired source and to suppress the others, precise knowledge of the desired source DOA must be available. Methods like the Minimum Variance Distortionless Response (MVDR) and the Generalized Sidelobe Canceller (GSC), (Johnson, 1982) etc., are just a few examples that require these assumptions to be valid for reasonable performance. Thus these algorithms could only find application in the case where the signal sources or targets are *static* and where their positions are perfectly

known throughout the entire engagement. Unfortunately, in practical systems and scenarios these assumptions are easily violated: the target source may move from observation to observation and the assumed DOAs may be different from the actual ones, and the number of sources may not be known, resulting in degraded performance in signal extraction. As a result, in applications where the sources are mobile, traditional beamforming techniques might not be very precise and better techniques are needed.

Recently, beamforming methods have been developed to handle sources that are not static (Gershman, 1999; Katkovnik and Gershman, 2000; Wigren and Eriksson, 1995). These methods propose the use of a polynomial model to estimate DOA motion within a short observation period (window) so that beamforming is possible in the environments where the DOAs are slowly changing. With these methods, the angular velocities of the DOA motion is taken into account, and optimization techniques are then used to search for the optimal DOAs of the sources. Once the estimates are available, traditional beamforming algorithms could then be used to extract signals of interest. While these methods take target motion into account, they still do not address the critical model order selection problem; also, the performance relies heavily on the smoothness of target motion within the observation period. The PASTd approach (Yang, 1995) has also been proposed as a means of dealing with targets that are not static. This method estimates a noise subspace over a sliding window from the observed data. Then, from this subspace, the number of sources can be determined and the DOAs tracked. Thus, assumptions about static sources and the knowledge of the number of sources are relaxed with this approach. However, the PASTd method assumes that the subspace is constant within the window, which means the DOAs must be almost static. Thus the method fails to track and extract the sources if they move substantially within the window.

The performance of standard beamforming algorithms is sensitive to error in the DOA estimate of the target, and deteriorates very significantly if the estimated DOA is different from the true one. Robustness to DOA uncertainty (Gershman, 1999) has been introduced such that the main lobe width is traded off with the degradation caused by the deviation of the estimated and true DOAs.

In this chapter, we present a novel alternative method to beamforming for online recovery of an unknown number of desired source waveforms in the presence of interference, that is effective in conditions where the DOAs are not static. The proposed method performs the DOA estimation/tracking, and the order detection components of the waveform extraction problem jointly in one seamless operation, rather than disjointly as before. The proposed method uses sequential Monte Carlo (MC) methods in conjunction with Markov Chain Monte Carlo (MCMC) methods (Andrieu *et al.*, 1999, 1998). The sequential MC methods (Doucet *et al.*, 2001; Andrieu *et al.*, 1999; Gordon *et al.*, 1993; Doucet, 1998) are suitable for estimating the state process using the observations and hence recursively updating the posterior distribution of interest as new observations arrive. Model order detection is achieved using the reversible jump MCMC procedure (Green, 1995). Simulation results show that the proposed particle filtering approach for waveform recovery does not suffer from the drawbacks associated with the more standard approaches, as discussed above, at addressing this problem.

The proposed procedure is first to determine the number of incident signals, and then jointly estimate their respective DOAs using sequential Monte Carlo methods. We then use the estimated DOAs to extract the desired source amplitudes using a *maximum a posteriori* (MAP) procedure. The desired sources can be distinguished from interferences using a priori knowledge of the desired source waveforms.

4.2 The State-Space Model

The transmission medium is assumed to be isotropic and nondispersive so that the radiation propagates along straight lines, and the sources are assumed *narrowband* and in the *far-field* of the array. In other words, the radiation impinging on the array is in the form of a sum of plane waves.

We denote the number of narrowband plane waves impinging on an M-element array from distinct directions at time n as $k(n)$, which is unknown and time–varying, such that $k(n) < M$ for all n. The signals are assumed to have the same known centre frequency and,

hence, the effect of a time delay on the received waveforms is simply a phase shift. Denoting the DOA vector by $\boldsymbol{\phi}(n) \in [0, 2\pi]^{k(n)}$, we define the steering matrix by $\boldsymbol{S}(\boldsymbol{\phi}(n)) \in \mathcal{C}^{M \times k(n)}$ as follows

$$\boldsymbol{S}(\boldsymbol{\phi}(n)) \quad = \quad [\boldsymbol{s}(\phi_1(n)), \boldsymbol{s}(\phi_2(n)), \ldots, \boldsymbol{s}(\phi_k(n))], \tag{4.1}$$

where $k = 1, 2, ..., k(n)$. Each column of $\boldsymbol{S}(\boldsymbol{\phi}(n))$ is the steering vector corresponding to a particular source, defined as follows

$$\boldsymbol{s}(\phi_k(n)) \quad = \quad [e^{-jd_1\xi_k(n)}, e^{-jd_2\xi_k(n)}, \ldots, e^{-jd_M\xi_k}]^T, \tag{4.2}$$

where

$$\xi_k(n) = (2\pi/\lambda) \sin \phi_k(n). \tag{4.3}$$

The quantity $\phi_k(n)$ is the angle of the kth source incident onto the array at time n, d_m for $m = 1, 2, \ldots, M$ is the position of the mth sensor, and λ is the wavelength. Denote a complex vector of observations $\boldsymbol{y}(n) \in \mathcal{C}^M$ that represents the data received by a linear array of sensors at the nth snapshot, and a complex vector of amplitudes of the sources at the nth instant by $\boldsymbol{a}(n) \in \mathcal{C}^{k(n)}$. We adopt a first order state-space Markov model in the proposed sequential sampling approach. It is assumed that the states $[\boldsymbol{\phi}(n), \boldsymbol{a}(n)]$ evolve according to the following equations

$$\boldsymbol{\phi}(n) \quad = \quad \boldsymbol{\phi}(n-1) + \sigma_w \boldsymbol{w}(n), \tag{4.4}$$

$$\boldsymbol{a}(n) \quad \sim \quad \mathcal{N}(\boldsymbol{0}, \delta^2 \sigma_w^2 \left[\boldsymbol{S}^H(\boldsymbol{\phi}(n)) \boldsymbol{S}(\boldsymbol{\phi}(n))\right]^{-1}). \tag{4.5}$$

The observation equation is defined as

$$\boldsymbol{y}(n) \quad = \quad \boldsymbol{S}(\boldsymbol{\phi}(n)) \boldsymbol{a}(n) + \sigma_v \boldsymbol{v}(n), \tag{4.6}$$

where the noise variables $\boldsymbol{v}(n) \in \mathcal{C}^M$ and $\boldsymbol{w}(n) \in \mathcal{R}^{k(n)}$ are Gaussian variables with zero mean and unit variance uncorrelated with the signal and each other, the noise variances σ_w^2 and σ_v^2 are assumed unknown and constant (for a static system), and the hyperparameter δ^2 is set to an a priori estimate of the SNR (Andrieu and Doucet, 1999).

We define the following parameter vector $\boldsymbol{\theta}_{1:n}$

$$\boldsymbol{\theta}_{1:n} \triangleq (\{\boldsymbol{\phi}_{k(n)}\}_{1:n}, \{\boldsymbol{a}_{k(n)}\}_{1:n}, k_{1:n}, \sigma_v^2, \sigma_w^2), \tag{4.7}$$

where the notation $(\cdot)_{n_1:n_2}$ denotes all values of the argument from time n_1 to n_2. Using Bayes' theorem, we have the following expression for the joint posterior distribution $\pi(\boldsymbol{\theta}_{1:n})$ from time 1 to n, involving the likelihood $l(\boldsymbol{y}_{1:n}|\boldsymbol{\theta}_{1:n})$ and the prior distribution $p(\boldsymbol{\theta}_{1:n})$

$$\pi(\boldsymbol{\theta}_{1:n}) \quad \propto \quad l(\boldsymbol{y}_{1:n}|\boldsymbol{\theta}_{1:n})p(\boldsymbol{\theta}_{1:n}), \tag{4.8}$$

The above can be expanded to give

$$\pi(\boldsymbol{\theta}_{1:n}) \propto l(\boldsymbol{y}_{1:n}|\boldsymbol{\phi}_{1:n}, \boldsymbol{a}_{1:n}, k_{1:n}, \sigma_v^2, \sigma_w^2)p(\boldsymbol{\phi}_{1:n}|k_{1:n}, \sigma_v^2)p(\boldsymbol{a}_{1:n}|\boldsymbol{\phi}_{1:n}, k_{1:n}, \sigma_w^2)p(\sigma_v^2)p(\sigma_w^2)p(k_{1:n}). \tag{4.9}$$

Assuming the observations, given the states are *iid*, the conditional update likelihoods of the states are also *iid*, and the distribution of the initial states is uniform, we can express the following individual distributions, using the Markov properties of the model, as follows

$$l(\boldsymbol{y}_{1:n}|\boldsymbol{\phi}_{1:n}, \boldsymbol{a}_{1:n}, k_{1:n}, \sigma_v^2, \sigma_w^2) = \prod_{l=1}^{n} \mathcal{N}(\boldsymbol{S}(\boldsymbol{\phi}_l)\boldsymbol{a}_l, \sigma_w^2 \boldsymbol{I}_M), \tag{4.10}$$

$$p(\boldsymbol{\phi}_{1:n}|k_{1:n}, \sigma_v^2) = \prod_{l=1}^{n} \mathcal{N}(\boldsymbol{\phi}_{l-1}, \sigma_v^2 \boldsymbol{I}_{k_l}), \tag{4.11}$$

$$p(\boldsymbol{a}_{1:n}|\boldsymbol{\phi}_{1:n}, k_{1:n}, \sigma_w^2) = \prod_{l=1}^{n} \mathcal{N}(\boldsymbol{0}, \delta^2 \sigma_w^2 \left[\boldsymbol{S}^H(\boldsymbol{\phi}_l)\boldsymbol{S}(\boldsymbol{\phi}_l)\right]^{-1}). \tag{4.12}$$

The prior distribution for the model order k is assigned a discrete uniform distribution equal to $\Lambda = 1/k_{\max}, k = 0, \ldots, k_{\max}$, where $k_{\max}$ is the maximum allowable model order, equal to $M - 1$. The prior distributions of the noise variances are both assumed to follow the inverse Gamma distribution, which is the conjugate distribution for the Normal distribution as follows

$$p(\sigma_v^2) \quad \sim \quad \mathcal{IG}(\frac{\alpha_0}{2}, \frac{\gamma_0}{2}), \tag{4.13}$$

$$p(\sigma_w^2) \quad \sim \quad \mathcal{IG}(\alpha_1, \gamma_1), \tag{4.14}$$

where $\alpha_0, \alpha_1, \gamma_0, \gamma_1$ are hyperparameters.

Substituting (4.10) – (4.14) into (4.9) and rearranging, it can be shown (Andrieu and Doucet, 1999; Larocque *et al.*, 2002) that the posterior distribution $\pi(\boldsymbol{\theta}_{1:n})$ can be represented as follows

$$\begin{aligned}\pi(\boldsymbol{\theta}_{1:n}) \propto & \prod_{l=1}^{n} \frac{1}{\sigma_w^{2k_l}\pi^{k_l}} \exp\left[\frac{-1}{\sigma_w^2}(\boldsymbol{a}_l - \boldsymbol{m}_{\boldsymbol{a}_l})^H \boldsymbol{\Sigma}_{\boldsymbol{a}_l}^{-1}(\boldsymbol{a}_l - \boldsymbol{m}_{\boldsymbol{a}_l})\right] \\ & \times \prod_{l=1}^{n} \frac{1}{\sigma_w^{2M}\delta^{2k_l}} \exp\left[\frac{-1}{\sigma_w^2}\boldsymbol{y}_l^H \boldsymbol{P}_S^{\perp}(\boldsymbol{\phi}_l)\boldsymbol{y}_l\right] \\ & \times \prod_{l=1}^{n} \frac{|\boldsymbol{S}^H(\boldsymbol{\phi}_l)\boldsymbol{S}(\boldsymbol{\phi}_l)|}{\sigma_v^{2k_l/2}(2\pi)^{k_l/2}} \exp\left[\frac{-1}{2\sigma_v^2}(\boldsymbol{\phi}_l - \boldsymbol{\phi}_{l-1})^H(\boldsymbol{\phi}_l - \boldsymbol{\phi}_{l-1})\right] \\ & \times \sigma_v^{2(-\frac{\alpha_0}{2}-1)} \exp\left[\frac{-\gamma_0}{2\sigma_v^2}\right] \times \sigma_w^{2(-\alpha_1-1)} \exp\left[\frac{-\gamma_1}{\sigma_w^2}\right] \times \prod_{l=1}^{n} \Lambda,\end{aligned} \tag{4.15}$$

where

$$\boldsymbol{\Sigma}_{\boldsymbol{a}_l}^{-1} = \boldsymbol{S}^H(\boldsymbol{\phi}_l)\boldsymbol{S}(\boldsymbol{\phi}_l)(1 + 1/\delta^2), \tag{4.16}$$

$$\boldsymbol{m}_{\boldsymbol{a}_l} = \boldsymbol{\Sigma}_{\boldsymbol{a}_l}\boldsymbol{S}^H(\boldsymbol{\phi}_l)\boldsymbol{y}_l, \tag{4.17}$$

and

$$\boldsymbol{P}_S^{\perp}(\boldsymbol{\phi}_l) = \boldsymbol{I} - \frac{\boldsymbol{S}(\boldsymbol{\phi}_l)[\boldsymbol{S}^H(\boldsymbol{\phi}_l)\boldsymbol{S}(\boldsymbol{\phi}_l)]^{-1}\boldsymbol{S}^H(\boldsymbol{\phi}_l)\boldsymbol{y}_l}{(1 + 1/\delta^2)}. \tag{4.18}$$

From (4.15), it is seen that (4.17) is a *maximum a posteriori* (MAP) estimate of the amplitudes $\boldsymbol{a}_l$, which implies that the amplitudes need not be included in the particle filter but can be estimated at each iteration after the sampling of the other parameters. Finally, integrating out $\boldsymbol{a}_l$ analytically from (4.15) yields

$$\begin{aligned}\pi(\boldsymbol{\alpha}_{1:n}) \propto & \prod_{l=1}^{n} \frac{1}{\sigma_w^{2M}(1+\delta^2)^{k_l}} \exp\left[\frac{-1}{\sigma_w^2}\boldsymbol{y}_l^H \boldsymbol{P}_S^{\perp}(\boldsymbol{\phi}_l)\boldsymbol{y}_l\right] \\ & \times \prod_{l=1}^{n} \frac{1}{\sigma_v^{2k_l/2}(2\pi)^{k_l/2}} \exp\left[\frac{-1}{2\sigma_v^2}(\boldsymbol{\phi}_l - \boldsymbol{\phi}_{l-1})^H(\boldsymbol{\phi}_l - \boldsymbol{\phi}_{l-1})\right] \\ & \times \sigma_v^{2(-\frac{\alpha_0}{2}-1)} \exp\left[\frac{-\gamma_0}{2\sigma_v^2}\right] \times \sigma_w^{2(-\alpha_1-1)} \exp\left[\frac{-\gamma_1}{\sigma_w^2}\right] \times \prod_{l=1}^{n} \Lambda,\end{aligned} \tag{4.19}$$

where the prior on the model order has been absorbed into the constant of proportionality. The above yields a much simpler posterior distribution in terms of the remaining parameters.

As a result, we can write the parameter vector $\boldsymbol{\alpha}_{1:n}$ as follows

$$\boldsymbol{\alpha}_{1:n} \triangleq (\boldsymbol{\phi}_{1:n}, k_{1:n}, \sigma_v^2, \sigma_w^2). \tag{4.20}$$

It is also possible to obtain the MAP estimators of the variance parameters as follows:

$$\sigma_{v,MAP}^2 = \frac{\frac{\gamma_0}{2} + \frac{1}{2}\sum_{l=1}^{n} \left(\boldsymbol{\phi}_l - \boldsymbol{\phi}_{l-1}\right)^H \left(\boldsymbol{\phi}_l - \boldsymbol{\phi}_{l-1}\right)}{\frac{\nu_0}{2} + \frac{1}{2}\sum_{l=1}^{n} k(l) + 1}, \tag{4.21}$$

$$\sigma_{w,MAP}^2 = \frac{\gamma_1 + \sum_{l=1}^{n} \boldsymbol{y}_l^H \boldsymbol{P}_S^{\perp}(\boldsymbol{\phi}_l)\boldsymbol{y}_l}{\nu_1 + Mn + 1}. \tag{4.22}$$

Thus, the proposed approach is to use the posterior distribution of (4.19) to estimate the DOAs $\boldsymbol{\phi}(n)$, and the model order $k(n)$, as discussed later. Once these estimates are available, the objective of this work can be achieved, which is to estimate the desired signal amplitudes. This is done using (4.17).

4.3 Sequential MC

This section describes the sequential importance sampling procedure, which is used to recursively estimate the DOAs. In this section, the background treatment on the sampling procedure is necessarily brief, and the reader is referred to Chapter 2 and other references (Andrieu *et al.*, 1999, 1998; Gordon *et al.*, 1993; Doucet, 1998) therein for a more complete coverage of this topic.

As described in Chapter 2, instead of drawing samples directly from the target posterior distribution function $\pi(\boldsymbol{x})$, we will draw samples from an importance function $q(\boldsymbol{x})$ that satisfies the recursivity condition as in (2.41) and minimizes the variance of the weights generated by (2.42) for the implementation of sequential MC procedure. According to (2.42), such an optimal importance function is given by

$$q_{\text{optimal}}(\cdot) = q(\boldsymbol{\alpha}_n^{(i)} | \boldsymbol{\alpha}_{n-1}^{(i)}, \boldsymbol{y}_n). \tag{4.23}$$

To obtain this distribution, given the state-space model of $\boldsymbol{\phi}(n)$ in (4.4) and the observation equation $\boldsymbol{y}(n)$ in (4.6), we can develop a suboptimal approximation by means of a first-order Taylor expansion of the observation equation $\boldsymbol{y}(n)$ around $\boldsymbol{\phi}(n-1)$. It is shown

in Appendix D.1 that an optimal importance function with the form as in (4.23) can be expressed as follows

$$q(\boldsymbol{\phi}_n^{(i)}|\boldsymbol{\phi}_{n-1}^{(i-1)}, \boldsymbol{y}_n) \sim \mathcal{N}(\boldsymbol{m}_n^{(i)}, \boldsymbol{\Sigma}_n^{(i)}), \tag{4.24}$$

where, for each particle,

$$\boldsymbol{\Sigma}_n^{-1} = \sigma_v^{-2}\boldsymbol{I}_{k_n} + \nabla_{\phi_n}^H \left(\sigma_w^{-2}\boldsymbol{I}_M\right) \nabla_{\phi_n}, \tag{4.25}$$

$$\boldsymbol{m}(n) = \boldsymbol{\Sigma}_n \left(\sigma_v^{-2}\boldsymbol{I}_{k_n}\boldsymbol{\phi}_{n-1} + \nabla_{\phi_n}^H \left(\sigma_w^{-2}\boldsymbol{I}_M\right) \left[\boldsymbol{y}_n - \boldsymbol{S}(\boldsymbol{\phi}_{n-1})\boldsymbol{a}_{n-1} + \nabla_{\phi_n}\boldsymbol{\phi}_{n-1}\right]\right), \tag{4.26}$$

and ∇_{ϕ_n} is the gradient vector defined as

$$\nabla_{\phi_n} = \left.\frac{\partial \boldsymbol{S}(\boldsymbol{\phi}_n)\boldsymbol{a}_n}{\partial \boldsymbol{\phi}_n}\right|_{\binom{\boldsymbol{\phi}_n=\boldsymbol{\phi}_{n-1}}{\boldsymbol{a}_n=\boldsymbol{a}_{n-1}}}. \tag{4.27}$$

Denoting the importance weight $w^{(i)}(n)$ by

$$w^{(i)}(n) = \frac{\pi(\boldsymbol{\alpha}_{1:n})}{q(\boldsymbol{\alpha}_{1:n}|\boldsymbol{y}_{1:n})}, \tag{4.28}$$

using the Markov properties of the model, the form of (4.23), and the *iid* assumptions on the noise variables, we have (Doucet, 1998)

$$w^{(i)}(n) \;=\; \tilde{w}^{(i)}(n-1) \times \frac{p(\boldsymbol{y}_n|\boldsymbol{\phi}_n^{(i)}, k_n^{(i)}, \boldsymbol{a}_n^{(i)}, \sigma_w^{2^{(i)}}) p(\boldsymbol{\phi}_n^{(i)}|\boldsymbol{\phi}_{n-1}^{(i)}, k_n^{(i)}, \sigma_v^{2^{(i)}})}{q(\boldsymbol{\phi}_n^{(i)}|\boldsymbol{\phi}_{n-1}^{(i)}, \boldsymbol{y}_n)}, \tag{4.29}$$

where $\tilde{w}^{(i)}(n-1)$ is the normalized importance weights, defined as

$$\tilde{w}^{(i)}(n-1) = \frac{w^{(i)}(n-1)}{\sum_{i=1}^{N} w^{(i)}(n-1)}. \tag{4.30}$$

As discussed in Chapter 2, the recursion of (4.29) degenerates quickly after a few iterations in such a way only a few particles are significant and lead to large estimate variance. Sampling Importance Resampling (SIR) procedure is used to resample the particles according to the significance of their respective importance weights and then the reversible jump MCMC procedure is adopted to reintroduce statistical diversity to the resampled particles as well as to facilitate the model order detection.

We summarize the sequential MC procedure in the following table

---

Sequential Importance Sampling Algorithm

Initialization

For time $n = 1$,

- sample N particles $\boldsymbol{\phi}^{(i)}, i = 1, \ldots, N$ from $q(\cdot|\cdot)$.
- initialize the weights $w^{(i)}$, $i = 1, ..., N$ to $\frac{\pi(\boldsymbol{\phi}^{(i)})}{q(\boldsymbol{\phi}^{(i)})}$.
- normalize the weights to $\tilde{w}^{(i)}(n) = \frac{w^{(i)}(n)}{\sum_{j=1}^{N} w^{(j)}(n)}$.

Then for $n = 2, 3, ...$

1. *Sequential Importance Sampling Step*

   (a) Sample N particles of $\boldsymbol{\phi}_n^{(i)}$ for $i = 1, 2, \ldots, N$ from the importance function as follows

   $$\boldsymbol{\phi}_n^{(i)} \sim q(\boldsymbol{\phi}_n^{(i)}|\boldsymbol{\phi}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n})$$

   (b) Evaluate the importance weights for N particles as follows

   $$w^{(i)}(n) = \tilde{w}^{(i)}(n-1) \times \frac{p(\boldsymbol{y}_n|\boldsymbol{\phi}_n^{(i)}, k_n^{(i)}, \boldsymbol{a}_n^{(i)}, \sigma_w^{2^{(i)}}) p(\boldsymbol{\phi}_n^{(i)}|\boldsymbol{\phi}_{n-1}^{(i)}, k_n^{(i)}, \sigma_v^{2^{(i)}})}{q(\boldsymbol{\phi}_n^{(i)}|\boldsymbol{\phi}_{n-1}^{(i)}, \boldsymbol{y}_n)},$$

   and hence the normalized importance weights as follows

   $$\tilde{w}^{(i)}(n) = \frac{w^{(i)}(n)}{\sum_{j=1}^{N} w^{(j)}(n)}$$

2. *Sampling Importance Resampling Step*

   Multiply/supress the particles $\boldsymbol{\phi}^{(i)}(n)$ respectively with high/low importance weights $\tilde{w}^{(i)}(n)$ to obtain N random samples approximately distributed according to $\pi(\boldsymbol{\phi}_{1:n}^{(i)})$.

   - Sample a vector of $\boldsymbol{l}$ distributed as:

   $$P(l(j) = i) = w^{(i)}(n)$$

- Resample the particles with the index vector:

$$\boldsymbol{\phi}_{1:k}^{(i)} = \boldsymbol{\phi}_{1:k}^{(l(i))}$$

- Re-assign all the weights to $w^{(i)}(n) = \frac{1}{N}$

3. *The Reversible Jump MCMC Step*

   Follow the MCMC procedure to be described in Section 4.4 to introduce diversity of the particles and to facilitate detection of model order.

■

4.4 The Reversible Jump MCMC Diversity Step

In this section, we present the use of the reversible jump MCMC (RJMCMC) method to recursively detect the time-varying model order $k_{1:n}$. As described in Chapter 2, the RJMCMC process (Green, 1995) samples the model order $k(n)$ from the joint posterior distribution, by jumping between subspaces of different dimensions, thus visiting all relevant model orders. The procedure is more computationally efficient, since the most likely model orders are visited most often, and hence correspondingly less effort is spent on model orders with lower probability. The RJMCMC algorithm inherently sets up a Markov chain that is capable of jumping between model orders, yet whose invariant distribution corresponds to the joint posterior $\pi(\boldsymbol{\phi}, k)$ of interest.

In the RJMCMC method, candidate samples are selected from a set of proposal distributions, which are randomly accepted according to an acceptance ratio the ensures the reversibility, and therefore the invariance of the Markov chain with respect to the desired posterior distribution. The most widely used candidate functions are the birth/death complementary moves. When the death move is selected, the algorithm proposes a candidate in the model of lower dimension, as opposed to the birth move, which proposes a candidate of higher dimension. However, if neither move is selected, the update move will be executed.

The probabilities for choosing each move are denoted u_k, b_k and d_k, respectively, such that $u_k + b_k + d_k = 1$ for all k. In this case, we choose the probability of a jump to be between 0.5 and 1 at each iteration (Green, 1995).

It is clear that once the joint posterior distribution $\pi(\boldsymbol{\alpha}(n), k(n))$ is available from the RJMCMC procedure, the model order k can be determined by marginalizing with respect to $\boldsymbol{\alpha}(n)$, leaving only $\pi(k(n))$. Then the estimate of model order $\hat{k}(n)$ can be determined using, e.g., a MAP procedure. Note that unlike the ordinary MCMC methods, the proposed method does not require "burn-in" period in this application. According to Doucet *et al.* (2000), the particles before the MCMC step are already distributed according to the limiting distribution of the chain. In other words, only one MCMC step is needed for each particle at each time step.

4.4.1 Update Move

In this move, the model order is kept fixed, i.e., $k(n) = k(n-1)$, and the candidate $\boldsymbol{\phi}^\star$ is generated according to the proposal distribution function $q(\boldsymbol{\phi}^{(i)}(n)|\boldsymbol{\phi}^{(i)}(n-1))$ given by

$$q(\boldsymbol{\phi}^{(i)}(n)|\boldsymbol{\phi}^{(i)}(n-1)) = \mathcal{N}(\boldsymbol{\phi}^{(i)}(n-1), \sigma_v^2 \boldsymbol{I}_{k(n)}). \tag{4.31}$$

Following the M-H sampling procedure described in Chapter 2, we compute the acceptance ratio of the candidate $\boldsymbol{\phi}^\star$ by substituting (4.31) and (4.19) into (2.24) to give

$$r_{update} = \frac{\exp\left[-\frac{1}{\sigma_w^2}\boldsymbol{y}^H(n)\boldsymbol{P}_S^{\perp}(\phi^\star)\boldsymbol{y}(n)\right]}{\exp\left[-\frac{1}{\sigma_w^2}\boldsymbol{y}^H(n)\boldsymbol{P}_S^{\perp}(\phi^{(i)}(n))\boldsymbol{y}(n)\right]}. \tag{4.32}$$

The candidate $\boldsymbol{\phi}^\star$ will then be accepted as the ith particle at time n with the following probability

$$\alpha_{update} = \min\{r_{update}, 1\}. \tag{4.33}$$

After all N particles have been processed in the update move, a MAP estimate of the DOAs at time n, $\hat{\boldsymbol{\phi}}(n)$, can be obtained from the histogram obtained from the particles. Using $\hat{\boldsymbol{\phi}}(n)$, we can estimate the amplitudes $\boldsymbol{a}(n)$ according to (4.17) and the noise variances σ_v^2 and σ_w^2 according to (4.21) and (4.22), respectively.

The schema for the update move is summarized as follows.

Update Move

- Propose a candidate element $\boldsymbol{\phi}^{\star}$ according to (4.31).
- Evaluate α_{update} according to (4.32) and (4.33).
- Sample $u \sim U_{[0,1]}$.
- if $(u \leq \alpha_{update})$ then
  - the state of the Markov Chain becomes $(\boldsymbol{\phi}^{\star}, k)$;
  - the amplitude $\boldsymbol{a}(n)$ are estimated according to (4.17); and
  - the variances $\sigma_w^{2(i+1)}$ and $\sigma_v^{2(i+1)}$ are updated according to (4.22) and (4.21), respectively.

  else it remains at $(\boldsymbol{\phi}^{(i)}, k)$.

4.4.2 Birth/Death moves

The procedure for executing the birth/death moves is similar to that of the update move discussed above. However, the birth move proposes a candidate in a higher dimension model, whereas the death move proposes a candidate in a lower dimension model. In the birth move, we assume that the current state is $(\boldsymbol{\phi}_k, k)$ and we wish to determine whether the next state is $\left(\boldsymbol{\phi}_{k+1}, k+1\right)$ at the next iteration. This involves the addition of a new source ϕ_c, which is proposed at random from the prior distribution for the directions of arrival such that

$$\boldsymbol{\phi}^{\star}_{k(n)+1} = [\boldsymbol{\phi}_k(n) | \phi_c] \,. \tag{4.34}$$

Then, the acceptance ratio corresponding to (4.32) for the birth move is

$$r_{birth} = \frac{\exp\left[-\frac{1}{\sigma_w^2}\boldsymbol{y}^H(n)\boldsymbol{P}_S^{\perp}(\phi_{k+1}^{\star})\boldsymbol{y}(n)\right]}{\exp\left[-\frac{1}{\sigma_w^2}\boldsymbol{y}^H(n)\boldsymbol{P}_S^{\perp}(\phi_k^{(i)})\boldsymbol{y}(n)\right]} \times \frac{1}{(1+\delta^2)(k+1)}, \tag{4.35}$$

where the $\boldsymbol{\phi}_{k+1}^{\star}$ will be accepted with a probability

$$\alpha_{birth} = \min\{r_{birth}, 1\}. \tag{4.36}$$

The underlying Markov chain must be reversible with respect to moves across subspaces of different model order, so that the desired invariant distribution of the chain with respect to model order is preserved by the MCMC algorithm. That is, the probability of moving from model order k to $k+1$ must be equal to that of moving from $k+1$ to k. Thus, there must also be the death move in which a source in the current state $\left(\boldsymbol{\phi}_{k+1}, k+1\right)$ is randomly selected to be removed such that the next state becomes $(\boldsymbol{\phi}_k, k)$ at the next iteration. In order to preserve reversibility, the acceptance ratio for the death move must be (Green, 1995)

$$r_{death} = \frac{1}{r_{birth}}, \tag{4.37}$$

and a new candidate of dimension k is accepted with probability:

$$\alpha_{death} = \min\{r_{death}, 1\}. \tag{4.38}$$

The schemas for the birth and death moves are similar to those for the update move.

Birth Move

- Propose a new element ϕ_c and a candidate,

$$\boldsymbol{\phi}_{k+1}^{(i+1)} = [\boldsymbol{\phi}_k^{(i)}, \phi_c].$$

- Evaluate α_{birth} according to (4.35) and (4.36).

- Sample $u \sim U_{[0,1]}$.

- if $(u \leq \alpha_{birth})$ then the state of the Markov Chain becomes $(\boldsymbol{\phi}_{k+1}^{(i+1)}, k+1)$, else it remains at $(\boldsymbol{\phi}_{k}^{(i)}, k)$.

Death Move

- Select randomly the jth element to form the candidate,

$$\boldsymbol{\phi}_{k}^{(i+1)} = [\boldsymbol{\phi}_{1:(j-1)}^{(i)}, \boldsymbol{\phi}_{(j+1):(k+1)}^{(i)}].$$

- Evaluate α_{death} according to (4.37) and (4.38).
- Sample $u \sim U_{[0,1]}$.
- if $(u \leq \alpha_{death})$ then the state of the Markov Chain becomes $(\boldsymbol{\phi}_{k}^{(i+1)}, k)$, else it remains at $(\boldsymbol{\phi}_{k+1}^{(i)}, k+1)$.

■

4.5 Simulation Results

The proposed PF algorithm is now applied to three sets of simulation data, generated for $k = 2$ sources with the parameter values listed in Tables 4.1 and 4.6, respectively. A uniform linear array composed of $M = 8$ elements with a half–wavelength spacing of the elements is used in all the simulations. In each experiment, $N = 300$ particles are used.

In the first experiment, the ability of the proposed method to simultaneously track and detect the number of sources as well as extract the source waveforms is demonstrated. The variance of the observation noise $\sigma_w^2(n)$ is made nonstationary, with corresponding SNR values shown in Table 4.2. In this experiment, one of the sources vanishes suddenly. The proposed particle filtering (PF) method is compared with the extended Kalman Filter (EKF) (Bar-Shalom *et al.*, 2001; Haykin, 2000), which linearizes the highly nonlinear observation model in (4.6).

Parameter	M	k	σ_v^2 (deg^2)	$\boldsymbol{a}(0)$
Value	8	2	10	$[10, 8]$

Table 4.1: Parameters of the state-space model for simulating the data for Experiment 1.

In the second experiment, the PF method is compared with the LPA-beamformer (Gershman, 1999; Katkovnik and Gershman, 2000). The LPA-beamformer has the ability to adapt to a target direction, which changes throughout the observation interval.

In the last experiment, a smoother track is used to compare the proposed method with the LPA-beamformer. This is a more favorable case for the latter method. In this case, the DOAs tracks correspond to two sinusoidal functions.

4.5.1 Experiment 1: Joint DOA tracking and detection of unknown sources

In the first experiment, the DOA tracks of the two sources correspond to autogressive (AR) processes whose coefficients represent a 10th–order low–pass Butterworth filter, with normalized cutoff frequency 0.1, variance σ_v^2, and with AR coefficients which are specified in Table 4.3. These simulated tracks are representative of a difficult but realistic tracking scenario in real life. Likewise, the source waveforms are AR processes, generated from a 10th–order low–pass butterworth filter with normalized cutoff frequency 0.3, whose coefficients are also listed in Table 4.3. Figure 4.1 depicts the locations of the roots of the AR coefficients used for the generation of DOAs and signal amplitudes, respectively. The initial value $\boldsymbol{\phi}(0)$ is $[-20°, 30°]$ and the initial SNR is about 18.56 dB for both sources. 300 snapshots are generated. One of the sources vanishes at $n = 200$. In order to show the robustness of the PF method in a nonstationary noise environment, the observation noise is generated such that the variances are varying or nonstationary in different regions of time. The average SNRs in the different time regions are summarized in Table 4.2.

In this experiment, the EKF simulation assumes the number of sources $k = 2$ is known and constant, but the PF method is capable of detecting the number of sources sequentially, and therefore does not require knowledge of k. Unlike the PF method, the EKF requires

Time range	SNR (dB)
1-100	18.56
101-200	15.32
201-300	19.78

Table 4.2: The average SNRs in the different time ranges.

AR coefficients for DOA generation	AR coefficients for signal amplitude generation
1	1
-7.9923	-3.9877
28.9122	8.0944
-62.3154	-10.4736
88.5877	9.4233
-86.7671	-6.0842
59.2810	2.8353
-27.8903	-0.9364
8.6457	0.2089
-1.5942	-0.0283
0.1328	0.0018

Table 4.3: The AR coefficients for the generation of the DOA and signal amplitude processes.

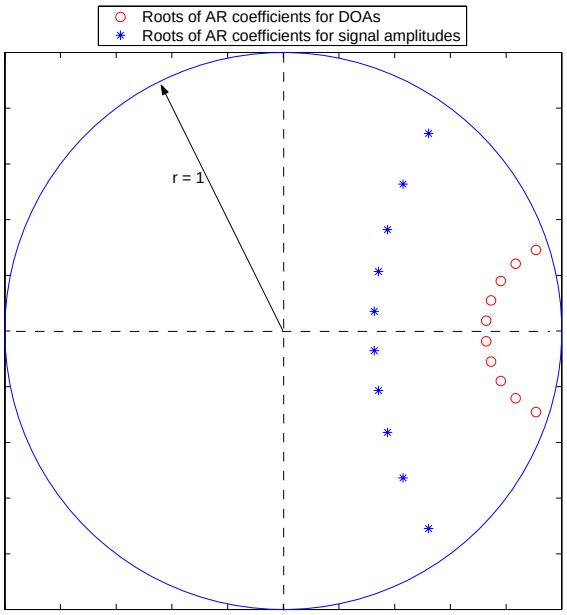

Figure 4.1: The roots of the AR processes.

the exact AR process model, and the noise covariance matrices.

The proposed algorithm randomly initializes the unknown parameters and assigns the model order $k(0)$ to $k_{\max} = M - 1 = 7$, where $k_{\max}$ is the maximum allowable model order.

First, we consider the performance of the PF approach in detecting the model order. The instantaneous model order estimate versus time is shown in Figure 4.2. We see that the detection process takes about 10 snapshots to converge to the true model order $k = 2$, and that the vanishing source is properly detected at $n = 200$.

Even though the objective of this work is to recover the amplitudes, the DOA estimation process is a critical step in determining the source waveforms. This is because the MAP amplitude estimate of (4.17) requires the ϕ's. Thus, we first consider a comparison of the DOA estimation performance of the PF method versus that of the EKF. As shown in Figure 4.3, it is seen the PF method significantly outperforms the EKF in terms of accuracy in tracking the DOAs. In the region $0 < n < 20$, the PF approach is recovering from a transient due to the tracks being initialized to the incorrect values. The relatively poor

performance of the EKF is due in part to the highly nonstatic behavior of the tracks. The performance of the PF method degrades over the interval $160 \leq n \leq 180$ where the sources are very closely spaced, as shown in Figure 4.4, which is a zoomed version of Figure 4.3, for the region $n = [151, 209]$. This behavior is expected since the matrix $\boldsymbol{S}(\boldsymbol{\phi})$ becomes poorly conditioned in this case. It is seen that after $n = 200$ when source 1 vanishes, the PF method detects the change in model order, and subsequently, produces accurate DOA estimates, as shown in Figures 4.3 and 4.4. However, since the EKF has no facility for model order detection, it assumes there are still $k = 2$ sources during this interval. As a result, because of this error in the model, the DOA estimates produced by the EKF are grossly in error.

We now consider the performance of the source waveform recovery. The performances of both approaches are directly dependent on the accuracy of the DOA estimation. Figure 4.5 shows amplitude waveforms recovered using three different methods. The first uses the PF approach to determine the DOAs at each time instant, and then uses (4.17) to generate the corresponding amplitudes. The second is similar, except it uses the EKF to generate the DOAs. In this case, the amplitudes are determined using (4.17) with $\delta^2 = 0$. The third method uses the EKF to estimate the DOAs, but then uses a conventional beamformer method (Johnson, 1982) to compute of a set of adaptive weights based on the estimated DOAs from the EKF procedure. The source waveforms $\hat{\boldsymbol{a}}(n)$ are then recovered from the output of the beamformer. By assuming that the two sources are the targets to be protected, we can compute the weights, $\boldsymbol{u}(n)$, using a constrained optimization procedure as follows

$$\begin{aligned} u_{opt}(n) = &\arg \min_{\boldsymbol{u}(n)} \boldsymbol{u}^H(n) \hat{\boldsymbol{R}}(n) \boldsymbol{u}(n) \\ &\mathbf{s.t.}\, \boldsymbol{C}^H(n) \boldsymbol{u}(n) = \boldsymbol{c}, \end{aligned} \tag{4.39}$$

where $\hat{\boldsymbol{R}}(n)$ is a sample covariance matrix, recursively updated as

$$\hat{\boldsymbol{R}}(n) = \hat{\boldsymbol{R}}(n-1) + \boldsymbol{y}(n)\boldsymbol{y}^H(n), \tag{4.40}$$

$\boldsymbol{C}(n)$ is the constraint matrix, the columns of which are the steering vectors evaluated at the estimates $\hat{\boldsymbol{\phi}}(n)$, and $\boldsymbol{c}$ is a column of constraining values. In this particular simulation,

all constraining values are set to one, that is, all targets are kept intact at unity gain. The beamformer output $\hat{\boldsymbol{a}}(n)$ is given as $\hat{\boldsymbol{a}}(n) = \boldsymbol{u}_{opt}^H(n)\boldsymbol{y}(n)$. The respective DOA estimates used by the PF and EKF approaches are shown in Figure 4.3.

Given the relatively poor performance of the EKF with regard to DOA estimation, the signal extraction for both the EKF approaches is adversely affected, as shown in Figure 4.5. It is seen that the waveforms recovered using the DOA estimates from the particle filter are significantly improved over those given using the EKF. It is only in the region $160 < n < 180$, where the DOAs are extremely close together, that the particle filter approach deteriorates. The improved behaviour of the PF method is in spite of the additional information (i.e., the covariance matrices and the model order) that are required by the EKF method. Even though the EKF is simple and relatively easy to implement, the linearization used in the EKF in the measurement and/or state prediction can introduce a bias, and the covariance computation based on a series expansion is not always accurate (Bar-Shalom *et al.*, 2001). In addition, unless accurate estimates of all other important parameters are given, the EKF will not perform as intended. Furthermore, a significant advantage of the proposed method is that it is capable of detecting the instantaneous number of sources, which is crucial in signal extraction, especially when the number of sources is time-varying.

4.5.2 Experiment 2: Comparison with the LPA beamformer

In the second experiment, the PF method is compared with the LPA-beamformer (Gershman, 1999; Katkovnik and Gershman, 2000). The DOA tracks and the two source waveforms are generated as in experiment 1, with initial values $\boldsymbol{\phi}(0) = [0°, -20°]$, and $k(0) = k_{\max}$, and with the other parameters listed as before in Table 4.1. Here, the DOA tracks cross each other at about $n = 140$, and the model order remains fixed at $k = 2$ throughout the entire observation interval. In this experiment, the observation noise variance is constant throughout the observation interval. The signal amplitudes are extracted using (4.17) directly for the PF method. For the LPA method, the sources are extracted using the beamformer approach described by (4.39) and (4.40).

Figure 4.6 shows the convergence of the model order estimate from its initial value of

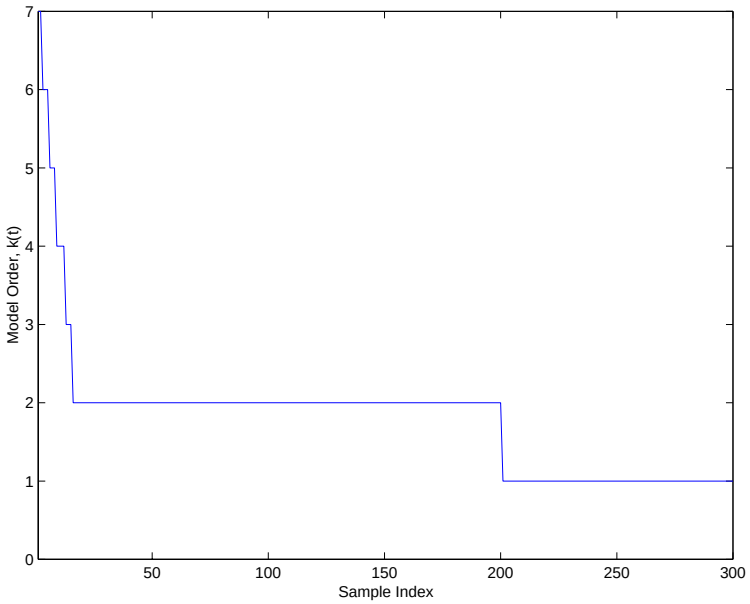

Figure 4.2: Sequential estimates of model order for Experiment 1.

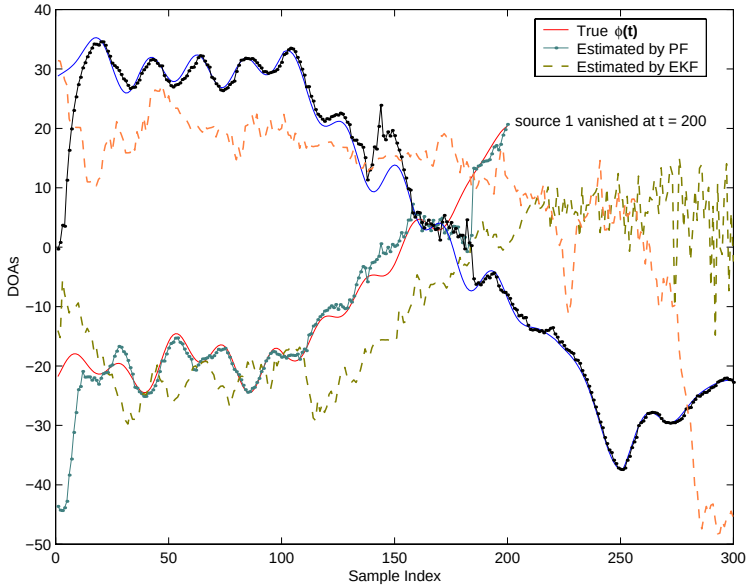

Figure 4.3: Comparison of DOA tracking performance for the PF and the EKF methods, for Experiment 1.

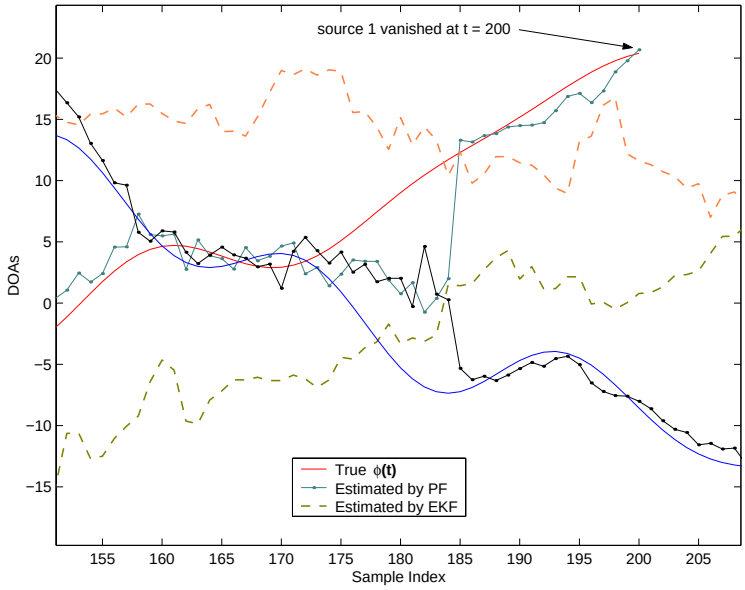

Figure 4.4: A zoomed version of Figure 4.3, from sample $n = 151$ to $n = 209$, highlighting the region where the source disappears.

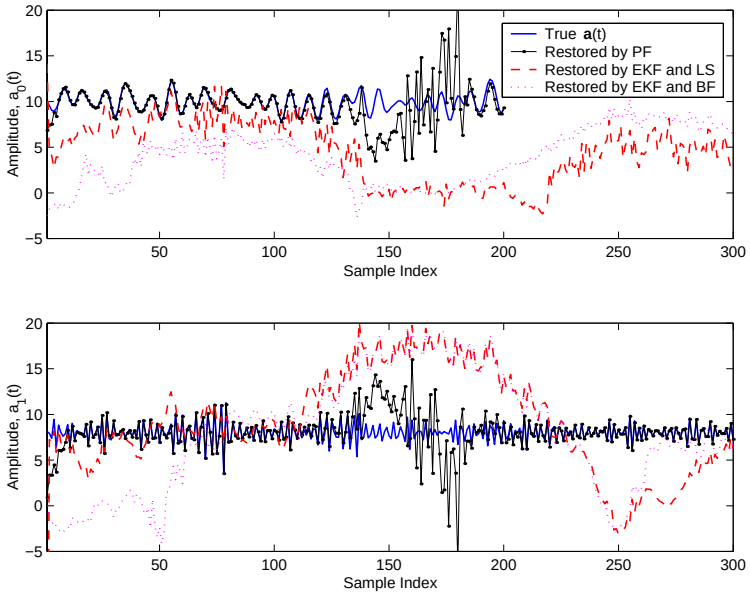

Figure 4.5: Comparison of the waveform extraction performance of the PF method and the LS and beamforming approaches based on the EKF, for Experiment 1. Top: recovered source waveform for source 0, Bottom: same for source 1.

$k_{\max}$ to the true value $k = 2$. As seen, it takes about 10 snapshots for the PF model order detection scheme to converge to the correct value. The performance for both the DOA and waveform estimation processes for the PF and the LPA methods are shown in Figures 4.7 and 4.8, respectively. It is seen that the observations concerning the relative performance of the PF vs. EKF approaches in Experiment 1 also persist with respect to the comparison between the PF and LPA approaches.

The LPA-beamformer technique was developed to relax the assumption that DOAs are constant within the entire observation period, as is required with conventional beamformers. It accommodates DOA variation within a sliding time window by modelling the DOA trajectory vs. time as a polynomial, whose coefficients must be estimated. Like the EKF, the LPA-beamformer assumes the number of sources is known and constant throughout the entire observation period. The length of the sliding window for the LPA-beamformer for this experiment is $L = 15$, and a first-order polynomial is used to track the changing DOAs within the window[1].

As shown in Figure 4.7, the estimated tracks produced by the LPA-beamformer are not close to the true tracks until they are well-separated and relatively smooth, as is the case for $n \geq 150$. The degraded performance of the LPA-beamformer relative to the PF approach can be explained by the fact that the DOA motion within the sliding window violates the assumption of smooth and linear behaviour. Since the performance of the LPA-beamformer is somewhat degraded with regard to DOA estimation, the signal extraction is also adversely affected, as shown in Figure 4.8.

Tables 4.4 and 4.5 provide a quantitative measure of the performance of the various algorithms in terms of the MSE of the estimated DOAs and source waveforms. As expected, the MSE for the DOA and amplitude estimates in the periods $n \in [1, 114]$ and $n \in [251, 300]$ are small, respectively, for the PF method. In contrast, the errors are comparatively larger for the LPA approaches. During $n \in [115, 250]$ when the tracks are very close, the DOA estimation MSE on average is far better than that for amplitude estimation for the PF

[1]The first-order polynomial was chosen because it gave the best performance. Higher-order polynomials have the potential to track changing directions better, but their performance degrades because more coefficients must be estimated.

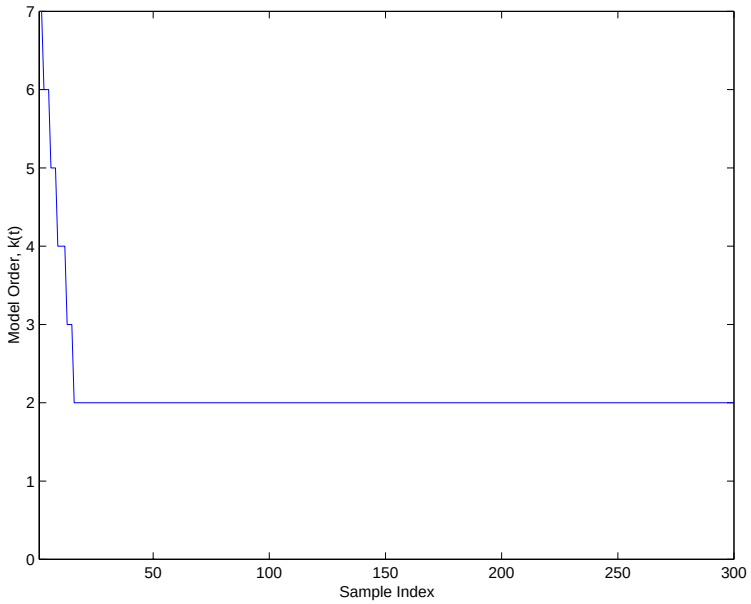

Figure 4.6: Sequential estimates of model order for Experiment 2, using the PF method.

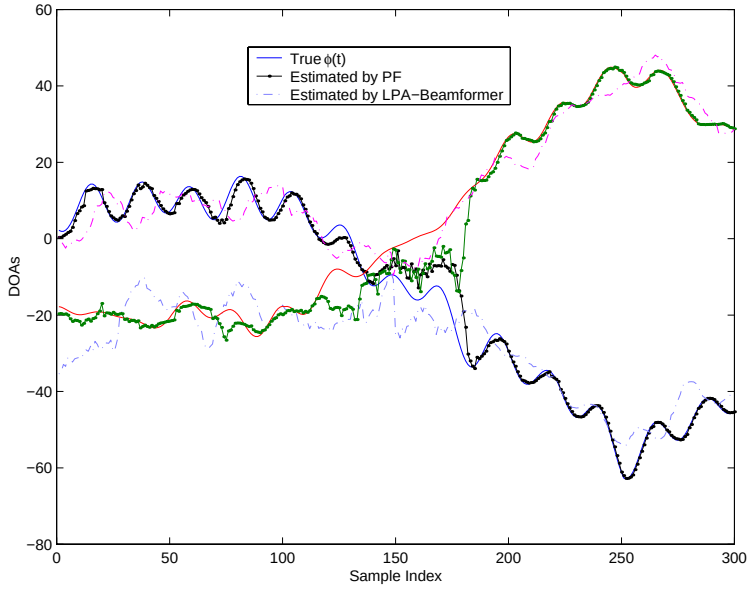

Figure 4.7: Comparison of the DOA tracking performance for the PF method and the LPA approach for Experiment 2.

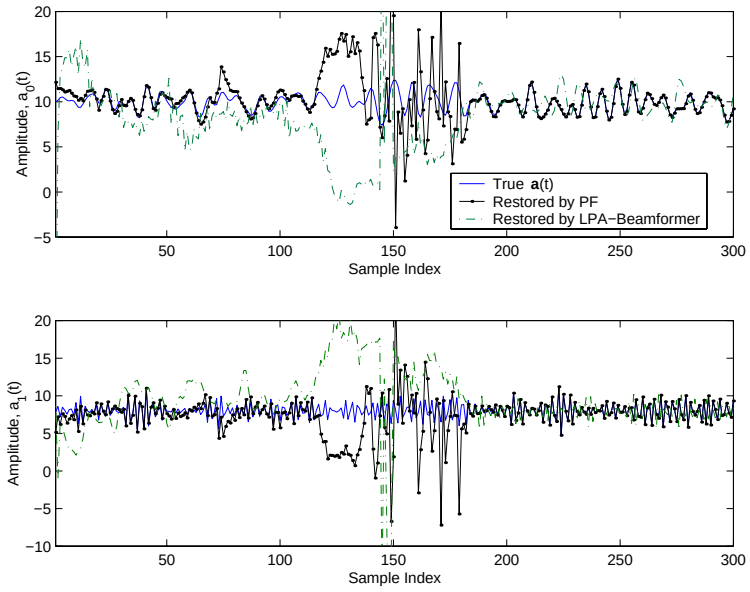

Figure 4.8: Comparison of the waveform extraction performance for the PF method and the LPA approach for Experiment 2.

method, implying that amplitude estimation is very sensitive to DOA estimation, when the DOA tracks are close. Nevertheless, these errors are still far smaller for the PF method than those produced by the other approaches.

We now investigate the effect of the number of particles N on the performance of the proposed PF method. Table 4.8 compares the MSE between the true and estimated DOAs for different values of N using the experimental configuration of Experiment 2. As can be seen, the algorithm does not improve significantly as N increases from 300 to 500, whereas the computational load is increased by approximately 60%. However, we notice that performance does deteriorate as N drops to 100 particles. In other words, one can tradeoff the performance of the algorithm and the complexity of the implementation by adjusting the number of particles N. The use of the optimal importance function in (4.31) is effective in reducing the required number of particles.

MSE (dB)	$n \in [1, 114]$	$n \in [115, 250]$	$n \in [251, 300]$
PF	-27.4, -30.3	-10.3, -18.2	-33.0, -40.5
LPA-beamformer	-20.0, -18.2	-8.2, -6.1	-24.9, -32.9

Table 4.4: The MSE between the true and estimated DOAs over 50 independent trials for the PF method and the LPA-beamformer, for Experiment 2.

MSE (dB)	$n \in [1, 114]$	$n \in [115, 250]$	$n \in [251, 300]$
PF	-17.3, -18.0	-5.6, -6.5	-24.3, -26.6
LPA-beamformer	10.8, -11.6	8.5, 13.7	-17.9, -19.0

Table 4.5: The MSE between the true and estimated amplitudes over 50 independent trials for the PF and beamforming approaches, for Experiment 2.

4.5.3 Experiment 3: Comparison between the PF method and the LPA-beamformer

In this experiment, the two DOA tracks are smooth sinusoidal functions, given by

$$\boldsymbol{\phi}(n) = \boldsymbol{\phi}(0) + 10 \sin\left(\frac{10\pi n}{300}\right), \quad n = 1, ..., 300,$$

where $\boldsymbol{\phi}(0) = (20, -20)^\circ$, are used to compare the tracking performance and hence the signal extraction of the proposed method with that of the LPA-beamformer. The amplitudes are generated using AR processes as in the previous experiments. The number of sources is assumed known and constant. The other parameters used in this simulation can be found in Table 4.6.

Figures 4.9 and 4.10 show the comparison between the proposed method and the LPA-beamformer in terms of DOA tracking and signal extraction, respectively. According to these figures, these methods provide comparable performance in both respects, although

Parameter	M	K	$SNR(dB)$	$\boldsymbol{\phi}(0)$	$\boldsymbol{a}(0)$
Value	8	2	15	$[-20^\circ, 20^\circ]$	$[8, 10]$

Table 4.6: Parameters of the state-space model for simulating the data for Experiment 3.

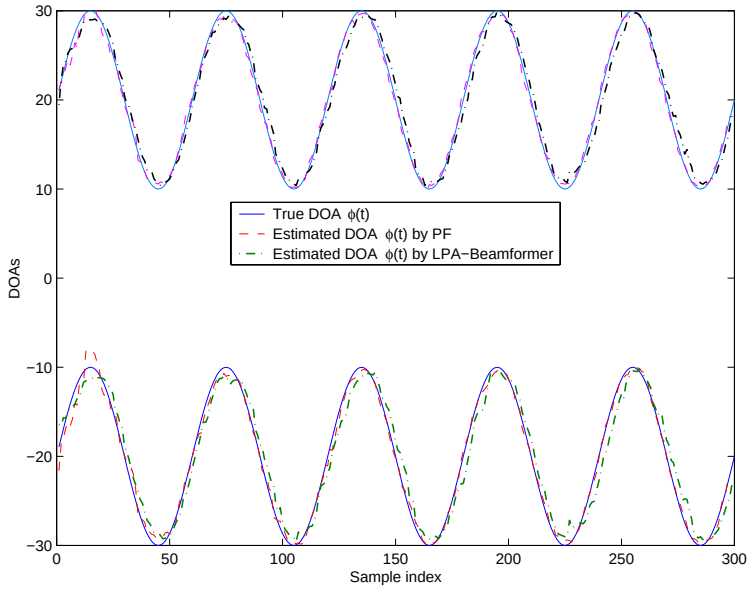

Figure 4.9: Comparison of the DOA tracking performance for the PF method and the LPA-beamformer.

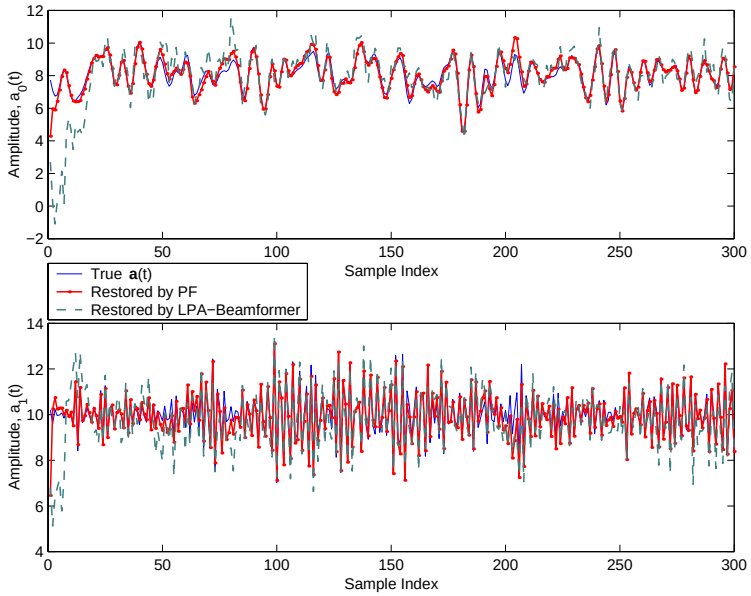

Figure 4.10: Comparison of the waveform extraction process for the PF method and the LPA beamformer.

MSE (dB)	MSE for DOA Estimation (dB)	MSE for amplitude Estimation (dB)
PF	-24.2, -26.0	-16.6, -18.4
LPA-beamformer	-21.3, -24.3	-14.0, -17.6

Table 4.7: The MSE for DOA estimation and signal extraction for the PF and the LPA-beamformer approaches, over 50 independent trials for Experiment 3.

MSE (dB)	$n \in [1, 114]$	$n \in [115, 250]$	$n \in [251, 300]$
N = 500	-28.4, -31.8	-10.9, -19.1	-34.0, -43.4
N = 300	-27.4, -30.3	-10.3, -18.2	-33.0, -40.5
N = 100	-20.9, -25.0	-9.9, -14.6	-28.2, -34.9

Table 4.8: The MSE between the true and estimated DOAs by PF for 50 independent trials for different number of particles N for Experiment 2.

the PF method is slightly better. Table 4.7 summarizes the MSE of the DOA estimation and signal extraction for these methods over 50 independent trials.

The LPA-beamformer, which relies on a first-order linear model to track slowly fluctuating DOAs, suffers from model mismatch in highly nonstatic conditions. If the assumptions for the LPA-beamformer are violated, i.e., the tracks are not smooth and moderately nonstatic, the LPA-beamformer performs poorly with regard to DOA tracking, and hence in extracting signals. However, we have seen that the PF method is capable of DOA tracking and source extraction in highly nonstatic situations. On the other hand, if all conditions favor the LPA-beamformer, as in this experiment, then it is seen that both methods have comparable performance, and the LPA-beamformer is then preferred over the PF method because it is less computationally intensive.

4.5.4 Discussion

In addition to the reasons previously cited, the superior performance of the PF method is in part, through (4.29), due to the availability of an approximate *instantaneous* joint posterior distribution of all the relevant parameters at each time instant. This is in contrast to

previous methods which are based on accumulating statistics by time averaging. These previous techniques assume the underlying process is static over an adequate time interval. This leads to degradation in performance in situations where the static assumption is no longer valid. On the other hand, the PF method does not need to accumulate statistics by time averaging, since it has the entire posterior distribution available at each instant of time.

A further advantage of the proposed PF method is that confidence regions of the parameter estimates can be easily evaluated. This is a consequence of the fact that an approximation to the posterior distribution is available at each time instant. An example of a posterior distribution of the DOA estimates, from which confidence regions can be obtained, corresponding arbitrarily to the 245th DOA sample in experiment 2 (see Figure 4.7), is shown in Figure 4.11.

Note that once the $\boldsymbol{\phi}$'s have been estimated, the waveforms could also be obtained using a Kalman filter. The proposed MAP procedure and the Kalman filter should give almost equivalent performance, since the MAP method is optimum in a Bayesian sense, while the Kalman filter is optimum in a least-squares sense.

It is easily shown that as $N \to \infty$, where N is the number of particles, the global optimum of the desired posterior distribution coincides with the most heavy-weighted histogram bin corresponding to the particles. In practice, the global optimum is achieved within a histogram bin-width with finite N with high probability. Thus, the global optimum can be attained by a simple search, instead of a complicated global optimization over a multi-modal surface.

We now briefly discuss the relative computational requirements of the EKF, the LPA-beamformer, and the PF algorithms. The EKF is $\mathcal{O}(M^2)$ (Bar-Shalom *et al.*, 2001), whereas the LPA-beamformer is $\mathcal{O}(LMQP)$ (Katkovnik and Gershman, 2002), where L is the number of observations used in the sliding window, and Q and P respectively are the number of points in the angular and angular velocity domains. On the other hand, the computational requirements for the particle filter are dominated by the evaluation of the posterior density (4.19) for each particle. This is approximately $\mathcal{O}(N\big[(K^3 + MK^2 + M^2K)\big])$, which

is higher than that of the other two methods. However, the evaluation of the particles is easily parallelizable, and this figure includes the expense of order detection, which is not included in the cited costs of the other methods. Furthermore, the relative computational expense of the method is offset by its advantages; namely, a joint detection capability and improved performance in nonstatic environments.

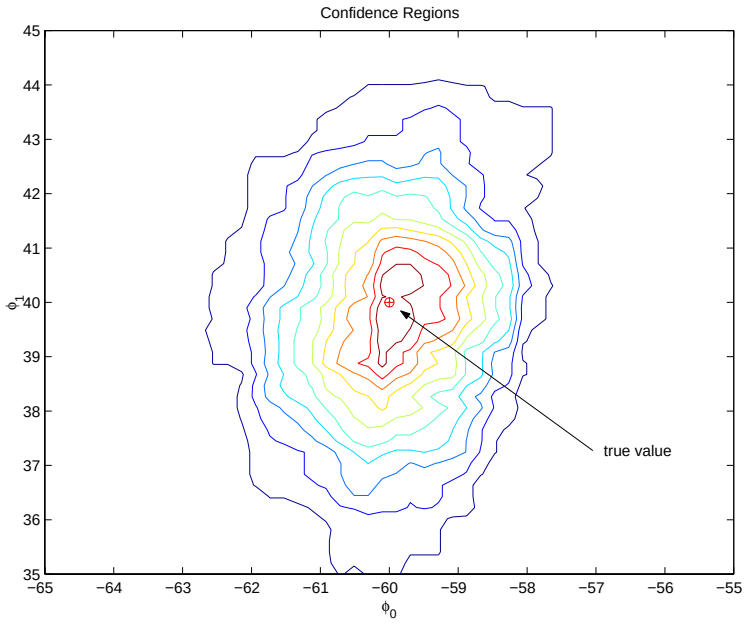

Figure 4.11: Contours of the approximate posterior distribution of the DOAs with true values at $[-60^o, 40^o]$, corresponding to time $n = 245$ of Fig. 4.7.

4.6 Conclusion

In this chapter, we present an alternative to the classical beamforming approach. This approach involves angle-only estimation and detection of the number of sources, for recovering and tracking multiple desired signals in nonstatic environments in the presence of interference. The proposed method implements sequential Markov Chain Monte Carlo (MCMC) estimation, also known as *particle filtering*. Unlike the traditional beamforming methods, the proposed method is able to track and recover source waveforms from highly nonstatic

sources.

In contrast to other methods like EKF and the LPA-beamformer, the proposed method is more robust to nonlinearities in the data model and to nonstatic environment. Simulation results show that the proposed method has superior performance in terms of DOA tracking ability and signal recovery, relative to the other methods.

Chapter 5

Wideband Array Signal Processing I

This chapter applies the novel data model presented in Chapter 3 to wideband array signal processing. The new interpolation model where the observations are linear functions of the source amplitudes, but nonlinear in the direction of arrival (DOA) parameters, is formed. The interpolation model also applies to the narrowband case. The proposed method lends itself well to a Bayesian approach for jointly estimating the model order and the DOAs through a reversible jump Markov chain Monte Carlo (MCMC) procedure. The source amplitudes are estimated through a *maximum a posteriori* (MAP) process. Advantages of the proposed method include joint detection of model order and estimation of the DOA parameters, the fact that reliable performance can be obtained using significantly fewer observations than previous wideband methods, and that only real arithmetic is required. The DOA estimation performance of the proposed method is compared with the theoretical Cramér-Rao lower bound (CRLB) for this problem. Simulation results demonstrate the effectiveness and robustness of the method.

5.1 Introduction

In Chapter 1, several wideband array processing methods, including the Coherent Signal-subspace Method (CSM) and the Two-sided Correlation Transformation (TCT) approaches, were discussed in some detail. The deficiencies of these methods were mentioned. We note that previously, separate models and analysis methods were required for the narrowband and wideband scenarios. In this chapter, the novel model structure presented in Chapter 3, which applies equally well to both these cases, detects model order, estimates DOA, and recovers the source waveforms in a computationally efficient manner. Markov chain Monte Carlo (MCMC) (Andrieu and Doucet, 1999; Andrieu *et al.*, 1998, 2001; Gilks *et al.*, 1998) methods are well suited for extraction of the parameters of interest associated with this model. The approach proposed in this chapter is an extension of the method of Andrieu and Doucet (1999) to seamlessly perform joint detection of the number of sources and estimation of ISDs (DOAs), and recovery of the sources, for both narrowband and wideband models.

The proposed approach offers the functionality of wideband beamforming Johnson (1982) and Veen and Buckley (1988), as well as wideband DOA estimation and detection. The proposed algorithm can be used in a beamforming context because it recovers the source signals which are mixed at the sensor outputs. The proposed method is somewhat more computationally expensive than previous wideband methods (e.g., Valaee and Kabal (1995)) which estimate the DOAs only. However, in cases where the model order, the DOAs, and the source waveforms are all required, the proposed method shows a comparable computational expense relative to the other methods.

There are several advantages offered by this approach in array signal processing. Firstly, by virtue of the *reversible jump Metropolis-Hastings algorithm* (Green, 1995), the proposed MCMC approach *jointly* detects the model order and estimates all parameters of interest for both wideband and narrowband scenarios. This procedure is more efficient and accurate than other common approaches that would perform each process independently. Secondly, unlike other approaches, like CSM and TCT, the proposed approach requires far fewer observations for processing, because the correlation matrix of the observations is

not explicitly required. Thirdly, the method benefits from requiring only real arithmetic, in contrast to previous methods which require complex arithmetic. This fact results in significant reductions in hardware complexity, since the need for quadrature mixing to IF frequencies is alleviated. Also, like other wideband methods, the sources can be partially or fully correlated.

5.2 Data Model

The data model we consider here has been described in Chapter 3. For the convenience of the further presentation, we briefly summarize the model here. Let $\boldsymbol{y}(n) \in \mathcal{R}^{M\times 1}$ be the data received by a uniform linear array of M sensors at the nth snapshot. The data vector is composed of K wideband signals impinging onto the array at K distinct incident angles. Let $s_k(n)$ be the kth source that is bandlimited according to (3.1) and transverses along the array with an inter-sensor delay τ_k defined as in (3.2). Following the definitions in Chapter 3, we can express the nth snapshot $\boldsymbol{y}(n)$ as follows

$$\boldsymbol{y}(n) = \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l) + \sigma_w \boldsymbol{w}(n), \quad n = 1, ..., N_t, \tag{5.1}$$

where $\boldsymbol{\tau} \in \mathcal{R}^{K\times 1}$ and $\boldsymbol{a}(n) \in \mathcal{R}^{K\times 1}$ are defined in (3.29) and (3.37), respectively, $\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}) \in \mathcal{R}^{M\times K}$ is defined in (3.39), $\boldsymbol{w}(n)$ is an *iid* zero-mean, unit-variance Gaussian random variable, defined as

$$\boldsymbol{w}(n) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_M), \tag{5.2}$$

σ_w^2 is the observation noise variance, L is the number of taps used in the interpolation, and N_t is the number of snapshots. Defining $\boldsymbol{z}(n) \in \mathcal{R}^{M\times 1}$

$$\boldsymbol{z}(n) = \boldsymbol{y}(n) - \sum_{l=1}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l), \quad n = 1, ..., N_t, \tag{5.3}$$

as described in Chapter 3 yields the desired model

$$\boldsymbol{z}(n) = \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n) + \sigma_w \boldsymbol{w}(n), \quad n = 1, ..., N_t. \tag{5.4}$$

The features of the desired model $\boldsymbol{z}(n)$ can be found in Section 3.4.

5.3 Development of The Marginal Posterior Distribution

Here, it is assumed 1) that the noise vectors $\boldsymbol{w}(n)$ are *iid*, and 2) that all the parameters describing the received signal are stationary throughout the entire observation interval. Using (5.4), we may define a set of N_t snapshots as

$$\boldsymbol{Z} = [\boldsymbol{z}(1), \ldots, \boldsymbol{z}(N_t)]\,. \tag{5.5}$$

Then, the total likelihood function of all data can be expressed as follows

$$\ell(\boldsymbol{Z}|\boldsymbol{a}(\cdot), \boldsymbol{\tau}, \sigma_w^2, k) = \prod_{n=1}^{N_t} \frac{1}{(2\pi\sigma_w^2)^M} \exp\left\{ \frac{-1}{2\sigma_w^2} \left(\boldsymbol{z}(n) - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n)\right)^T \left(\boldsymbol{z}(n) - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n)\right) \right\}, \tag{5.6}$$

where k represents an estimate of the true number of sources, K. The desired posterior distribution function can be expressed using Bayes' theorem in terms of the total likelihood function (5.6) and the prior distributions of the unknown parameters as

$$\pi(\boldsymbol{a}(\cdot), \boldsymbol{\tau}, \sigma_w^2, k|\boldsymbol{Z}) \propto p(\boldsymbol{Z}|\boldsymbol{a}(\cdot), \boldsymbol{\tau}, \sigma_w^2, k) p(\boldsymbol{a}(\cdot)|k, \boldsymbol{\tau}, \delta^2\sigma_w^2) p(\boldsymbol{\tau}|k) p(\sigma_w^2) p(k), \tag{5.7}$$

where δ^2 is a hyperparameter closely related to the signal–to–noise ratio. (The choice of this parameter is discussed further in Section 5.4.4) The prior distribution for the source amplitude vector $\boldsymbol{a}$ is chosen as in Andrieu and Doucet (1999). For a single sample $\boldsymbol{a}(n)$, we take the prior distribution to be zero–mean Gaussian, with covariance matrix corresponding to the *maximum entropy* prior

$$p(\boldsymbol{a}|k, \boldsymbol{\tau}, \delta^2\sigma_w^2) = \mathcal{N}\left(0, \delta^2\sigma_w^2 \left[\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\right]^{-1}\right). \tag{5.8}$$

The joint prior in (5.7) of the source samples over N_t snapshots is then given as

$$p(\boldsymbol{a}(1)\ldots, \boldsymbol{a}(N_t)|k, \boldsymbol{\tau}, \delta^2\sigma_w^2) = \prod_{n=1}^{N_t} \mathcal{N}\left(0, \delta^2\sigma_w^2 \left[\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\right]^{-1}\right), \tag{5.9}$$

where, for analytical convenience, we have assumed the sources to be temporally *iid*. The prior distribution of $\boldsymbol{\tau}$ is chosen to be uniform:

$$p(\boldsymbol{\tau}|k) = \mathcal{U}\left[-T_{max}, T_{max}\right]^k. \tag{5.10}$$

The prior for the parameter σ_w^2 is chosen as the inverse-Gamma distribution (Andrieu and Doucet, 1999), which is the conjugate prior corresponding to a Gaussian likelihood function. It is defined as

$$p(\sigma_w^2) = \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\gamma_0}{2}\right) = \left(\sigma_w^2\right)^{-\frac{\nu_0}{2}-1} \exp\left\{\frac{-\gamma_0}{2\sigma_w^2}\right\}. \tag{5.11}$$

Note that this distribution is noninformative when $\gamma_0 = \nu_0 = 0$.[1] Finally, the prior distribution on k is chosen to be Poisson with parameter Ξ

$$p(k) = \frac{\Xi^k}{k!} \exp\left(-\Xi\right), \tag{5.12}$$

where Ξ is the expected number of sources. This choice of prior is not strictly noninformative, but it contributes to a more efficient MCMC sampling routine. As a result, we can rewrite (5.7) as

$$\begin{aligned}
\pi\Big(\boldsymbol{a}, \boldsymbol{\tau}, &\sigma_w^2, k | \boldsymbol{Z}\Big) \propto \\
&\frac{1}{(2\pi\sigma_w^2)^{MN_t/2}} \exp\left\{\frac{-1}{2\sigma_w^2} \sum_{n=1}^{N_t} \Big(\boldsymbol{z}(n) - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n)\Big)^T \Big(\boldsymbol{z}(n) - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n)\Big)\right\} \times \\
&\frac{|\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})|^{(N_t+L-1)/2}}{(2\pi\delta^2\sigma_w^2)^{(N_t+L-1)k/2}} \exp\left\{\frac{-1}{2\delta^2\sigma_w^2} \sum_{n=1}^{N_t} \boldsymbol{a}^T(n)\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n)\right\} \times \\
&\left(\frac{1}{2T_{max}}\right)^k \times \frac{\Xi^k}{k!} \exp\left(-\Xi\right) \times \left(\sigma_w^2\right)^{-\frac{\nu_0}{2}-1} \exp\left\{\frac{-\gamma_0}{2\sigma_w^2}\right\}.
\end{aligned} \tag{5.13}$$

5.3.1 Simplification of the posterior distribution function

We can simplify the estimation of the parameters in the posterior distribution function (5.13) by considering $\boldsymbol{a}$ and σ_w to be nuisance parameters, and analytically integrating them out. The only quantities of interest at this stage are $\boldsymbol{\tau}$ and k. We recover the $\boldsymbol{a}$ later by other means. By following procedures similar to those in Andrieu and Doucet (1999)

[1] Strictly speaking, this choice for γ_0 and ν_0 results in an improper prior distribution. Regardless, the resulting posterior distribution still has a well-defined maximum, which is used for MAP estimation of the parameters of interest.

and Andrieu *et al.* (1998), it can be shown that the desired posterior distribution function in (5.13) can be expressed as

$$\begin{aligned}\pi(\boldsymbol{a},\boldsymbol{\tau},&\sigma_w^2,k|\boldsymbol{Z}) \propto \\ &\frac{1}{(2\pi\sigma_w^2)^{MN_t/2}}\exp\left\{\frac{-1}{2\sigma_w^2}\sum_{n=1}^{N_t}\boldsymbol{z}^T(n)\tilde{\boldsymbol{P}}_{H_0}^{\perp}(\boldsymbol{\tau})\boldsymbol{z}(n)\right\}\times \\ &\frac{|\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})|^{N_t/2}}{(2\pi\delta^2\sigma_w^2)^{N_tk/2}}\exp\left\{\frac{-1}{2\sigma_w^2}\sum_{n=1}^{N_t}\left(\boldsymbol{a}(n)-\boldsymbol{m}_a(n)\right)^T\tilde{\boldsymbol{\Sigma}}_{H_0}^{-1}(\boldsymbol{\tau})\left(\boldsymbol{a}(n)-\boldsymbol{m}_a(n)\right)\right\}\times \\ &\left(\frac{\Xi}{2T_{max}}\right)^k\times\frac{1}{k!}\exp\left(-\Xi\right)\times\left(\sigma_w^2\right)^{-\frac{\nu_0}{2}-1}\exp\left\{\frac{-\gamma_0}{2\sigma_w^2}\right\}.\end{aligned} \tag{5.14}$$

where

$$\tilde{\boldsymbol{\Sigma}}_{H_0}^{-1}(\boldsymbol{\tau}) = (1+\delta^{-2})\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}), \tag{5.15}$$

$$\tilde{\boldsymbol{P}}_{H_0}^{\perp}(\boldsymbol{\tau}) = \boldsymbol{I} - \frac{\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\left[\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\right]^{-1}\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})}{(1+\delta^{-2})}, \tag{5.16}$$

and

$$\begin{aligned}\boldsymbol{m}_a(n) &= \tilde{\boldsymbol{\Sigma}}_{H_0}(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\boldsymbol{z}(n), \\ &= \tilde{\boldsymbol{\Sigma}}_{H_0}(\boldsymbol{\tau})\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau})\left(\boldsymbol{y}(n)-\sum_{l=1}^{L-1}\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l)\right).\end{aligned} \tag{5.17}$$

From (5.17) and (5.14) a *maximum a posteriori* estimate of the amplitudes, given the other parameters is readily available as

$$\hat{\boldsymbol{a}}_{MAP}(n) \triangleq \boldsymbol{m}_a(n). \tag{5.18}$$

We can simplify (5.14) by analytically integrating out the nuisance parameters $\boldsymbol{a}(n)$ and σ_w^2 as follows

$$\pi(\boldsymbol{\tau},k|\boldsymbol{Z}) \propto \int_0^{\infty}\int_{-\infty}^{\infty} p(\boldsymbol{a},\boldsymbol{\tau},\sigma_w^2,k|\boldsymbol{Z})d\boldsymbol{a}\,d\sigma_w^2. \tag{5.19}$$

As a result, the desired posterior distribution can now be expressed as (Andrieu and Doucet, 1999)

$$\pi(\boldsymbol{\tau}, k|\boldsymbol{Z}) \propto \frac{1}{(1+\delta^2)^{N_t k/2}} \left(\frac{\Xi}{2T_{max}}\right)^k \frac{\exp(-\Xi)}{k!} \left(\gamma_0 + \mathrm{tr}\left(\tilde{\boldsymbol{P}}^{\perp}_{\mathrm{H}_0}(\boldsymbol{\tau})\hat{\boldsymbol{R}}_{\mathrm{zz}}\right)\right)^{-\left(\frac{MN_t+\nu_0}{2}\right)}, \quad (5.20)$$

where $\mathrm{tr}(\cdot)$ is the trace operator, and

$$\mathrm{tr}\left(\tilde{\boldsymbol{P}}^{\perp}_{\mathrm{H}_0}(\boldsymbol{\tau})\hat{\boldsymbol{R}}_{\mathrm{zz}}\right) = \sum_{n=1}^{N_t} \boldsymbol{z}^T(n)\tilde{\boldsymbol{P}}^{\perp}_{H_0}(\boldsymbol{\tau})\boldsymbol{z}(n), \quad (5.21)$$

and $\hat{\boldsymbol{R}}_{zz}$ is the sample covariance matrix of $\boldsymbol{z}(n)$

$$\hat{\boldsymbol{R}}_{zz} = \sum_{n=1}^{N_t} \boldsymbol{z}(n)\boldsymbol{z}^T(n). \quad (5.22)$$

The primary goal of this work is to estimate the delay parameters $\boldsymbol{\tau}$ and the model order k using the distribution of (5.20). Because there are no direct methods for estimating model order for this problem, and because of the intractable form of this distribution, MCMC methods (Andrieu and Doucet, 1999; Gilks *et al.*, 1998; Ruanaidh and Fitzgerald, 1996), as described in Chapter 2, are well-suited for this task.

The proposed MCMC methods require evaluation of the posterior distribution (5.20) for a proposed value of $\boldsymbol{\tau}$ and k. This involves evaluation of the quantity $\hat{\boldsymbol{R}}_{zz}$, which in turn requires knowledge of the source amplitudes $\boldsymbol{a}(n)$. The amplitudes can in principle be determined through a least–squares procedure using (5.4), or through (5.18). However, the use of (5.4) involves only the matrix $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})$ (instead of all the matrices $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}), \ldots, \tilde{\boldsymbol{H}}_{L-1}(\boldsymbol{\tau})$). For typical interpolation functions, only a few elements of each of the columns of $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})$ will be significantly different from zero, resulting in a portion of the observation vector being suppressed, with a corresponding loss of performance. Further, use of (5.18) also implicitly involves only the matrix $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})$. This is because the term within the parentheses in (5.17) is equal to $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n) + \sigma_w\boldsymbol{w}(n)$. All other terms in (5.18) only involve $\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})$. Thus, use of (5.18) also results in a degradation in performance. This undesired situation can be mitigated using the following suboptimal procedure for estimating the source amplitudes.

5.3.2 Signal Recovery

According to the signal model, it is clear that the source sample $\boldsymbol{a}(n-L+1)$ contributes to L successive snapshots $\boldsymbol{y}(n-L+1)$ to $\boldsymbol{y}(n)$. Efficient estimation of the source sample $\boldsymbol{a}(n-L+1)$ requires use of all these snapshots. From (5.1), we have

$$\boldsymbol{y}(n) = \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l) + \sigma_w \boldsymbol{w}(n), \quad n = 1, .., N_t. \tag{5.23}$$

The natural log of the posterior in (5.13) can therefore be written as

$$\begin{aligned} L(\boldsymbol{a}, \boldsymbol{\tau}, \sigma_w^2, k|\boldsymbol{Z}) \propto \\ \kappa - \frac{1}{2\sigma_w^2} \sum_{n=1}^{N_t} \left(\boldsymbol{y}(n) - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l) \right)^T \left(\boldsymbol{y}(n) - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l) \right) \\ - \frac{\delta^{-2}}{2\sigma_w^2} \sum_{n=1}^{N_t} \boldsymbol{a}^T(n) \tilde{\boldsymbol{H}}_l^T(\boldsymbol{\tau}) \tilde{\boldsymbol{H}}_l(\tau) \boldsymbol{a}(n), \end{aligned}$$

where κ is a constant independent of $\boldsymbol{a}(n)$. The desired estimate $\hat{\boldsymbol{a}}(n-L+1)$ is then obtained by maximizing (5.24) with respect to $\boldsymbol{a}(n-L+1)$. The result is

$$\hat{\boldsymbol{a}}(n-L+1) = (1+\delta^{-2})^{-1} \left[\tilde{\boldsymbol{\mathcal{H}}}^T(\boldsymbol{\tau}) \tilde{\boldsymbol{\mathcal{H}}}(\boldsymbol{\tau}) \right]^{-1} \tilde{\boldsymbol{\mathcal{H}}}^T(\boldsymbol{\tau}) \boldsymbol{\epsilon}(n), \tag{5.24}$$

where

$$\tilde{\boldsymbol{\mathcal{H}}}(\boldsymbol{\tau}) = \left[\tilde{\boldsymbol{H}}_{L-1}^T(\boldsymbol{\tau}), \ldots, \tilde{\boldsymbol{H}}_1^T(\boldsymbol{\tau}), \tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau}) \right]^T. \tag{5.25}$$

and

$$\boldsymbol{\epsilon}(n) = \left[\boldsymbol{\epsilon}_0(n), \boldsymbol{\epsilon}_1(n), \ldots, \boldsymbol{\epsilon}_{L-1}(n) \right]^T. \tag{5.26}$$

The quantity $\boldsymbol{\epsilon}_p(n), p = 0, \ldots, L-1$ removes the contribution from samples other than $\boldsymbol{a}(n-L+1)$ in the observation $\boldsymbol{y}(n-p)$. It is defined as

$$\begin{aligned} \boldsymbol{\epsilon}_p(n) &= \boldsymbol{y}(n-p) - \boldsymbol{x}_p(n), \\ &= \tilde{\boldsymbol{H}}_{L-1-p}(\boldsymbol{\tau})\boldsymbol{a}(n-L+1) + \sigma_w \boldsymbol{w}(n), \end{aligned} \tag{5.27}$$

where

$$\boldsymbol{x}_p(n) \triangleq \sum_{l \neq p}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-p-l). \tag{5.28}$$

The $\boldsymbol{\epsilon}_p(n)$ are determined by evaluating the $\boldsymbol{x}_p(n)$ in (5.28), where the unknown quantities $\boldsymbol{a}(n-p-l)$ for $n-p-l > n-L+1$ in (5.28) are tentatively estimated suboptimally using (5.17) and (5.18). The estimation of $\boldsymbol{a}$ according to (5.24) results in significantly improved performance compared to the other more direct methods for the estimation of this parameter.

The MCMC procedure described in the next section proposes a candidate sample $(\boldsymbol{\tau}^\star, k^\star)$ and requires evaluation of the posterior distribution given by (5.20) for that candidate. The following schema summarizes the process used for this evaluation.

Evaluation of the Posterior Density

1. Given a candidate sample $(\boldsymbol{\tau}^\star,\ k^\star)$ from the MCMC procedure, and a suitable interpolation function such as a windowed sinc$(\cdot)$, compute $\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}^\star), l = 0, \ldots, L-1$ of order $k^\star$.

2. For sample index $n = L-1, \ldots, N_t$

   - Follow the steps described in Section 5.3.2 to obtain $\hat{\boldsymbol{a}}(n-L+1)$, in (5.24).
   - Given the source amplitudes, evaluate $\boldsymbol{z}(n-L+1)$ according to (5.3).

3. Given the $\boldsymbol{z}(n)$, $\boldsymbol{R}_{zz}$ in (5.22) can be computed, and the posterior density $\pi(\boldsymbol{\tau}, k|\boldsymbol{Z})$ in (5.20) can be evaluated. This quantity is then used by the MCMC procedure to determine whether the candidate $(\boldsymbol{\tau}^\star,\ k^\star)$ is accepted as a sample.

The computational requirements of the above algorithm are mitigated by the fact that the matrix $(1+\delta^{-2})^{-1}\left[\tilde{\boldsymbol{\mathcal{H}}}^T(\boldsymbol{\tau})\tilde{\boldsymbol{\mathcal{H}}}(\boldsymbol{\tau})\right]^{-1}\tilde{\boldsymbol{\mathcal{H}}}^T(\boldsymbol{\tau})$ in (5.24) is independent of n and therefore need only be computed once per MCMC iteration.

5.4 The reversible jump MCMC algorithm

We now use the *reversible jump* MCMC algorithm (Green, 1995) to perform the Bayesian computation in jointly detecting the desired model order and extracting the other parameters of interest from the posterior distribution in (5.20).

As described in Chapter 2, the reversibile jump MCMC algorithm allows the sampling process to jump between subspaces of different dimensions, which facilitates the detection of model order. In the reversible jump algorithm, candidate samples are chosen from *a set* of proposal distributions, which are randomly accepted according to an acceptance ratio that ensures reversibility, and therefore the invariance of the Markov chain with respect to the desired posterior distribution. Here, we choose our set of proposal distributions to correspond to the following set of moves

1. the *birth* move, valid for $k < M$. Here, a new τ is proposed at random on $[-T_{max}, T_{max}]$.
2. the *death* move, valid for $k > 0$. Here, a τ is randomly chosen to be removed.
3. the *update* move. Here, the order of the model is held fixed and the parameters describing the sources are updated.

The probabilities for choosing each move are denoted by u_k, b_k, and d_k, respectively, such that $u_k + b_k + d_k = 1$ for all k. In accordance with (Andrieu and Doucet, 1999), we choose

$$b_k = c \min\left\{\frac{p(k+1)}{p(k)}, 1\right\}, \quad d_{k+1} = c \min\left\{\frac{p(k)}{p(k+1)}, 1\right\}, \tag{5.29}$$

where $p(\cdot)$ is the prior distribution of the kth model according to (5.12), and c is a tuning parameter that determines the ratio of update moves to jump moves. We choose $c = 0.5$ so that the probability of a jump is between 0.5 and 1 in each iteration (Green, 1995). A detailed description of the reversibile MCMC algorithm and the moves has already been given in Section 2.2.6. For convenience, we present the algorithm and the moves once again as follows

Reversible Jump MCMC

1. Initialization: set $\boldsymbol{\Phi}^{(i=0)} = \left(\boldsymbol{\tau}^{(i=0)}, k^{(i=0)}\right)$, where i is the iteration index
2. Iteration i
   - Sample $u \sim U_{[0,1]}$,
   - if $(u < b_{k^{(i)}})$ then execute a "birth move" (see Section 5.4.2),
   - else if $(u < b_{k^{(i)}} + d_{k^{(i)}})$ then execute a "death move" (see Section 5.4.3),
   - else, execute an update move (see Section 5.4.1).
3. $i \leftarrow i + 1$, goto step 2

■

5.4.1 Update Move

Here, we assume that the current state of the algorithm is in $(\boldsymbol{\tau}_k, k)$. When the update move is selected, the algorithm samples only on the space of $\boldsymbol{\Phi}_k$ for a fixed k. A candidate $\boldsymbol{\tau}^\star$ is sampled from a proposal distribution function $q(\boldsymbol{\tau}|\boldsymbol{\tau}^\star)$, defined as

$$q(\boldsymbol{\tau}|\boldsymbol{\tau}^\star) = p(\boldsymbol{\tau}|k) = p(\boldsymbol{\tau}^\star|k), \tag{5.30}$$

and $p(\tau|k)$ is the prior density for τ given by (5.10). The acceptance function for an update move is defined according to (2.24) as

$$r_{update} = \frac{\pi(\boldsymbol{\tau}^\star|\boldsymbol{Z}^\star)q(\boldsymbol{\tau}|\boldsymbol{\tau}^\star)}{\pi(\boldsymbol{\tau}|\boldsymbol{Z})q(\boldsymbol{\tau}^\star|\boldsymbol{\tau})}. \tag{5.31}$$

Accordingly, substituting (5.20) into (5.31) yields

$$r_{update} = \frac{\pi(\boldsymbol{\tau}^\star|\boldsymbol{Z}^\star)}{\pi(\boldsymbol{\tau}|\boldsymbol{Z})}, \tag{5.32}$$

$$= \left(\frac{\gamma_0 + \mathrm{tr}\left(\tilde{\boldsymbol{P}}^{\perp}_{H_0}(\boldsymbol{\tau}_k)\hat{\boldsymbol{R}}_{zz}\right)}{\gamma_0 + \mathrm{tr}\left(\tilde{\boldsymbol{P}}^{\perp}_{H_0}(\boldsymbol{\tau}^\star_k)\hat{\boldsymbol{R}}^\star_{zz}\right)}\right)^{\frac{MN+\nu_0}{2}}. \tag{5.33}$$

The candidate $\boldsymbol{\tau}^{\star}$ is then accepted as the current state $\boldsymbol{\tau}^{(i+1)} = \boldsymbol{\tau}^{\star}$, with probability

$$\alpha_{update}\left(\boldsymbol{\tau}, \boldsymbol{\tau}^{\star}\right) = \min\left\{1, r_{update}\right\}. \tag{5.34}$$

The schema for the update move is summarized as follows.

Update Move

- Propose a candidate element $\boldsymbol{\tau}^{\star}$ according to (5.10).
- Evaluate α_{update} according to (5.33) and (5.34).
- Sample $u \sim U_{[0,1]}$.
- if $(u \leq \alpha_{update})$ then the state of the Markov Chain at iteration $i+1$ becomes $(\boldsymbol{\tau}^{\star}, k)$, else it remains at $(\boldsymbol{\tau}^{(i)}, k)$.

5.4.2 Birth Move

In the birth move, we assume the state of the algorithm is in $(\boldsymbol{\tau}_k, k)$ at the present ith iteration, and we wish to determine whether the state is in $(\boldsymbol{\tau}_{k+1}, k+1)$ at the next iteration. According to (2.24), the acceptance ratio of the birth move is therefore defined as

$$r_{birth} = \frac{\pi(\boldsymbol{\tau}^{\star}_{k+1}, k+1|\boldsymbol{Z}) q(\boldsymbol{\tau}_k, k|\boldsymbol{\tau}^{\star}_{k+1}, k+1)}{\pi(\boldsymbol{\tau}_k, k|\boldsymbol{Z}) q(\boldsymbol{\tau}^{\star}_{k+1}, k+1|\boldsymbol{\tau}_k, k)}. \tag{5.35}$$

We propose a delay vector $\boldsymbol{\tau}^{\star}$ as

$$\boldsymbol{\tau}^{\star}_{k+1} = \left[\boldsymbol{\tau}^{(i)}_k, \tau_c\right], \tag{5.36}$$

where $\boldsymbol{\tau}^{(i)}_k$ is the delay vector at the ith iteration, and τ_c is a new time delay candidate selected uniformly on $[-T_{max}, T_{max}]$. Note that the prior for model order k in (5.12) is independent of that for $\boldsymbol{\tau}$ in (5.10). In the birth move, only the one new source is a

random variable; the remaining sources are treated as constants. Accordingly, the proposal distribution $q\left(\boldsymbol{\tau}_{k+1}^{\star}, k+1|\boldsymbol{\tau}_k, k\right)$ from (5.35) is then

$$q\left(\boldsymbol{\tau}_{k+1}^{\star}, k+1|\boldsymbol{\tau}_k, k\right) = p(k+1) \times \frac{1}{2T_{max}}. \tag{5.37}$$

In contrast, the distribution $q(k, \boldsymbol{\tau}_k|k+1, \boldsymbol{\tau}_{k+1}^{\star})$ in (5.35) refers to the proposal distribution when one source of $k+1$ is randomly removed. Note that the prior for model order k is independent of that for $\boldsymbol{\tau}$. Thus, we have

$$q(\boldsymbol{\tau}_k, k|\boldsymbol{\tau}_{k+1}^{\star}, k+1) = p(k) \div \binom{k+1}{1}. \tag{5.38}$$

As a result, the ratio of proposal functions in (5.35) becomes

$$\frac{q(\boldsymbol{\tau}_k, k|\boldsymbol{\tau}_{k+1}^{\star}, k+1)}{q(\boldsymbol{\tau}_{k+1}^{\star}, k+1|\boldsymbol{\tau}_k, k)} = \frac{2T_{max}}{\Lambda}, \tag{5.39}$$

and the acceptance ratio r_{birth} becomes

$$\begin{aligned} r_{birth} &= \frac{\pi(\boldsymbol{\tau}_{k+1}^{\star}, k+1|\boldsymbol{Z}^{\star})}{\pi(\boldsymbol{\tau}_k, k|\boldsymbol{Z})} \times \frac{2T_{max}}{\Lambda}, \\ &= \left(\frac{\gamma_0 + \mathrm{tr}\left(\tilde{\boldsymbol{P}}_{H_0}^{\perp}(\boldsymbol{\tau}_k)\hat{\boldsymbol{R}}_{zz}\right)}{\gamma_0 + \mathrm{tr}\left(\tilde{\boldsymbol{P}}_{H_0}^{\perp}(\boldsymbol{\tau}_{k+1}^{\star})\hat{\boldsymbol{R}}_{zz}^{\star}\right)} \right)^{\frac{MN+\nu_0}{2}} \times \frac{1}{(1+\delta^2)^{N/2}} \times \frac{1}{k+1}. \end{aligned} \tag{5.40}$$

The probability of accepting a birth move is therefore defined as

$$\alpha_{birth} = \min\left\{1, r_{birth}\right\}. \tag{5.41}$$

The schema for the birth move is summarized as follows.

Birth Move

- Propose a new element τ_c according to (5.10) and a candidate,

$$\boldsymbol{\phi}_{k+1}^{(i+1)} = [\boldsymbol{\phi}_k^{(i)}, \phi_c].$$

- Evaluate α_{birth} according to (5.40) and (5.41).

- Sample $u \sim U_{[0,1]}$.

- if $(u \leq \alpha_{birth})$ then the state of the Markov Chain at iteration $i+1$ becomes $(\boldsymbol{\tau}_{k+1}^{(i+1)}, k+1)$, else it remains at $(\boldsymbol{\tau}_{k}^{(i)}, k)$.

■

5.4.3 Death Move

In order to maintain the invariant distribution of the reversible jump MCMC algorithm with respect to model order, the Markov chain must be *reversible* with respect to moves across subspaces of different model orders. That is, the probability of moving from model order k to $k+1$ must be equal to that of moving from $k+1$ to k. Therefore we propose a death move in which a source in the current state $(\boldsymbol{\tau}_{k+1}, k+1)$ is randomly selected to be removed such that the next state becomes $(\boldsymbol{\tau}_{k}, k)$ at the next iteration. A sufficient condition for reversibility with respect to model order (Green, 1995) is that the acceptance ratio for the death move be defined as

$$r_{death} = \frac{1}{r_{birth}}, \tag{5.42}$$

and the new candidate of dimension k is accepted with probability

$$\alpha_{death} = \min\left\{1, r_{death}\right\}. \tag{5.43}$$

The schema for the death move is summarized as follows.

Death Move

- Select randomly the jth element to form the candidate,

$$\boldsymbol{\tau}_{k}^{(i+1)} = [\boldsymbol{\tau}_{1:(j-1)}^{(i)}, \boldsymbol{\tau}_{(j+1):(k+1)}^{(i)}].$$

- Evaluate α_{death} according to (5.42) and (5.43).

- Sample $u \sim U_{[0,1]}$.

- if $(u \leq \alpha_{death})$ then the state of the Markov Chain at iteration $i+1$ becomes $(\boldsymbol{\tau}_k^{(i+1)}, k)$, else it remains at $(\boldsymbol{\tau}_{k+1}^{(i)}, k+1)$.

■

5.4.4 Model Order Determination

Even though δ^2 in (5.7) is an estimate of the SNR, in practice it is an unknown quantity. Therefore, in this section we discuss conditions that must be placed on this hyper-parameter to achieve consistent determination of the model order. According to the simplified posterior distribution in (5.20), we can obtain the marginal posterior distribution for model order k as

$$\pi(k|\boldsymbol{Z}) \quad \propto \quad \int_{\boldsymbol{\Phi}_k} \pi(\boldsymbol{\tau}, k|\boldsymbol{Z}) d\boldsymbol{\tau}. \tag{5.44}$$

Denoting the true model order and delay values by k_0 and $\boldsymbol{\tau}_0$ respectively, we perform the following eigendecomposition, at $\boldsymbol{\tau} = \boldsymbol{\tau}_0$

$$\tilde{\boldsymbol{P}}_{H_0}^{\perp}(\boldsymbol{\tau}_0) \hat{\boldsymbol{R}}_{zz} \tilde{\boldsymbol{P}}_{H_0}^{\perp^T}(\boldsymbol{\tau}_0) = \boldsymbol{Q}(\boldsymbol{\tau}_0) \Lambda(\boldsymbol{\tau}_0) \boldsymbol{Q}^T(\boldsymbol{\tau}_0), \tag{5.45}$$

where $\boldsymbol{Q}(\boldsymbol{\tau}_0)$ is an orthonormal matrix which contains the $M-k$ eigenvectors associated with the $M-k$ smallest (noise) eigenvalues of the matrix $\hat{\boldsymbol{R}}_{zz}$, which are placed in the diagonal matrix, $\boldsymbol{\Lambda}(\boldsymbol{\tau}_0)$

$$\boldsymbol{\Lambda}(\boldsymbol{\tau}_0) = \text{diag}\left[\lambda_1, \lambda_2, \ldots, \lambda_{M-k}\right]. \tag{5.46}$$

For convenience, these eigenvalues λ_i, $i = 1, 2, ..., M-k$ are arranged in *ascending* order. Assuming that N_t is large and that the SNR level is moderate, the posterior distribution function $\pi(\boldsymbol{\tau}, k|\boldsymbol{Z})$ concentrates around the true value $\boldsymbol{\tau}_0$. As a result, we can approximate the integral in (5.44) as follows

$$\pi(k|\boldsymbol{Z}) \quad \approx \quad \frac{1}{(1+\delta^2)^{N_t k/2}} \left(\frac{\Xi}{2T_{max}}\right)^k \frac{\exp\left(-\Xi\right)}{k!} \left(\gamma_0 + \sum_{i=1}^{M-k} \lambda_i\right)^{-\left(\frac{MN_t+\nu_0}{2}\right)}. \tag{5.47}$$

Let us define the event E_i as the declaration of a model order in error by i signals. Thus, the event E_i will occur if we declare $\hat{k} = k_0 + i$ or $\hat{k} = k_0 - i$. We assume $P(E_1) > P(E_2) > \cdots > P(E_{M-1})$, which implies that $\pi(k|\boldsymbol{Z})$ is unimodal in k. Accordingly, sufficient conditions which must be satisfied for consistent detection of the model order are

$$\lim_{N_t \to \infty} \frac{\pi(k_0+1|\boldsymbol{Z})}{\pi(k_0|\boldsymbol{Z})} \to 0, \tag{5.48}$$

$$\lim_{N_t \to \infty} \frac{\pi(k_0-1|\boldsymbol{Z})}{\pi(k_0|\boldsymbol{Z})} \to 0. \tag{5.49}$$

From (5.20), we have

$$\frac{\pi(k_0+1|\boldsymbol{Z})}{\pi(k_0|\boldsymbol{Z})} = \frac{\Xi}{2T_{max}} \times \frac{1}{(k_0+1)(1+\delta^2)^{N_t/2}} \left(\frac{\gamma_0 + \sum_{i=1}^{M-k_0-1} \lambda_i}{\gamma_0 + \sum_{i=1}^{M-k_0} \lambda_i} \right)^{-\left(\frac{MN_t+\nu_0}{2}\right)}, \tag{5.50}$$

and

$$\frac{\pi(k_0-1|\boldsymbol{Z})}{\pi(k_0|\boldsymbol{Z})} = \frac{2T_{max}}{\Xi} \times k_0(1+\delta^2)^{N_t/2} \left(\frac{\gamma_0 + \sum_{i=1}^{M-k_0+1} \lambda_i}{\gamma_0 + \sum_{i=1}^{M-k_0} \lambda_i} \right)^{-\left(\frac{MN_t+\nu_0}{2}\right)}. \tag{5.51}$$

From (5.50), we can see that (5.48) is satisfied if

$$1 + \delta^2 > \left(\frac{\gamma_0 + \sum_{i=1}^{M-k_0} \lambda_i}{\gamma_0 + \sum_{i=1}^{M-k_0-1} \lambda_i} \right)^M. \tag{5.52}$$

Similarly, from (5.51) we can see that (5.49) is satisfied if

$$1 + \delta^2 < \left(\frac{\gamma_0 + \sum_{i=1}^{M-k_0+1} \lambda_i}{\gamma_0 + \sum_{i=1}^{M-k_0} \lambda_i} \right)^M. \tag{5.53}$$

Note that the argument on the right in (5.52) contains only noise eigenvalues and hence is very close to one, whereas that in (5.53) contains the smallest signal eigenvalue and the noise eigenvalues is significantly larger than one. Therefore, from (5.52) and (5.53) we have

$$\left(\frac{\gamma_0 + \sum_{i=1}^{M-k_0} \lambda_i}{\gamma_0 + \sum_{i=1}^{M-k_0-1} \lambda_i} \right)^M < 1 + \delta^2 < \left(\frac{\gamma_0 + \sum_{i=1}^{M-k_0+1} \lambda_i}{\gamma_0 + \sum_{i=1}^{M-k_0} \lambda_i} \right)^M. \tag{5.54}$$

Therefore, the specific range of the hyper-parameter δ^2 is dependent on the number of sensors, M, and the current SNR level. Note that if δ^2 is set too small, the expression in

(5.50) will not converge to zero. Thus, it is possible for the algorithm to overestimate the model order as $\pi(k_0+1|\boldsymbol{Z})$ is comparable to $\pi(k_0|\boldsymbol{Z})$. Likewise, if δ^2 is set too large, the expression in (5.51) will not converge to zero. As a result, the algorithm can underestimate the model order when $\pi(k_0-1|\boldsymbol{Z})$ is comparable to $\pi(k_0|\boldsymbol{Z})$.

Referring to (5.54), the determination of the correct value of δ^2 requires the knowledge of the true model order k_0. However, we can still obtain useful information from (5.54) by having an estimate of the SNR level. In the radar application, this is a reasonable assumption as we can obtain an estimate of the noise level by listening for the power level when it is assumed there is no signal present. Similarly, we can get an estimate of signal power by listening when the signal is transmitting. Thus, the right limit in (5.54) can be approximated if there is some knowledge of the SNR level. In practice, the left–hand term in (5.54) is close to unity. With this knowledge, we can obtain a reasonable estimate of the range within which δ^2 must fall.

5.5 Simulation Results

The proposed algorithm is now applied to two scenerios. One is wideband and the other narrowband. In each scenerio, snapshots are generated using (5.1) with the parameters described in Tables 5.1 to jointly detect and estimate the relevant parameters $(\boldsymbol{\tau}, k)$ and the source amplitudes $\boldsymbol{s}_k(n)$. In these experiments, the model order k and the ISD parameters $\boldsymbol{\tau}$ are kept constant throughout the entire observation period. To set up the corresponding interpolation matrix for a set of $\boldsymbol{\tau}$, a *sinc* function is chosen such that the (m,l)th element of the interpolation matrix for the kth source is given by

$$\left[\tilde{\boldsymbol{H}}(\tau_k)\right]_{m,l} = \frac{\sin\left(\pi f_c\left(l-m\tau_k\right)\right)}{\pi f_c\left(l-m\tau_k\right)} \times W\left(l-m\tau_k\right), \tag{5.55}$$

where $l = 0, 1, ..., L-1$ and $m = 0, 1, ..., M-1$, f_c is the cutoff frequency and $W(\cdot)$ is a Hamming window function. In all experiments, the hyperparameters γ_0 and ν_0 are set to zero.

Parameter	Value
SNR (dB)	14
M	8
K	2
L	8
N_t	50
σ_w^2	0.0169
δ^2	25.12
F_s (Hz)	1,000
$\boldsymbol{\theta}$ (deg)	$[-3.44, 3.44]$
$\boldsymbol{\tau}$ (sec)	$[-7.5, 7.5] \times 10^{-5}$

Table 5.1: Common parameters for the Experiments 1 and 2.

5.5.1 Experiment 1: Wideband Scenario

In this experiment, we generate $K = 2$ Gaussian processes for the sources that are zero mean with variance $\delta^2\sigma_w^2$, and bandlimited as follows

$$f \in [100, 400]\,\text{Hz}, \tag{5.56}$$

where the bandwidth of the signals is 300 Hz. According to (3.3), the interspacing of two adjacent sensors, Δ, can be determined as

$$\Delta = \frac{1}{2}\lambda_{min} = \frac{C}{800}. \tag{5.57}$$

The incident angles are -3.44 and 3.44 degrees, respectively, which are separated by an angle less than a standard half-beamwidth. A standard beamwidth is given as (Johnson, 1982)

$$\Delta BW = \sin^{-1}\left(\frac{\lambda}{M\Delta}\right) = \sin^{-1}\left(\frac{1}{4}\right) = 14.18^\circ, \tag{5.58}$$

where $\Delta = \lambda/2$. Using the definition in (3.2), we can obtain the corresponding ISDs as follows

$$\boldsymbol{\tau} = \frac{\Delta}{C}\sin\boldsymbol{\theta} = [-7.5, 7.5] \times 10^{-5}. \tag{5.59}$$

We can then generate $N_t = 50$ snapshots, according to the other parameters in Table 5.1. Figure 5.1 exhibits the magnitude spectrum of the generated wideband signals. The

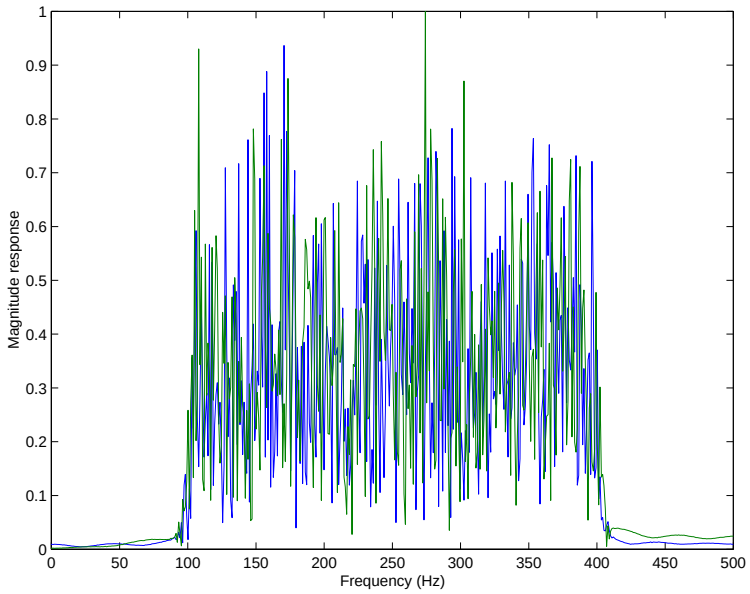

Figure 5.1: The magnitude responses of the wideband signals that are bandlimited within 100 and 400 Hz for Experiment 1.

hyper-parameter δ^2 is assumed known and is chosen as follows

$$\delta^2 = 10^{SNR/10} = 10^{14/10} = 25.12, \tag{5.60}$$

which is within the bounds specified by (5.54). The proposal distribution $q(\tau, k)$ used for these experiments is given by

$$q(\tau, k) = p(k) \cdot p(\tau|k) \tag{5.61}$$

where $p(\tau|k)$ and $p(k)$ are the prior distributions given respectively by (5.10) and (5.12).

The proposed algorithm randomly initializes all unknown parameters, and randomly assigns the initial model order k uniformly in $[1, k_{max}]$, where $k_{max} = M - 1 = 7$, is the maximum allowable model order. The number of MCMC iterations used in the algorithm is 10,000. Figure 5.2 shows the resulting histogram for the number of sources, from which the algorithm predicts the correct number of sources, $k = 2$, whereas Figure 5.3 displays the estimates of the number of sources for each iteration as the algorithm proceeds. The algorithm takes about 25 iterations to converge to the correct order. However, the algorithm takes about 2,000 iterations for a burn-in before the chain centres on the true ISD values.

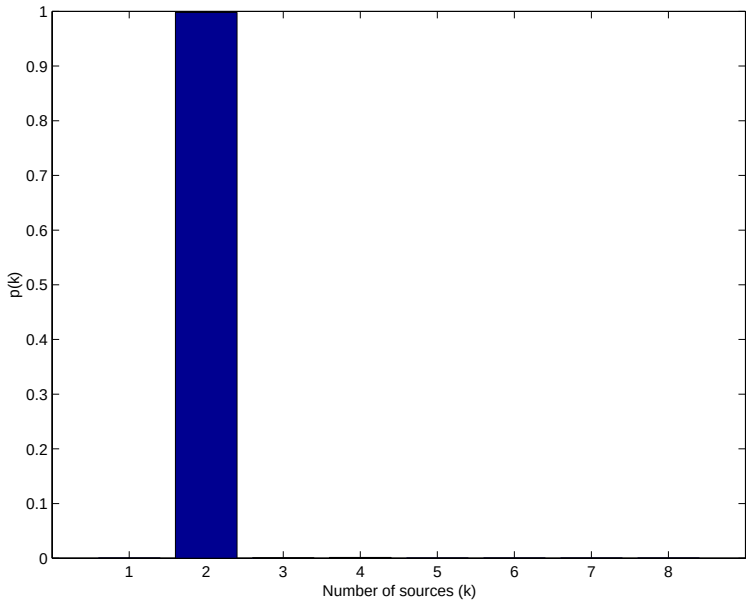

Figure 5.2: Histogram of the number of sources after burn-in for Experiment 1.

Figure 5.4 shows a comparison between the ISD estimates of the sources and the true values after the burn-in stage, versus iteration number. It is clear that the chain centers on the true ISD values. Figure 5.5 shows the marginal histograms of the ISDs from which the MAP estimates are obtained. As a result, the MAP delay estimates $\boldsymbol{\tau}$ from the above simulation are shown in Table 5.2, which also summarizes a comparison between the true and estimated values of the incident angles and the corresponding ISD parameters. Given the MAP estimate of the ISDs $\boldsymbol{\tau}$, the algorithm can now restore the source signals, as shown in Figure 5.6. It is clear that the signal amplitudes are well separated and restored by MCMC. Table 5.3 lists the mean-squared error of the restored amplitudes.

5.5.2 Experiment 2: Narrowband Scenario

In this experiment, we apply the algorithm to a narrowband scenario, where we reduce the bandwidth of the signals from 300 Hz to only 50 Hz. As in the previous experiment, we generate two Gaussian processes that are zero mean with variance $\delta^2\sigma_w^2$ and bandlimited

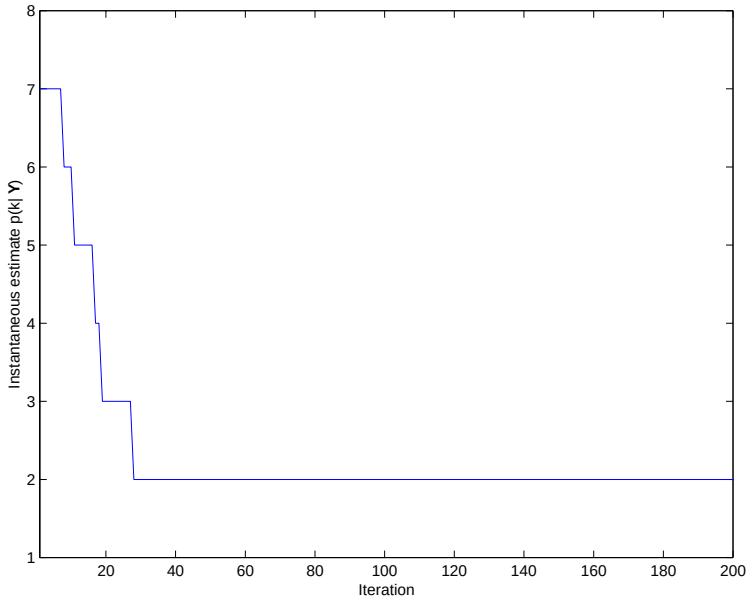

Figure 5.3: Instantaneous estimate of $p(k|\boldsymbol{Y})$, for the first 200 iterations of the chain for Experiment 1.

Parameter	True	Estimated	Relative Difference (%)
τ_0	-7.510^{-5}	-7.9510^{-5}	6.00
τ_1	7.510^{-5}	7.2510^{-5}	3.36
θ_0	-3.44	-3.65	6.00
θ_1	3.44	3.32	3.37

Table 5.2: Comparison between the true and estimated parameters for Experiment 1.

Source	1	2
MSE (dB)	-16.19	-15.97

Table 5.3: The MSE of the restored source amplitudes relative to the true signal amplitudes for Experiment 1.

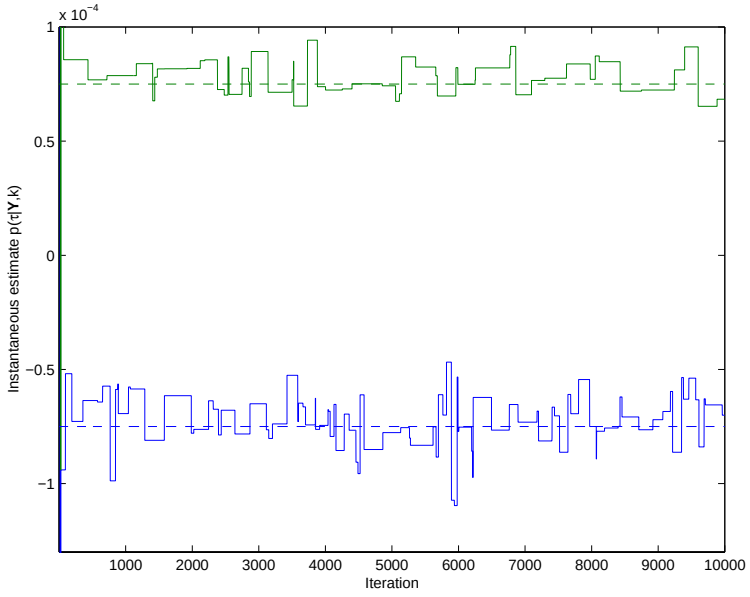

Figure 5.4: Instantaneous estimate of the ISDs $\boldsymbol{\tau}$ for two sources: the solid lines are the estimates and the dashed lines are the true values for Experiment 1.

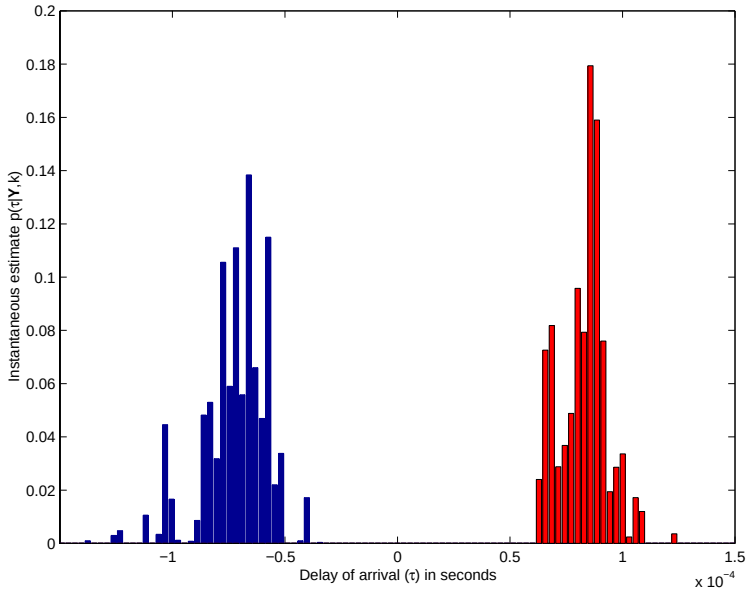

Figure 5.5: Histogram of the ISDs of the sources after burn-in for Experiment 1. The true values are $\pm 7.5 \times 10^{-5}$ seconds.

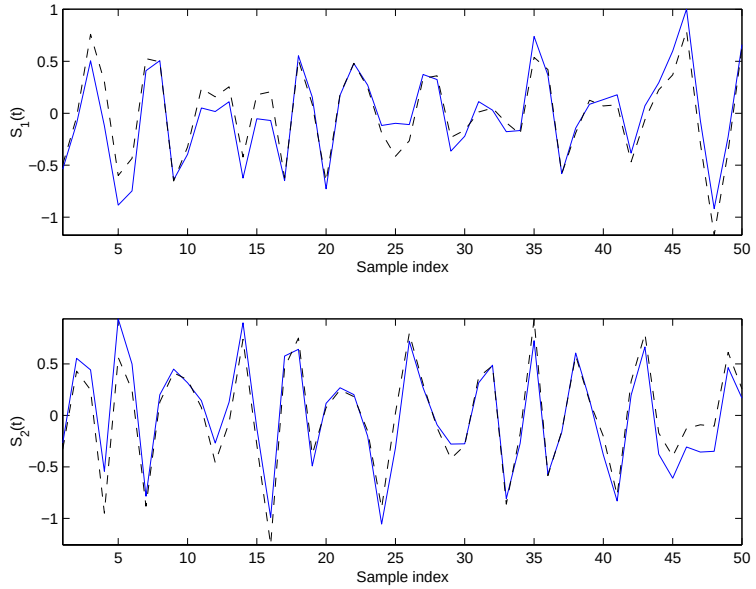

Figure 5.6: A comparison between the true and the restored amplitudes using MCMC in one realization for Experiment 1: solid lines correspond the restored amplitudes using MCMC and dashed lines correspond the true amplitudes.

as follows

$$f \in [350, 400]\,\text{Hz}, \tag{5.62}$$

where the bandwidth of the signals is 50 Hz. Using the parameter values as given in Table 5.1, we can then generate $N_t = 50$ snapshots. In other words, by using the same data model developed in section 5.2 and by tuning the bandwidth parameter Δf_k, one can model both narrowband and wideband scenarios. Figure 5.7 shows the magnitude responses of the generated narrowband signals.

Figures 5.8 and 5.9 show the histogram and the trajectories of the estimation of the ISD parameters, respectively. As in the experiment for the wideband scenario, the algorithm runs for 10,000 iterations, and the chain centers very quickly around the true parameter values for the narrowband scenario as well. Table 5.4 summarizes a comparison between the true and estimated values of the incident angles and the corresponding ISD parameters. Furthermore, Figure 5.10 exhibits a comparison between the restored and the true amplitudes. Table 5.5 lists the mean-squared error of the restored amplitudes for the narrowband

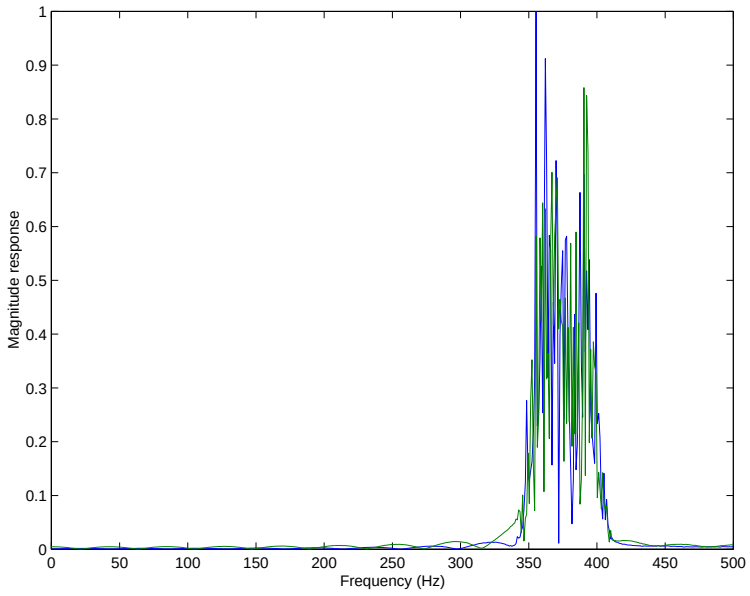

Figure 5.7: The magnitude responses of the narrowband signals that are bandlimited within 350 and 400 Hz for Experiment 2.

case.

As seen in the simulations, the proposed method can perform joint detection, estimation and signal recovery for both narrowband and wideband scenarios. To date, there appears to be no previous method which accomplishes the same set of tasks. Therefore, performance comparisons with previous methods involves comparing the performance of only a subset of the capabilities of the proposed method with the respective criteria of previous methods. We present comparisions with the theoretical Cramér-Rao lower bound (CRLB) derived in

Parameter	True	Estimated	Relative Difference (%)
τ_0	-7.510^{-5}	-6.9410^{-5}	7.47
τ_1	7.510^{-5}	8.1310^{-5}	8.40
θ_0	-3.44	-3.18	7.56
θ_1	3.44	3.73	8.43

Table 5.4: Comparison between the true and estimated parameters for Experiment 2.

Source	1	2
MSE (dB)	-17.31	-17.76

Table 5.5: The MSE of the restored signals relative to the true signal amplitudes for Experiment 2.

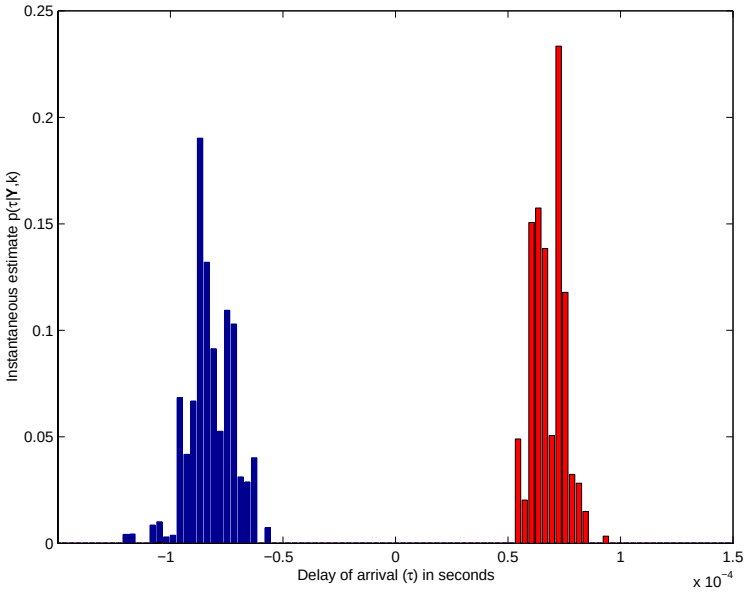

Figure 5.8: Histogram of the ISDs of the sources after burn-in for Experiment 2.

Parameter	Value
M	8
K	2
L	8
N_t	50
No. of MCMC iterations	2,000
σ_w^2	0.0169
F_s (Hz)	1,000
$\boldsymbol{\tau}$ (seconds)	$[-0.510^{-4}0.510^{-4}]$

Table 5.6: Parameters for the performance evaluation for the proposed method.

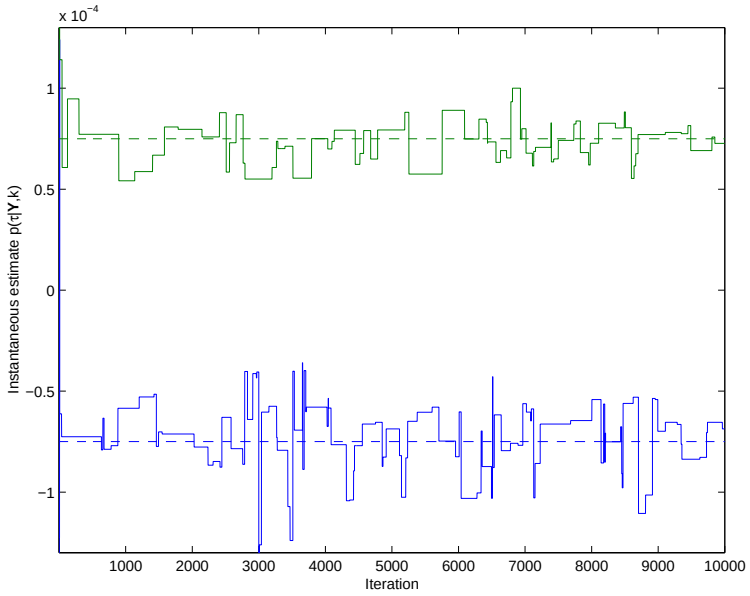

Figure 5.9: Instantaneous estimate of the ISDs $\boldsymbol{\tau}$ for two sources for Experiment 2: the solid lines are the estimates and the dashed lines are the true values.

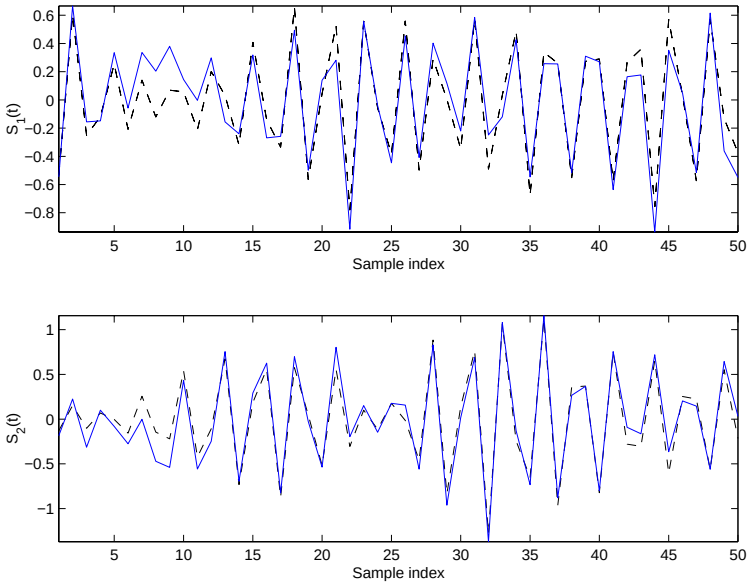

Figure 5.10: A comparison between the true and the restored amplitudes using MCMC for one realization for Experiment 2: solid lines correspond the restored amplitudes using MCMC and dashed lines correspond the true amplitudes.

Appendix C for this problem, and goodness-of-fit and efficiency tests (Li *et al.*, 2001). We also present a comparison of the proposed method and the method of Valaee and Kabal (1995).

We now present the evaluation of the performance of the new method in terms of variances of the estimates of $\boldsymbol{\tau}$ as a function of SNR. This evaluation is obtained by applying the algorithm to 100 independent trials over a range of SNR, from -5dB to 18 dB. The remaining parameter values are given in Table 5.6.

The variances of the estimated $\boldsymbol{\tau}$ are plotted in Figure 5.11 along with the respective theoretical CRLBs. As shown in Figure 5.11, for SNR levels lower than -2dB, the algorithm starts to break down (i.e., it departs rapidly from the CRLB). However, it is seen that the variances approach the CRLB closely, above this level. The reasons why the variances do not come closer to the theoretical CRLB are: 1) interpolation errors due to a non-ideal interpolation function being used and 2) the suboptimal procedure for estimating the source amplitudes. This procedure has an impact on the estimation accuracy of the DOA parameters. Further simulation results, as shown in Figure 5.12, demonstrate that the probability of an error in detection of the model order tends to diminish toward zero with increasing number of snapshots, N_t, with moderate SNR values.

We also use the goodness-of-fit and efficiency tests (Li *et al.*, 2001) to evaluate the proposed method. Denote the normalized estimation error squared (NEES) for $\boldsymbol{\tau}$ by

$$\epsilon_{\boldsymbol{\tau}} \triangleq \tilde{\boldsymbol{\tau}}^T \boldsymbol{R}_{\tilde{\boldsymbol{\tau}}}^{-1} \tilde{\boldsymbol{\tau}}, \tag{5.63}$$

where

$$\tilde{\boldsymbol{\tau}} \triangleq \boldsymbol{\tau} - \hat{\boldsymbol{\tau}}, \tag{5.64}$$

$$\boldsymbol{R}_{\tilde{\boldsymbol{\tau}}} \triangleq E\left[\tilde{\boldsymbol{\tau}}\tilde{\boldsymbol{\tau}}^T\right]. \tag{5.65}$$

The quantity $\epsilon_{\boldsymbol{\tau}}$ is chi-squared distributed with K degrees of freedom, that is

$$\epsilon_{\boldsymbol{\tau}} \sim \chi_K^2. \tag{5.66}$$

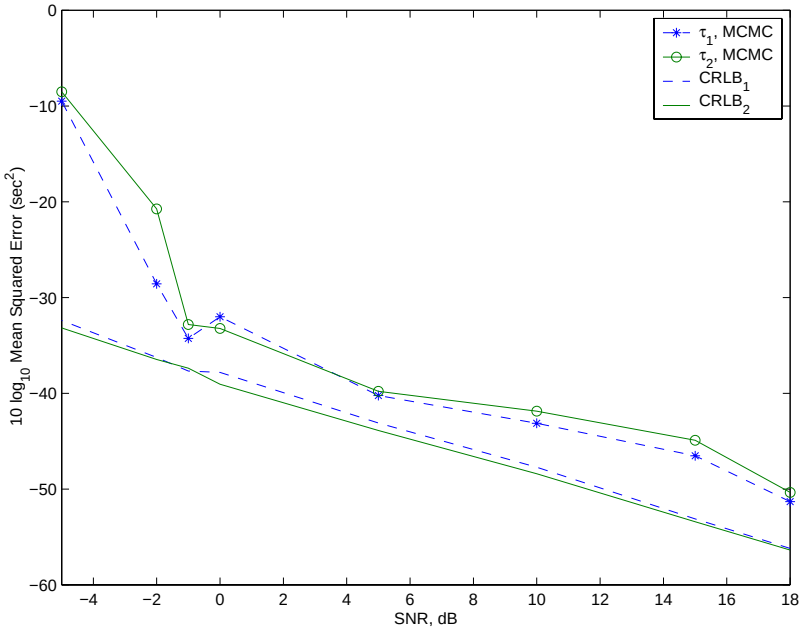

Figure 5.11: Mean squared error of $\boldsymbol{\tau}$ versus the CRLB.

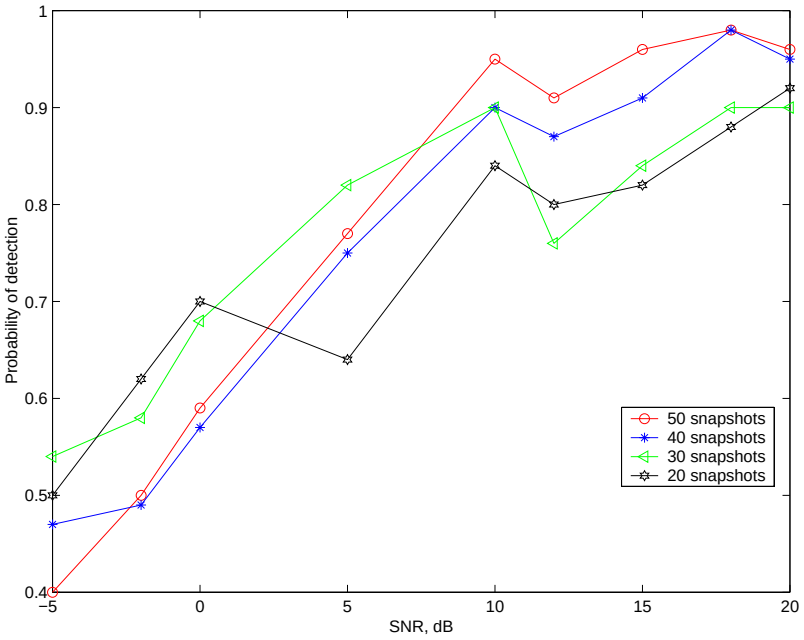

Figure 5.12: Probability of detection as a function of number of snapshots for different SNR values.

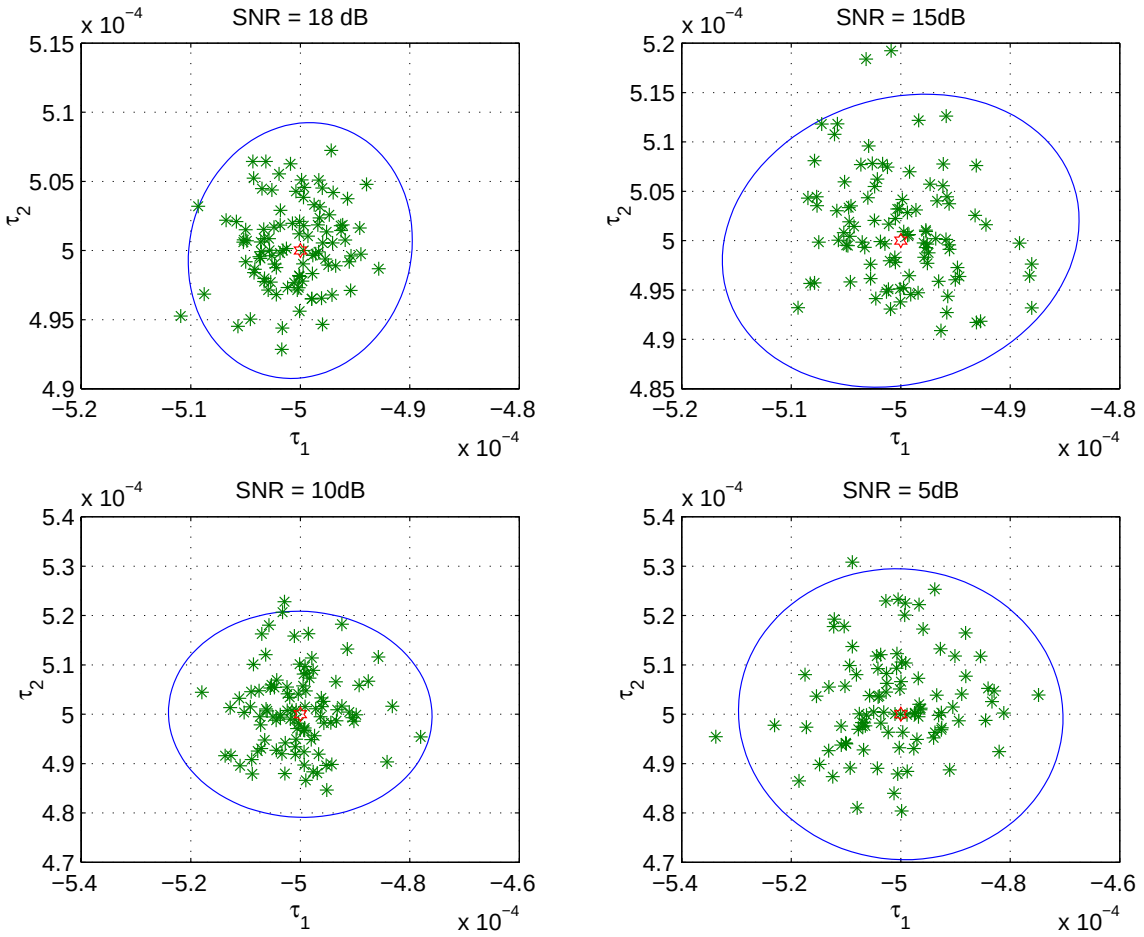

Figure 5.13: The ellipsoids of the NEES with 95% confidence interval on MCMC with different SNR's: the center of each ellipse corresponds the true value of $\boldsymbol{\tau}$, and the asterisks represent the distribution of the estimates of $\boldsymbol{\tau}$ of the 100 independent trials for a particular SNR.

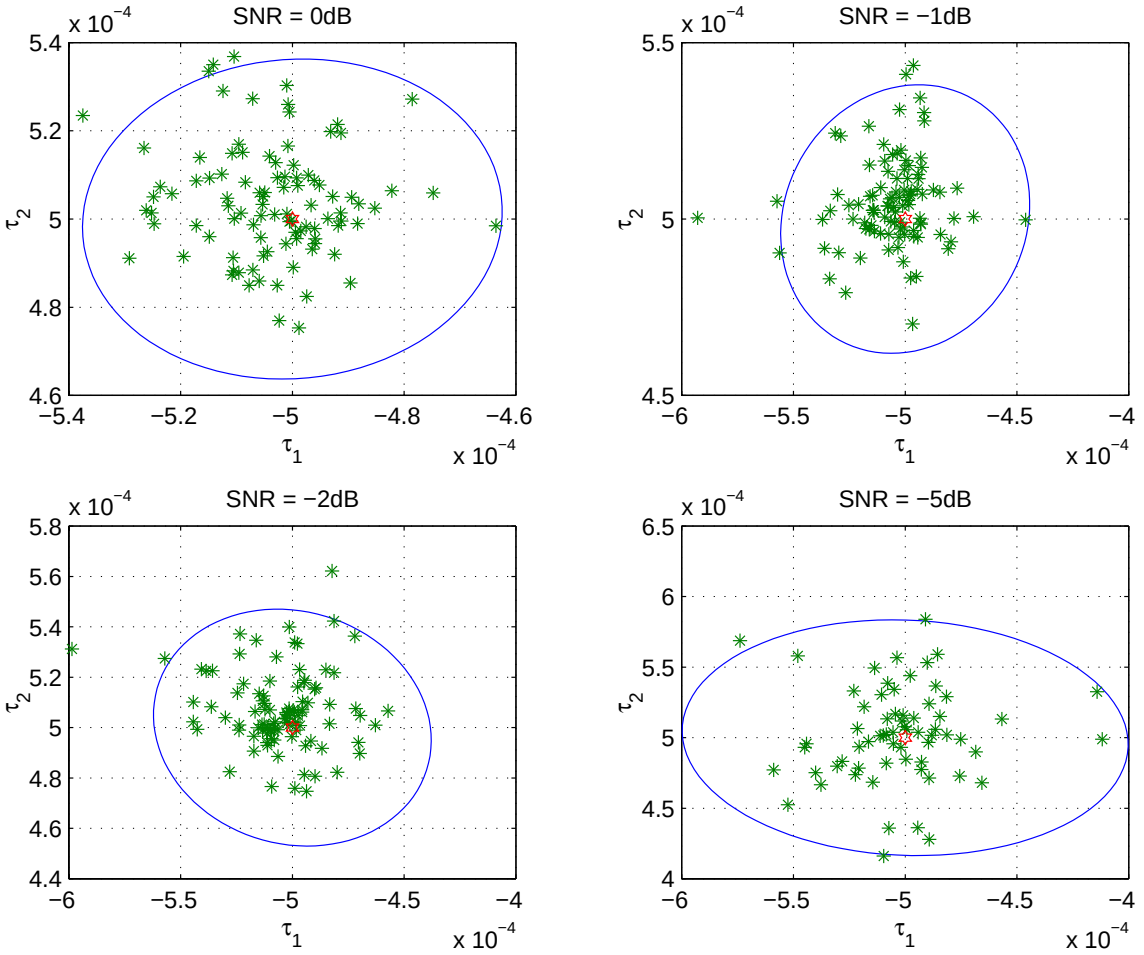

Figure 5.14: The ellipsoids of the NEES with 95% confidence interval on MCMC with different SNR's: the center of each ellipse corresponds the true value of $\boldsymbol{\tau}$, and the asterisks represent the distribution of the estimates of $\boldsymbol{\tau}$ of the 100 independent trials for a particular SNR.

Given a confidence level, say 95%, a set of *g-sigma ellipsoids* for $\epsilon_{\boldsymbol{\tau}}$ (Li *et al.*, 2001) can be obtained with different SNR values. These ellipsoids can be used to evaluate how well the MCMC method is performing in estimating $\boldsymbol{\tau}$. Figures 5.13 and 5.14 shows the impact of different SNR values on the estimation of $\boldsymbol{\tau}$ with a 95% confidence interval. It is clear that as SNR becomes large, the estimates concentrate more closely to the center of each ellipsoid, which corresponds to the true values of $\boldsymbol{\tau}$. However, as SNR falls below 0dB, it is found that the estimates distribute loosely around the center, as evident in Figures 5.13 and 5.14. Accordingly, we can conclude that the proposed method would start to break down in estimating $\boldsymbol{\tau}$ when SNR is below 0dB under the conditions used for this set of experiments.

In addition, the proposed method was tested to determine whether it is an efficient estimator. Using the 95% confidence level, the two-sided probability region for the NEES (Li *et al.*, 2001) for $\boldsymbol{\tau}$ is

$$[\epsilon_1, \epsilon_2] = [1.558, 2.48]. \tag{5.67}$$

Figure 5.15 shows the NEES for $\epsilon_{\boldsymbol{\tau}}$ versus different SNR values. For SNR values above 0dB, the normalized errors fall inside the regions, that is, the method is consistent. However, as the SNR falls below 0dB, the errors are outside the bounds, indicating that the method starts with break down when the SNR is below 0dB. These findings are indeed consistent with those in Figures 5.11, 5.13, and 5.14.

Wideband array processing methods that rely on focusing in Valaee and Kabal (1995) and Wang and Kaveh (1985) require significantly more observations than the proposed method. Focusing methods require at least JM snapshots, where J is the number of frequency bins, so that full–rank covariance matrices can be formulated at each frequency, a requirement for DOA estimation. Since J is typically on the order of 32, a minimum of 256 snapshots would be required before DOA estimates could be produced using the example of this section. Typically 10 times this number would normally be used, so that stable covariance matrix estimates would be obtained, yielding stable DOA estimates. As seen in this section, DOA estimates that come close to the CRLB can be produced with less than 50 snapshots with the proposed method.

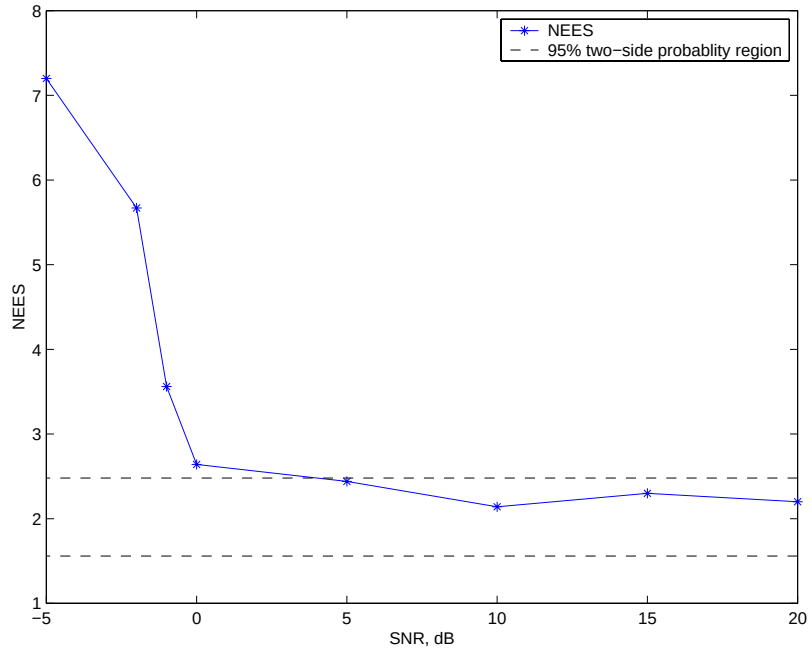

Figure 5.15: Normalized estimation error squared from different SNR values with its 95% probability region.

We now compare the performance between the proposed method and the TCT approach in Valaee and Kabal (1995). Table 5.7 lists the common parameters used in the comparison. A total of 50 independent trials is run on each of these algorithms for different SNR values. The incident angles used are separated by a half-beamwidth. For the proposed method, 5,000 MCMC iterations are run for each simulation, whereas for the TCT algorithm, 32 frequency bins are used. Figure 5.16 depicts the variances of the ISD estimates obtained by these algorithms for different SNR values. According to Figure 5.16, the proposed method outperforms the TCT method throughout the range of SNR values shown. This can be explained by a few reasons. First, the TCT method requires sufficient data in each frequency bin so that a good estimate of the covariance matrix at that frequency can be determined. Therefore, when N_t is not large, the performance of the TCT method degrades. Second, at low SNR, the signal and noise subspace estimates, which are needed by the TCT method, degrade significantly at low SNR, giving rise to an early threshold.

Moreover, according to Figure 5.17, at high SNR, the detection performance of the TCT

Parameter	M	K	N_t	$\boldsymbol{\theta}$ (degrees)
Value	8	2	320	$[3.67, 10.60]$

Table 5.7: Parameters for the performance comparison between the proposed method and the TCT.

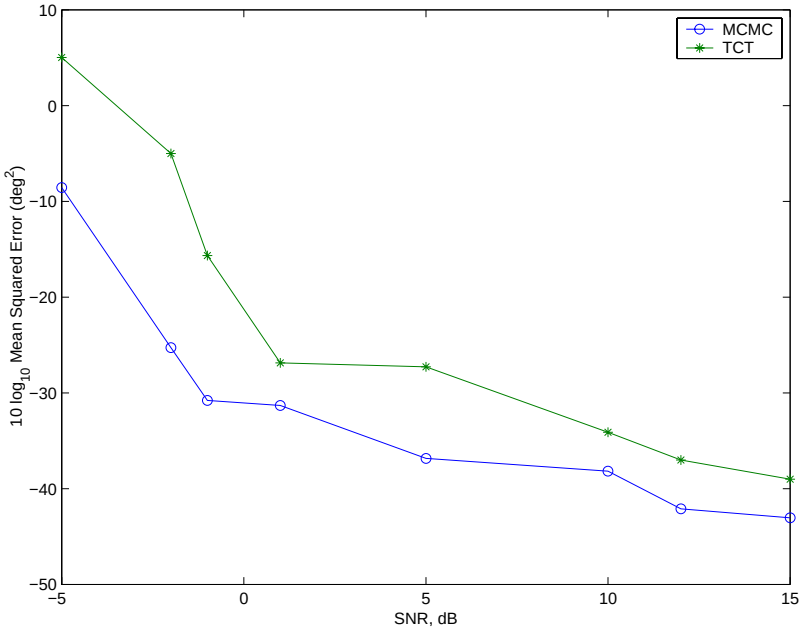

Figure 5.16: Performance comparison between the proposed method and the TCT.

using the MDL is comparable with that of the proposed algorithm, but at low values of SNR, the detection performance of MDL used by the TCT degrades significantly. On the other hand, the proposed method, using a completely different detection approach, outperforms the TCT method under these conditions.

The MCMC method requires roughly 4 to 5 times more computations than the TCT method, for a typical parameter set used here. The number of flop counts per each iteration of the MCMC method is

$$\text{flop counts}_{\text{MCMC}} \approx 2N_tM^2 + 2M^3 - 2N_t + LM \times [2k^2 + kL], \tag{5.68}$$

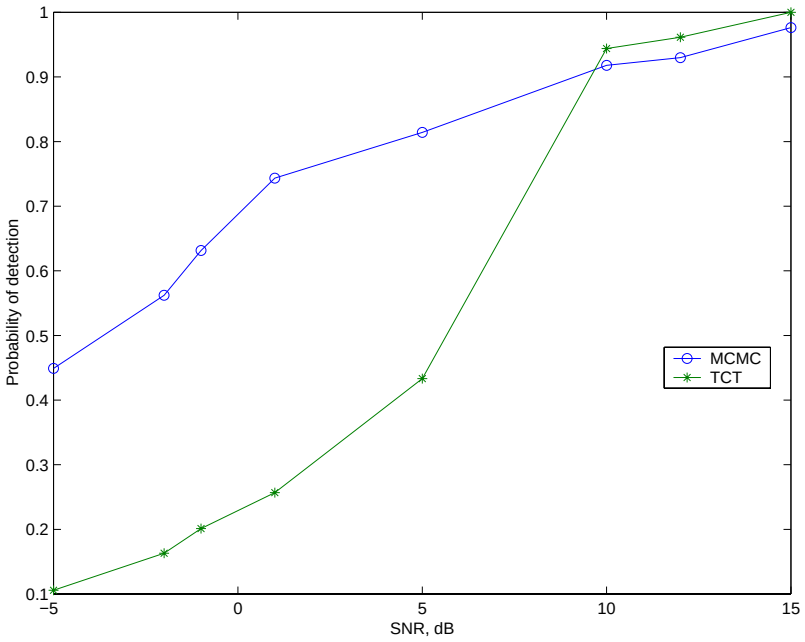

Figure 5.17: Probability of detection obtained from the proposed method and the TCT as a function of SNR values.

and that of the TCT in Valaee *et al.* (1999) is

$$\text{flop counts}_{\text{TCT}} \approx \frac{MN_t}{2}\log_2 M + 2N_tM^2 + [\tfrac{15}{2}J + 9]M^3. \tag{5.69}$$

This apparent disadvantage is mitigated by the fact that the MCMC algorithm provides estimation of the source waveforms, in conjunction with joint detection of model order and estimation of the DOA parameters. Substantially more computations would be required by the TCT method if the source amplitudes were also to be recovered. Further, the MCMC method is easily "parallelizable", thus offering the potential to considerably reduce computation times.

The source estimation procedure described in section 5.4 is recursive. Under adverse conditions, the source estimates generated according to this procedure may occasionally become unstable, with very large error. As an example, when the SNR is 2dB, such a phenomenon appears with an approximate probability of 5%. However, when the SNR is -2dB, the probability grows to 22%. This problem may be alleviated by re–initializing the source amplitudes at periodic intervals throughout the observation interval.

5.6 Conclusion

A novel structure for wideband array signal processing is proposed. It has been demonstrated the method applies equally well to the narrowband case. A Bayesian approach, where a posterior density function that has the nuisance parameters integrated out is formulated, is used. The desired model order and DOA estimation parameters are determined through a reversible jump MCMC procedure. The source amplitudes are given using a MAP estimate. Simulation results support the effectiveness of the method, and demonstrate reliable detection of the number of sources and estimation of their times of arrival in a white noise environment with a single linear array. As a result, the source signals are reliably restored. It has been demonstrated that the method requires only real arithmetic, and that significantly fewer observations are needed relative to what is required for focussing methods.

Chapter 6

Wideband Array Signal Processing II

This chapter proposes a novel online tracking and detection algorithm for wideband array signal processing. Based on the wideband data model in Chapter 2, this method extends the sequential Monte Carlo methods (otherwise known as *particle filtering*) techniques for recursive state estimation. This proposed method finds application in various areas of telecommunications, including radar, sonar, and other wireless communication problems, where locating moving targets and extracting the waveforms for classification are needed.

Conventional approaches to this problem assume the availability of angle-bearing information that is used to track the kinematics, like the locations, velocities, and accelaration, of the targets. In practice, the angle-bearing information must first be estimated; this is much more difficult process in the wideband case than in the narrowband.

The proposed method can recusively estimate the direction-of-arrival (DOA) information corresponding to the sources, which will be used as the input to another process to track the kinematics of the targets. The proposed method first detects the number of unknown wideband sources using a statistical testing procedure, followed by the sequential estimation of the DOAs of these identified sources and the waveform extraction of these sources using a particle filter. Computer simulations demonstrate the ability of the proposed method

in detecting the time-varying nunber of sources in different scenarios as time progresses. The results also show that the DOA trajectories of the identified sources are tracked well and their signal waveforms are also well restored. Finally, the posterior Cramér-Rao bound (PCRB) Tichavsky *et al.* (1998) is also presented to demonstrate that the estimates by the proposed method lie consistently within the bound.

6.1 Introduction

A new approach for wideband array signal processing proposed in Chapter 5 in Ng *et al.* (2002a) adopts a novel data model in the time-domain, and incorporates the Markov chain Monte Carlo methods for parameter estimation. Previous array signal processing approaches assume that the parameters of interest are static within an observation window such that batch or *offline* processing must be used. Unfortunately, this static assumption is often violated in practice, leading to suboptimal results. Since the problem of moving source localization is critical to several important applications in array signal processing, *online* approaches that can recursively estimate the parameters of interest are needed.

In this chapter, we propose a new integrated procedure for joint recursive *online* detection of model order, estimation and tracking of DOAs, and recovery of the source waveforms for the *wideband* scenario. Previous approaches for array signal processing in the wideband case have addressed only one, or possibly only a few of these problems. Specifically, there exist many previous algorithms for DOA tracking, e.g., Gershman (1999); Katkovnik and Gershman (2000); Wigren and Eriksson (1995); Larocque *et al.* (2002), but these only apply to the narrowband scenario. The wideband tracking problem using arrays of sensors has not received much attention to date.

The array processing problem for the wideband scenario is of considerable interest in many communication systems, like radar, sonar, and the wireless 911-problem, etc. In radar applications Haykin (1985), the instantaneous detection and location of targets of interest can be used in improving the accuracy of navigation aids and/or surveillance systems. In addition, the waveforms extracted from the multiple targets can be used to aid in target

classification. Further, wideband target tracking followed by waveform recovery of sources is an essential component of wideband communications for classification with highly maneuvering platforms in hostile mobile environments. In sonar applications, the important goals are the detection and tracking of submarines Van Trees (2002). For applications in the wireless 911-problem, the need to locate wireless callers has recently gained attention. As a matter of fact, it is now a requirement for Commercial Mobile Radio Service providers in the US to provide ubiquitous location coverage for all wireless 911 callers within 125 meters *rms* by October 2001 (Swales *et al.*, 1999; Reed *et al.*, 1988).

The proposed method uses the sequential Monte Carlo (SMC) methods in conjunction with the Markov Chain Monte Carlo (MCMC) methods Gilks *et al.* (1998); Andrieu *et al.* (1999). Simulation results show that the proposed particle filtering approach can track the DOAs of the moving sources and recover the source waveforms. In this chapter, we also present posterior Cramér-Rao bound (PCRB) Tichavsky *et al.* (1998) to demonstrate that the estimates made by the proposed method are consistently close to the bound.

6.2 The State-Space model

In a manner similar to Chapter 4, we assume the states $[\boldsymbol{\tau}(t), \boldsymbol{a}(t)]$ at $t = n$ evolve according to

$$\boldsymbol{\tau}(n) = \boldsymbol{\tau}(n-1) + \sigma_v \boldsymbol{v}(n), \tag{6.1}$$

$$\boldsymbol{a}(n) \sim \mathcal{N}\left(\mathbf{0}, \sigma_a^2 \boldsymbol{I}_{k(n)}\right), \tag{6.2}$$

and the transformed observation model $\boldsymbol{z}(n)$ is given in (3.47) as follows

$$\boldsymbol{z}(n) = \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}(n))\boldsymbol{a}(n) + \sigma_w \boldsymbol{w}(n). \tag{6.3}$$

The noise vector $\boldsymbol{v}(n) \in \mathcal{R}^{k(n)}$ is an *iid* Gaussian variable, defined as

$$\boldsymbol{v}(n) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{I}_{k(n)}\right), \tag{6.4}$$

where σ_v^2 is the state noise variance. The dimension $k(n)$ of the model is described by the following stochastic relationship at time n,

$$k(n) = k(n-1) + \epsilon_k(n), \tag{6.5}$$

where $\epsilon_k(n) \in [-1, 0, 1]$ are discrete *iid* random variables such that

$$\begin{aligned} P(\epsilon_k(n) = -1) &= \begin{cases} 0, & k = 0 \\ h/2, & \text{otherwise,} \end{cases} \\ P(\epsilon_k(n) = 0) &= \begin{cases} 1 - h/2, & k = 0, k_{max} \\ 1 - h, & \text{otherwise,} \end{cases} \\ P(\epsilon_k(n) = 1) &= \begin{cases} 0, & k = k_{max} \\ h/2, & \text{otherwise,} \end{cases} \end{aligned} \tag{6.6}$$

where h is the probability of a change in model order when $k \neq 0, k_{max}$, and k_{max} is the maximum allowable number of sources. In (6.6), it is tacitly assumed that the model order changes by no more than one in each sample period.

In the proposed system of equations, the noise variances σ_v^2 and σ_w^2 are assumed unknown and constant over time. The unknown vectors of amplitudes $\boldsymbol{a}(n)$ are assumed *iid* between snapshots.

We define the vector of all parameters $\boldsymbol{\theta}$ describing the received signal model as

$$\boldsymbol{\theta}_{1:n} \triangleq \left(\{\boldsymbol{\tau}\}_{1:n}, \{\boldsymbol{a}\}_{1:n}, \sigma_v^2, \sigma_w^2 \right), \tag{6.7}$$

where the notation $(.)_{1:n}$ indicates all the elements from time 1 to time n.

The following procedure for developing the desired posterior distribution is similar to that presented in Chapters 4 and 5. The joint posterior distribution of all the parameters is $\pi\left(\boldsymbol{\theta}_{1:n}\right) \triangleq p\left(\boldsymbol{\theta}_{1:n} | \boldsymbol{z}_{1:n}\right)$, which can then be expanded using appropriately selected prior distributions of the parameters, according to Bayes' theorem as

$$\begin{aligned} \pi\left(\boldsymbol{\theta}_{1:n}\right) \propto & p\left(\boldsymbol{z}_{1:n} | \boldsymbol{\tau}_{1:n}, \boldsymbol{a}_{1:n}, k_{1:n}, \sigma_v^2, \sigma_w^2\right) p\left(\boldsymbol{\tau}_{1:n} | k_{1:n}, \sigma_v^2\right) \times \\ & p\left(\boldsymbol{a}_{1:n} | \boldsymbol{\tau}_{1:n}, k_{1:n}, \sigma_w^2\right) p(k_{1:n}) p(\sigma_w^2) p(\sigma_v^2), \end{aligned} \tag{6.8}$$

where $p(z_{1:n}|\cdot)$ is the likelihood term, and the remaining distributions constitute the joint prior distribution for the parameters $\boldsymbol{\theta}$. The individual terms in (6.8) are given as

$$p\left(\boldsymbol{z}_{1:n}|\boldsymbol{\tau}_{1:n}, \boldsymbol{a}_{1:n}, k_{1:n}, \sigma_w^2\right) = \prod_{j=1}^{n} \mathcal{N}\left(\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_j)\boldsymbol{a}_j, \sigma_w^2 \boldsymbol{I}_M\right), \tag{6.9}$$

$$p\left(\boldsymbol{\tau}_{1:n}|k_{1:n}, \sigma_v^2\right) = \prod_{j=1}^{n} \mathcal{N}\left(\boldsymbol{\tau}_{j-1}, \sigma_v^2 \boldsymbol{I}_{k_j}\right), \tag{6.10}$$

$$p\left(\boldsymbol{a}_{1:n}|\boldsymbol{\tau}_{1:n}, k_{1:n}, \sigma_w^2\right) = \prod_{j=1}^{n} \mathcal{N}\left(\boldsymbol{0}, \delta^2\sigma_w^2 \left(\tilde{\boldsymbol{H}}_0^T\left(\boldsymbol{\tau}_j\right)\tilde{\boldsymbol{H}}_0\left(\boldsymbol{\tau}_j\right)\right)^{-1}\right), \tag{6.11}$$

$$p\left(k_{1:n}\right) = \prod_{j=1}^{n} p\left(k_j|k_{j-1}\right) = \prod_{j=1}^{n} \epsilon_k(j). \tag{6.12}$$

The distribution of (6.9) follows from (3.47) and the Gaussian assumption on the noise. The prior distribution of (6.10) follows (6.1) from and the normality assumption on $\boldsymbol{v}(n)$.

The prior distribution defined in (6.11) is the maximum entropy prior with the parameter δ^2 set to an estimate of the SNR. The prior distribution on the variances σ_v^2 and σ_w^2 are both assumed to follow the inverse Gamma distribution, defined as follows (Orton and Fitzgerald, 2002)

$$p\left(\sigma_v^2\right) \sim \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\gamma_0}{2}\right), \tag{6.13}$$

$$p\left(\sigma_w^2\right) \sim \mathcal{IG}\left(\frac{\nu_1}{2}, \frac{\gamma_1}{2}\right). \tag{6.14}$$

The inverse Gamma distribution is noninformative for $\nu, \gamma = 0$.

The parameters of interest are primarily the ISDs $\boldsymbol{\tau}_{1:n}$ and the model order $k_{i:n}$. The amplitudes $\boldsymbol{a}_{1:n}$ along with the noise variances σ_v^2 and σ_w^2 can be estimated separately (as discussed later), and hence may be considered nuisance parameters.

By substituting the prior distributions into (6.8) and integrating out the amplitude

parameters in a manner similar to that in Chapter 4, the resulting posterior distribution is

$$\begin{aligned}\pi\left(\boldsymbol{\tau}_{1:n}, k_{1:n}, \sigma_u^2, \sigma_v^2\right) = &\prod_{j=1}^{n} \frac{1}{\left(2\pi\sigma_u^2\right)^{M/2}\left(1+\delta^2\right)^{k_j/2}} \exp\left\{\frac{-1}{2\sigma_w^2}\boldsymbol{z}_j^T \boldsymbol{P}_0^{\perp}\left(\boldsymbol{\tau_j}\right)\boldsymbol{z}_j\right\} \times \\ &\prod_{j=1}^{n} \frac{1}{\left(2\pi\sigma_v^2\right)^{k_j/2}} \exp\left\{\frac{-1}{2\sigma_v^2}\left(\boldsymbol{\tau}_j - \boldsymbol{\tau}_{j-1}\right)^T\left(\boldsymbol{\tau}_j - \boldsymbol{\tau}_{j-1}\right)\right\} \times \\ &\prod_{j=1}^{n} p\left(k_j | k_{j-1}\right) \times \left(\sigma_v^2\right)^{\frac{-\nu_0}{2}-1} \exp\left\{\frac{-\gamma_0}{2\sigma_v^2}\right\} \times \left(\sigma_w^2\right)^{\frac{-\nu_1}{2}-1} \exp\left\{\frac{-\gamma_1}{2\sigma_w^2}\right\},\end{aligned} \tag{6.15}$$

where

$$\boldsymbol{\Sigma}_0^{-1}\left(\boldsymbol{\tau}_j\right) = \left(1+\delta^{-2}\right)\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau}_j)\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_j), \tag{6.16}$$

$$\begin{aligned}\boldsymbol{m}_j &= \boldsymbol{\Sigma}_0\left(\boldsymbol{\tau}_j\right)\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau}_j)\boldsymbol{z}_j, \\ &= \boldsymbol{\Sigma}_0\left(\boldsymbol{\tau}_j\right)\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau}_j)\left(\boldsymbol{y}(j) - \sum_{l=1}^{L-1}\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}_j)\boldsymbol{a}(t-l)\right),\end{aligned} \tag{6.17}$$

and

$$\boldsymbol{P}_0^{\perp}\left(\boldsymbol{\tau}_j\right) = \boldsymbol{I} - \frac{\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_j)\left(\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau}_j)\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_j)\right)^{-1}\tilde{\boldsymbol{H}}_0^T(\boldsymbol{\tau}_j)}{\left(1+\delta^{-2}\right)}. \tag{6.18}$$

A *maximum a posteriori* estimate of the amplitudes, knowing the other parameters is readily available as

$$\hat{\boldsymbol{a}}_{MAP}(n) = \boldsymbol{m}_n. \tag{6.19}$$

Thus the amplitude parameters need not be included in the particle filter. Instead, they can be estimated at each iteration, after sampling the other parameters.

The MAP estimators of the variances can be readily obtained by comparing (6.15) with a product of Inverted Gamma distributions. Using the fact that the mode of the Inverted Gamma distribution is $\frac{\gamma}{\nu+1}$, it follows that

$$\sigma_{vMAP}{}^2(n) = \frac{\frac{\gamma_0}{2} + \frac{1}{2}\sum_{j=1}^{n}\left(\boldsymbol{\tau}_j - \boldsymbol{\tau}_{j-1}\right)^T\left(\boldsymbol{\tau}_j - \boldsymbol{\tau}_{j-1}\right)}{\frac{\nu_0}{2} + \frac{1}{2}\sum_{j=1}^{n} k_j + 1}, \tag{6.20}$$

$$\sigma_{wMAP}{}^2(n) = \frac{\frac{\gamma_1}{2} + \sum_{j=1}^{n}\boldsymbol{z}_j^T\boldsymbol{P}_0^{\perp}\left(\boldsymbol{\tau}_j\right)\boldsymbol{z}_j}{\frac{\nu_1}{2} + Mn + 1}. \tag{6.21}$$

We choose however to keep these parameters in the expression of the posteriori distribution in (6.15). Since the nuisance parameters can be estimated, we now define a new vector $\boldsymbol{\alpha}$ of parameters to sample with the particle filter, as

$$\boldsymbol{\alpha}_{1:n} \triangleq (\boldsymbol{\tau}_{1:n}, k_{1:n})\,. \tag{6.22}$$

Note that the probabilities in the term $\prod_{j=1}^{n} p(k_j|k_{j-1}) \equiv p(\epsilon_k(j))$ in (6.15) are independent of $\boldsymbol{\tau}$ and only weakly dependent on k. Hence, this term does not significantly affect the estimation of $\boldsymbol{\alpha}$ and is therefore ignored in subsequent analysis. Thus, our estimation problem is now independent of the parameters h and $\epsilon_k(j)$ in (6.6).

6.3 Sequential MC

Details about the sequential Monte Carlo (SMC) procedure has been given in Section 2.3. For convenience, we briefly describe the SMC procedure in this section.

Since it may be difficult or impossible to draw samples directly from the desired distribution $\pi(\boldsymbol{\alpha}_n)$, an importance function $q(\boldsymbol{\alpha}_n)$ that is "easy-to-sample" is used instead to generate $N \gg 1$ samples. The distribution $q(\boldsymbol{\alpha}_n)$ must include the support of $\pi(\boldsymbol{\alpha}_n)$. The so-generated samples, also known as *particles*, are used to compute a set of time-varying importance weights $w(\boldsymbol{\alpha}_n^{(i)})$, $i = 1, ..., N$, that can numerically estimate the target distribution $\pi(\boldsymbol{\alpha}_n)$, i.e.,

$$\widehat{\pi}(d\boldsymbol{\alpha}_n) = \frac{\sum_{i=1}^{N} w(\boldsymbol{\alpha}_n^{(i)}) \delta_{w(\boldsymbol{\alpha}_n^{(i)})}(d\boldsymbol{\alpha}_n)}{\sum_{i=1}^{N} w(\boldsymbol{\alpha}_n^{(i)})}, \tag{6.23}$$

where $(d\boldsymbol{\alpha}_n)$ is a small, finite region surrounding an $\boldsymbol{\alpha}_n$ of interest and $\delta_{w(\boldsymbol{\alpha}_n^{(i)})}$ is an indicator function defined as

$$\delta_{w(\boldsymbol{\alpha}_n^{(i)})}(d\boldsymbol{\alpha}_n) = \begin{cases} 1, & \text{if } \boldsymbol{\alpha}_n^{(i)} \in (d\boldsymbol{\alpha}_n), \\ 0, & \text{otherwise.} \end{cases} \tag{6.24}$$

For notational convenience, we adopt $w^{(i)}(n) = w(\boldsymbol{\alpha}_n^{(i)})$, $i = 1, ..., N$, from this point onwards. Denote a set of N normalized importance weights by $\tilde{w}^{(i)}(n)$, $i = 1, ..., N$, defined

as follows

$$\tilde{w}^{(i)}(n) = \frac{w^{(i)}(n)}{\sum_{j=1}^{N} w^{(j)}(n)}. \tag{6.25}$$

The objective of the SMC procedure is to update the unnormalized weights $w^{(i)}(n)$ of the approximate posterior distribution in (6.25) recursively and sequentially, based on the arrival of new observations. According to (2.43), this set of weights can be recursively updated by the following expression

$$w^{(i)}(n) \quad = \quad \tilde{w}^{(i)}(n-1) \times \frac{p(\boldsymbol{z}_n|\boldsymbol{\alpha}_n^{(i)})p(\boldsymbol{\alpha}_n^{(i)}|\boldsymbol{\alpha}_{n-1}^{(i)})}{q(\boldsymbol{\alpha}_n^{(i)}|\boldsymbol{\alpha}_{1:n-1}^{(i)}, \boldsymbol{z}_{1:n})}, \quad i = 1, ..., N, \tag{6.26}$$

where the terms in the numerator are the likelihood function and the prior distribution function defined in (6.9) and (6.10), respectively, whereas the term in the numerator refers to the importance function.

To approximate an optimal importance function (Doucet *et al.*, 2000; Doucet, 1998; Gordon *et al.*, 1993) that minimizes the overall variance of the weights, that includes the support of the target distribution $\pi(\boldsymbol{\alpha}_n)$, and allows the recursivity defined in (6.26), one can select it to be proportional to the distribution functions (Orton and Fitzgerald, 2002) in the numerator in (6.26), that is

$$q(\boldsymbol{\alpha}_n^{(i)}|\boldsymbol{\alpha}_{n-1}^{(i)}, \boldsymbol{z}_n) \sim p(\boldsymbol{z}_n|\boldsymbol{\alpha}_n^{(i)})p(\boldsymbol{\alpha}_n^{(i)}|\boldsymbol{\alpha}_{n-1}^{(i)}). \tag{6.27}$$

To determine such an optimal importance function, we let $\mathcal{L}(\boldsymbol{\tau}_n)$ be the logarithm of the optimal importance function, i.e.,

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\tau}_n\right) &\triangleq \mathcal{L}_z(\boldsymbol{\tau}_n) + \mathcal{L}_\tau(\boldsymbol{\tau}_n), \\ &= \log p\left(\boldsymbol{z}_n|\boldsymbol{\tau}_{1:n}, \boldsymbol{z}_{1:n-1}\right) + \log p\left(\boldsymbol{\tau}_n|\boldsymbol{\tau}_{1:n-1}, \boldsymbol{z}_{1:n}\right). \end{aligned} \tag{6.28}$$

We use a second-order Taylor expansion on $\mathcal{L}\left(\boldsymbol{\tau}_n\right)$ about the sensibly chosen point $\boldsymbol{\tau}_{n-1}$ to give

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\tau}_n\right) \approx & \mathcal{L}_z\left(\boldsymbol{\tau}_{n-1}\right) + \mathcal{L}_\tau\left(\boldsymbol{\tau}_{n-1}\right) + \left(\nabla\mathcal{L}_z\left(\boldsymbol{\tau}_n\right) + \nabla\mathcal{L}_\tau\left(\boldsymbol{\tau}_n\right)\right)^T\left(\boldsymbol{\tau}_n - \boldsymbol{\tau}_{n-1}\right) + \\ & \frac{1}{2}\left(\boldsymbol{\tau}_n - \boldsymbol{\tau}_{n-1}\right)^T\left(\nabla^2\mathcal{L}_z\left(\boldsymbol{\tau}_n\right) + \nabla^2\mathcal{L}_\tau\left(\boldsymbol{\tau}_n\right)\right)\left(\boldsymbol{\tau}_n - \boldsymbol{\tau}_{n-1}\right), \end{aligned} \tag{6.29}$$

where $\nabla\mathcal{L}_x(\boldsymbol{\tau}_n) \in \mathcal{R}^{k\times 1}$ and $\nabla^2\mathcal{L}_x(\boldsymbol{\tau}_n) \in \mathcal{R}^{k\times k}$ are the gradient vectors and the Hessian matrix of $\mathcal{L}_x(\boldsymbol{\tau}_n)$, respectively, defined as follows

$$\nabla\mathcal{L}(\boldsymbol{\tau}_n) = \frac{\partial}{\partial\boldsymbol{\tau}_n}\left(\mathcal{L}_z(\boldsymbol{\tau}_n) + \mathcal{L}_\tau(\boldsymbol{\tau}_n)\right)\Big|_{\substack{\boldsymbol{\tau}_n=\boldsymbol{\tau}_{n-1}\\ \boldsymbol{a}_n=\boldsymbol{a}_{n-1}}}, \tag{6.30}$$

$$\nabla^2\mathcal{L}(\boldsymbol{\tau}_n) = \frac{\partial^2}{\partial\boldsymbol{\tau}_n^2}\left(\mathcal{L}_z(\boldsymbol{\tau}_n) + \mathcal{L}_\tau(\boldsymbol{\tau}_n)\right)\Big|_{\substack{\boldsymbol{\tau}_n=\boldsymbol{\tau}_{n-1}\\ \boldsymbol{a}_n=\boldsymbol{a}_{n-1}}}. \tag{6.31}$$

Analytical expressions for (6.30) and (6.31) can be found in the Appendix D. Using (6.29), it can be shown (Orton and Fitzgerald, 2002) that the importance sampling function for $\boldsymbol{\tau}_n$ can be expressed as a Gaussian distribution with the following form

$$q(\boldsymbol{\tau}_n^{(i)}|\boldsymbol{\tau}_{n-1}^{(i)}, \boldsymbol{z}_n) \sim \mathcal{N}\left(\boldsymbol{m}_n^{(i)}, \boldsymbol{\Sigma}_n^{(i)}\right), \tag{6.32}$$

where

$$\boldsymbol{\Sigma}_n^{(i)} = -\left(\nabla^2\mathcal{L}_z(\boldsymbol{\tau}_n) + \nabla^2\mathcal{L}_\tau(\boldsymbol{\tau}_n)\right)^{-1}, \tag{6.33}$$

$$\boldsymbol{m}_n^{(i)} = \boldsymbol{\tau}_{n-1}^{(i)} + \boldsymbol{\Sigma}_n^{(i)}\left(\nabla\mathcal{L}_z(\boldsymbol{\tau}_n) + \nabla\mathcal{L}_\tau(\boldsymbol{\tau}_n)\right). \tag{6.34}$$

As described in Section 2.3, the recursion in (6.26) quickly degenerates, so that after a few time steps, only a handful of particles have weights significantly different from zero. The Sampling Importance Resampling (SIR), which resamples the particles according to their respective importance weights, is first used to cope with the degeneracy, followed by an MCMC step to re-introduce the statisical diversity amongst the resampled particles (Andrieu *et al.*, 1998, 1999; Larocque *et al.*, 2002).

In summary, the proposed SMC approach is basically a combination of a sequential Bayesian importance sampling, the SIR procedure, and an MCMC step. Note that the only parameter in $\boldsymbol{\alpha}$ which is subjected to the sampling procedure is the ISD vector $\boldsymbol{\tau}$. The model order $k(n)$ is determined using a statistical testing procedure to be described in Section 6.4. The desired amplitudes (which give the source waveforms), are estimated from (6.17), and the variances are estimated according to (6.20) and (6.21). We summarize these steps in the following table.

Sequential Importance Sampling Algorithm

Initialization

For time $n = 1$,

- sample N particles $\boldsymbol{\tau}^{(i)}, i = 1, \ldots, N$ from $q(\cdot|\cdot)$.
- initialize the weights $w^{(i)}$, $i = 1, ..., N$ to $\frac{\pi(\boldsymbol{\tau}^{(i)})}{q(\boldsymbol{\tau}^{(i)})}$.
- normalize the weights to $\tilde{w}^{(i)}(n) = \frac{w^{(i)}(n)}{\sum_{j=1}^{N} w^{(j)}(n)}$.

Then for $n = 2, 3, ...$

1. *Sequential Importance Sampling Step*

   (a) Sample N particles of $\boldsymbol{\tau}_n^{(i)}$ for $i = 1, 2, \ldots, N$ from the approximately optimal importance function given by (6.32) as follows

   $$\boldsymbol{\tau}_n^{(i)} \sim q(\boldsymbol{\tau}_n^{(i)}|\boldsymbol{\tau}_{1:n-1}^{(i)}, \boldsymbol{z}_{1:n})$$

   (b) Evaluate the importance weights for N particles as follows:

   $$w^{(i)}(n) = \tilde{w}^{(i)}(n-1) \times \frac{p(\boldsymbol{z}_n|\boldsymbol{\tau}_n^{(i)})p(\boldsymbol{\tau}_n^{(i)}|\boldsymbol{\tau}_{n-1}^{(i)})}{q(\boldsymbol{\tau}_n^{(i)}|\boldsymbol{\tau}_{1:n-1}^{(i)}, \boldsymbol{z}_{1:n})},$$

   and hence the normalized importance weights as follows:

   $$\tilde{w}^{(i)}(n) = \frac{w^{(i)}(n)}{\sum_{j=1}^{N} w^{(j)}(n)}$$

2. *Sampling Importance Resampling Step*

   Multiply/Supress the particles $\boldsymbol{\tau}^{(i)}(n)$ respectively with high/low importance weights $\tilde{w}^{(i)}(n)$ to obtain N random samples approximately distributed according to $\pi(\boldsymbol{\tau}_{1:n}^{(i)})$.

   - Sample a vector of $\boldsymbol{l}$ distributed as:

   $$P(l(j) = i) = w^{(i)}(n)$$

- Resample the particles with the index vector:

$$\boldsymbol{\tau}^{(i)}(n) = \boldsymbol{\tau}^{(l(i))}(n)$$

- Re-assign all the weights to $w^{(i)}(n) = \frac{1}{N}$

3. *MCMC Step* Here we discuss resampling the particles using an MCMC procedure with fixed model order $k = k(n)$.

   As mentioned in Section 2.3, we choose the Metropolis-Hastings (M-H) algorithm (Gilks *et al.*, 1998; Ruanaidh and Fitzgerald, 1996; Hastings, 1970) for the sampling procedure. Given that the current state is in $(\boldsymbol{\tau}_k(n), k(n))$, the algorithm samples a candidate $\boldsymbol{\tau}_k^\star$ according to the proposal distribution $q\left(\boldsymbol{\tau}_k^\star|\boldsymbol{\tau}_k(n)\right)$ which is given from (6.1) as

$$q\left(\boldsymbol{\tau}_k^\star|\boldsymbol{\tau}_k(n)\right) \sim \mathcal{N}\left(\boldsymbol{\tau}_k(n-1), \sigma_v^2 \boldsymbol{I}_k\right). \tag{6.35}$$

   The candidate is accepted with probability

$$\alpha_{update}\left(\boldsymbol{\tau}_k(n), \boldsymbol{\tau}_k^\star\right) = \min\left\{1, r_{update}\right\}, \tag{6.36}$$

   where r_{update} is given by (Hastings, 1970)

$$r_{update} = \frac{\pi\left(\boldsymbol{\tau}_k^\star|\boldsymbol{z}^\star\right) q\left(\boldsymbol{\tau}_k(n)|\boldsymbol{\tau}_k^\star\right)}{\pi\left(\boldsymbol{\tau}_k(n)|\boldsymbol{z}_t\right) q\left(\boldsymbol{\tau}_k^\star|\boldsymbol{\tau}_k(n)\right)}. \tag{6.37}$$

   Accordingly, substituting (6.15) into (6.37) yields

$$r_{update} = \frac{\exp\left\{-\frac{1}{2\sigma_w^2}\boldsymbol{z}_t^{\star T}\boldsymbol{P}_0^\perp(\boldsymbol{\tau}^\star)\boldsymbol{z}_t^\star\right\}}{\exp\left\{-\frac{1}{2\sigma_w^2}\boldsymbol{z}_t^T\boldsymbol{P}_0^\perp(\boldsymbol{\tau_t})\boldsymbol{z}_t\right\}}. \tag{6.38}$$

   If the candidate $\boldsymbol{\tau}_k^\star$ is accepted, then the current state becomes $\boldsymbol{\tau}_k^{(i)}(n) = \boldsymbol{\tau}_k^\star$. Otherwise, it remains at the current state, that is, $\boldsymbol{\tau}_k^{(i)}(n) = \boldsymbol{\tau}_k^{(i)}(n)$.

■

Note that the MCMC process normally requires a "burn–in" period for the chain to reach equilibrium. However in this case, this is not required, since the particles are already distributed according to the desired posterior distribution, which is the invariant distribution of the chain, before application of the MCMC procedure.

6.4 Model Order Detection

In this section, we present a rational technique that consists of two stages for determining which model is most *consistent* with a given set of data. It is necessary that the number of sources be accurately determined before the particle filters can accurately estimate the DOA tracks. The proposed method assumes the observation noise variance σ_w^2 is known. If the value of σ_w^2 is not known beforehand, it could be estimated for a hypothesized value of k by (6.21).

To facilitate the model order detection, we work on the data model that is expressed as a linear combination of individual sources, as follows

$$\boldsymbol{y}(n) = \sum_{k=0}^{k(n)-1} \tilde{\boldsymbol{H}}(\tau_k(n))\boldsymbol{s}_k(n) + \boldsymbol{w}(n), \tag{6.39}$$

$$= \hat{\boldsymbol{y}}(n) + \boldsymbol{w}(n), \tag{6.40}$$

where

$$\hat{\boldsymbol{y}}(n) = \sum_{k=0}^{k(n)-1} \tilde{\boldsymbol{H}}(\tau_k(n))\boldsymbol{s}_k(n). \tag{6.41}$$

Denote $\hat{\boldsymbol{y}}_{-k_0}(n)$ as follows

$$\hat{\boldsymbol{y}}_{-k_0}(n) = \sum_{k\neq k_0}^{k(n)-1} \tilde{\boldsymbol{H}}(\tau_k(n))\boldsymbol{s}_k(n), \tag{6.42}$$

which amounts to an estimate of the observation $\boldsymbol{y}(n)$ by excluding the contribution of the k_0th source.

In the model order detection, two hypotheses are considered at each time n. They are $\mathcal{H}_b$ and $\mathcal{H}_d$, which represent the birth of a new source and the death of an existing

source, respectively. In the hypothesis $\mathcal{H}_b$, a new source in addition to the existing sources is proposed such that $k(n) \rightarrow k(n)+1$, whereas in the hypothesis $\mathcal{H}_d$, one of sources from the existing sources is proposed for removal such that $k(n) \rightarrow k(n)-1$. Deciding which hypothesis is taken amounts to testing whether the current model order fits the observation. Next, we present these procedures and the associated statistical testing proceduces in detail.

6.4.1 Death of an existing source

In this step, the existing $k(n)$ sources are sequentially removed, and the normalized squared residuals, $\epsilon_{-k}(n)$, for $k = 0, ..., k(n)-1$, are each evaluated as follows

$$\epsilon_{-k}(n) = \frac{\boldsymbol{e}_y^T(n)\boldsymbol{e}_y(n)}{\sigma_w^2}, \tag{6.43}$$

where $\boldsymbol{e}_y(n)$ is the residual, defined as

$$\boldsymbol{e}_y(n) = \boldsymbol{y}(n) - \hat{\boldsymbol{y}}_{-k}(n). \tag{6.44}$$

Under the hypothesis that the kth source is zero, the quantity $\epsilon_{-k}(n)$ is χ^2-squared distributed with $M - k(n) + 1$ degrees of freedom, that is

$$\epsilon_{-k}(n) \sim \chi^2_{M-k(n)+1}. \tag{6.45}$$

Thus, it is possible to test whether a particular source contributes significantly to the observation $\boldsymbol{y}(n)$ at time n. The normalized error $\epsilon_{-k}(n)$ is compared to a threshold $c_d > 0$ to test whether the current model order $k(n)$ fits the data, that is

$$P(\epsilon_{-k}(n) > c_d) = \chi^2_{M-k(n)+1}(1-\alpha_d). \tag{6.46}$$

Here, the probability of the normalized error $\epsilon_{-k}(n)$ exceeding a threshold $c_d > 0$ is α_d. If $\epsilon_{-k}(n)$ is larger than c_d, it is likely that the model order $k(n)-1$ with the removal of the kth source is too low. In other words, the kth source should not be removed from the set of existing sources. On the other hand, if $\epsilon_{-k}(n)$ is smaller than c_d, it implies that the contribution of the kth source to the observation is probably insignificant and the removal of this source should be considered. However, in order to further justify the decision to

remove the kth source, an additional statistical test is imposed. Since the signal amplitude is estimated according to (6.17), which is essentially a least-square solution, if a source exists and contributes to the observation $\boldsymbol{y}(n)$ at time n, its corresponding amplitude $s_k(n)$, as well as the normalized signal power $\sigma^2_{s_k}(n)$, defined as

$$\sigma^2_{s_k}(n) = \frac{\boldsymbol{s}_k^T(n)\tilde{\boldsymbol{H}}^T(\tau_k(n))\tilde{\boldsymbol{H}}(\tau_k(n))\boldsymbol{s}_k(n)}{\sigma_w^2}, \tag{6.47}$$

should be persistently significant. On the other hand, if the source has indeed suddenly or slowly vanished, both the restored amplitude and the normalized signal power $\sigma^2_{s_k}(n)$ will become insignificant. To test the significance of $\sigma^2_{s_k}(n)$, which amounts to testing the presence of the source, an *F-test* (Snedecor and Cochran, 1980) can be used.

An F-test, used to test if the estimated variances from two populations are equal, is given as follows

$$F = \frac{\sigma_1^2}{\sigma_2^2}, \tag{6.48}$$

where σ_1^2 and σ_2^2 are the estimates from the two populations. The more the ratio F deviates from 1, the stronger the evidence for unequal population variances. Given a significance level α_f, the hypothesis that the two variances are equal is rejected if

$$F > \mathcal{F}\left(\alpha_f/2, \beta_1 - 1, \beta_2 - 1\right), \tag{6.49}$$

where $\mathcal{F}\left(\alpha_f/2, \beta_1 - 1, \beta_2 - 1\right)$ is known as the critical value of the F distribution (Snedecor and Cochran, 1980), where β_1 and β_2 are the degrees of freedom in the numerator and denominator in (6.47). The quantity α_f represents the significance level of the test. Therefore, to test whether $\sigma^2_{s_k}(n)$ is significant when compared with the noise variance, an F-test on $\sigma^2_{s_k}(n)$ is applied.

6.4.2 Birth of a new source

In this step, we test whether the current model order is too low to fit the observation $\boldsymbol{y}(n)$ at time n. Given the current model order $k(n)$, we first compute the normalized residual

as follows

$$\epsilon(n) = \frac{\boldsymbol{e}_y^T(n)\boldsymbol{e}_y(n)}{\sigma_w^2}, \tag{6.50}$$

where the residual $\boldsymbol{e}_y(n)$ is given by

$$\boldsymbol{e}_y(n) = \boldsymbol{y}(n) - \hat{\boldsymbol{y}}(n), \tag{6.51}$$

where $\hat{\boldsymbol{y}}(n)$ is given by (6.41). Under the hypothesis that the number of signals is correct, the quantity $\epsilon(n)$ is also χ^2-squared distributed with $M - k(n)$ degrees of freedom, that is

$$\epsilon(n) \sim \chi^2_{M-k(n)+1}. \tag{6.52}$$

To test whether the current model fits the current observation, the normalized error $\epsilon(n)$ is compared with a threshold $c_b > 0$, that is

$$P(\epsilon(n) > c_b) = \chi^2_{M-k(n)}(1 - \alpha_b). \tag{6.53}$$

The probability of the normalized error $\epsilon(n)$ exceeding a threshold $c_b > 0$ is α_b. If $\epsilon(n) < c_b$, it is likely that the current model is appropriate for the current observation, and there is no need to introduce a new source to the existing tracks. However, if $\epsilon(n) > c_b$, the current model order is likely too low to fit the observation, leading to larger estimation error. In this case, the current model order will be incremented by one to $k(n) = k(n)+1$, and a new source will be generated according to importance sampling procedure described earlier.

Model Order Detection Using Hypothesis Testing

In summary, for $n = 1, 2, ...,$ the MAP values of $\boldsymbol{\tau}(n)$ and $\boldsymbol{a}(n)$ obtained from the proposed Particle Filter method are used to detect the instantaneous model order, according to the following hypothesis testing procedure

1. $\mathcal{H}_d$ – *Death of an existing source*

(a) evaluate $\epsilon_{-k}(n)$ in (6.43) for $k = 0, ..., k(n)-1$, and compare it with the threshold c_d

- if $\epsilon_{-k}(n) < c_d$, the model order appears to fit the observation
- else, evaluate $\sigma^2_{s_k}(n)$ as in (6.47) and test it using a F-test
  - if the F-test fails, the model order remains intact. Proceed with testing $\mathcal{H}_b$.
  - else, remove the source k.

2. $\mathcal{H}_b$ – *Birth of a new source*

(a) evaluate $\epsilon(n)$ as in (6.50), and compare it with the threshold c_b

- if $\epsilon(n) < c_b$, the model order appears to fit the observation
- else, the current model order is probably too low,
  - increment the model order by 1,
  - re-run the sequential importance sampling steps for the new model order to estimate $\boldsymbol{\tau}(n)$ and $\boldsymbol{a}(n)$,
  - once completed, by-pass the test for $\mathcal{H}_d$ and $\mathcal{H}_b$ for the current time step.

■

6.5 Simulation Results

In the following simulations, a uniform linear array composed of $M = 8$ elements with a half-wavelength spacing of the elements at the highest frequency component is used in the simulations. The wideband signals in these experiments are bandlimited to the normalized frequency values as follows

$$f \in [0.1, 0.4]\,,$$

thus the normalized bandwidth of the signals is 0.3. The simulation environment is defined using the parameters listed in Table 6.1.

Parameter	SNR (dB)	M	K	L	N	σ_w^2	σ_v^2	δ^{-2}	F_s (Hz)
Value	14	8	2	8	$1,000$	0.0169	$(3,000F_s)^{-2}$	25.12	1

Table 6.1: Parameters for the experiment.

In these simulations, we demonstrate the capability of the proposed algorithm in jointly tracking the inter-sensor delays (ISDs) $\boldsymbol{\tau}(n)$ of the sources[1], detecting the model order $k(n)$, and recovering the source waveforms $\boldsymbol{a}(n)$, for the wideband scenario. The ISDs of the $K = 2$ sources are generated as autogressive (AR) processes whose coefficients represent a 10th–order low–pass Butterworth filter, with normalized cutoff frequency 0.1. The variance of the process is σ_v^2. The associated AR coefficients are specified in Table 4.3. Likewise, the source waveforms are also generated as AR processes, corresponding to a 10th–order low–pass butterworth filter with normalized cutoff frequency 0.3 and variance $\delta^2\sigma_w^2$. The associated waveform AR coefficients are also listed in Table 4.3.

To demonstrate the algorithm's capability to detect the time-varying model order, the ISD trajectories $\boldsymbol{\tau}_n$ are constructed so that one of the sources suddenly vanishes or appears. The evolution of the model order $k(n)$ is first tracked by the proposed statistical testing procedure. Then the ISDs with the latest model order are tracked using the particle filter procedure. The source waveforms are then extracted using (6.17) and (6.19). Two different scenarios are presented in demonstrating the ability of the algorithm in handling the model order detection in different situations.

6.5.1 Scenario 1: A source suddenly appears

In this scenario, one source suddenly appears during the course of the ISD tracking. Figure 6.1 shows the trajectories of the ISDs of the two sources. Before the algorithm starts, the parameters are appropriately initialized as described earlier in the SMC procedure with $k(0) = 1$. As the algorithm starts for $n \geq 1$, only one source is found, which is $\tau_0(0) = -0.9T_s$. According to Figures 6.1 and 6.2, both the ISD (the lower trajectory in

[1]Once the ISDs of the sources are available, the directions-of-arrival (DOAs) of the sources $\boldsymbol{\phi}(n)$ are determined through (3.42).

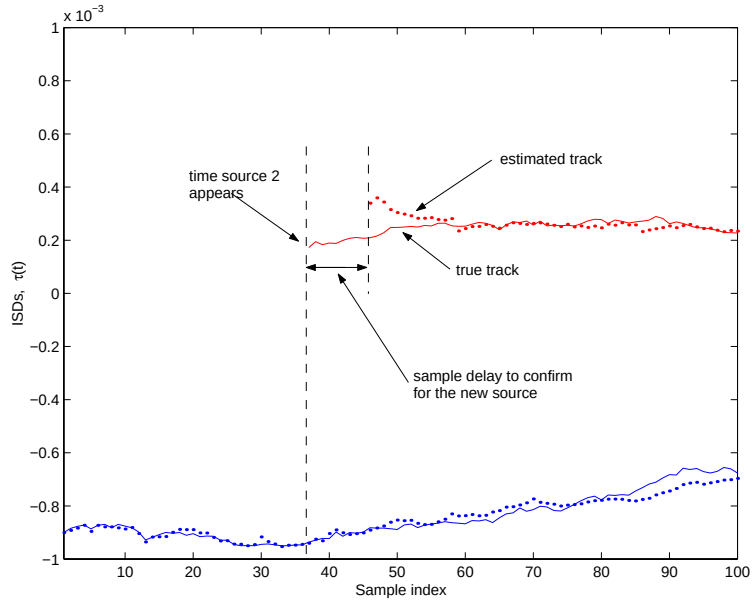

Figure 6.1: The tracking of ISDs $\boldsymbol{\tau}(t)_{|t=n}$ for scenario 1.

Figure 6.1) and the signal amplitude of this source $s_0(n)$ are well estimated by the algorithm in the time region $n \in [1, 37]$. When $n > 37$, a new source $s_1(n)$ appears and the corresponding ISD is approximately $0.2T_s$. Based on the statistical testing procedure, the algorithm does not detect the presence of the new source until $n > 45$, as shown in Figure 6.3. After the new source has been found, it takes approximately 15 additional time steps for the algorithm to stabilize as shown in Figure 6.1 before the new source is well tracked. Note that in the course of the process, the noise variances σ_v^2 and σ_w^2 can also be sequentially estimated using the MAP procedure in (6.20) and (6.21), respectively.

The mean-squared errors for the ISD estimation and the waveform recovery process of the sources are shown in Table 6.2, corresponding to 50 independent trials of the same simulation scenario. As one would expect, the errors from tracking and recovering the waveform of only one source are much smaller than in the case of multiple sources, especially when the SNR is about 14dB. Moreover, given the MAP procedure stated in (6.17), the performance of waveform recovery is heavily dependent on that of the ISD tracking, as evident from Table 6.2.

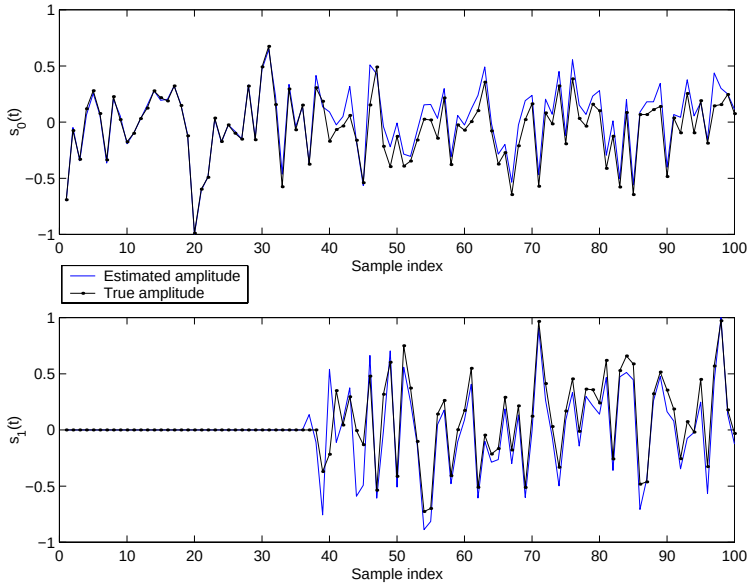

Figure 6.2: The signal recovery of $\boldsymbol{a}(t)_{|t=n}$ for scenario 1.

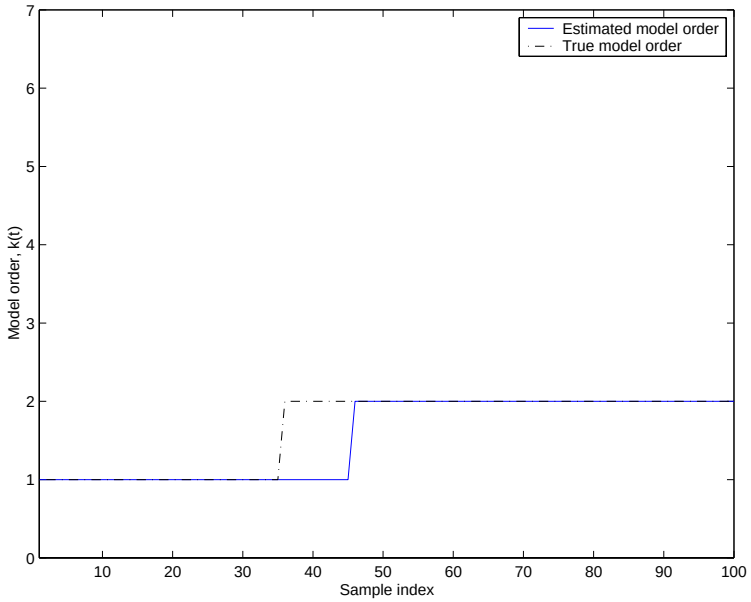

Figure 6.3: The tracking of the model order $k(t)_{|t=n}$ for scenario 1.

Time period	MSE (dB) ISD	MSE(dB) Amplitude
$n \in [1, 37]$	-27.4, –	-32.6, –
$n \in [38, 44]$	-26.3, –	-28.1, –
$n \in [45, 100]$	-33.0, -32.5	-16.5, -18.8

Table 6.2: The MSE between the true and estimated ISDs over 50 independent trials for the proposed method for the scenario 1.

6.5.2 Scenario 2: A source suddenly vanishes

In this scenario, one source suddenly vanishes during the ISD tracking of all sources. The initial ISD values of the sources are $\boldsymbol{\tau}(0) = [0.05, -0.65] \times T_s$. As the algorithm starts when $n \geq 1$, these two sources are detected, and their ISD trajectories and waveform recovery are shown in Figures 6.4 and 6.5, respectively. According to these figures, in the time period $n \in [1, 20]$, the algorithm tracks the ISDs and recovers their waveforms very well. At time $n > 20$, one of the sources, $s_1(n)$, the upper trajectory in Figure 6.6, suddenly disappears. It takes about 11 time steps for the algorithm to respond to this change and reduce the model order by 1. During the time period $n \in [21, 32]$, the ISD estimates and hence the restored waveform for the vanished source are essentially noise processes, whose variances are about σ_v^2 and σ_w^2, respectively. In the time period $n \in [33, 100]$, the algorithm tracks the ISD of the remaining source and recovers the waveform very well, as evident in Figures 6.4 and 6.5.

We also run 50 independent trials for the same setup to evaluate the mean-squared errors for the ISD estimation and the waveform recovery process for the sources. Table 6.3 shows the results. Similar to the findings in the last scenario, the MSEs for both the ISD estimation and waveform recovery are larger when the algorithm tracks two sources than that when it tracks one only source.

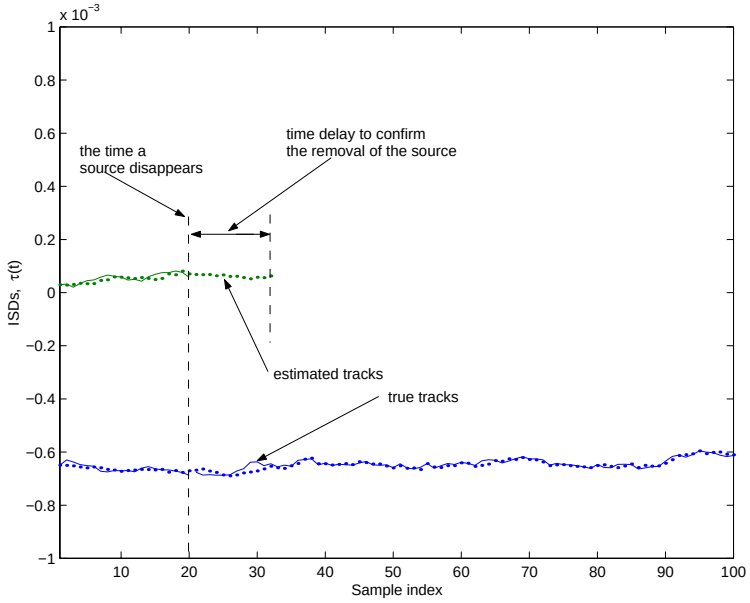

Figure 6.4: The tracking of ISDs $\boldsymbol{\tau}(t)_{|t=n}$ for scenario 2.

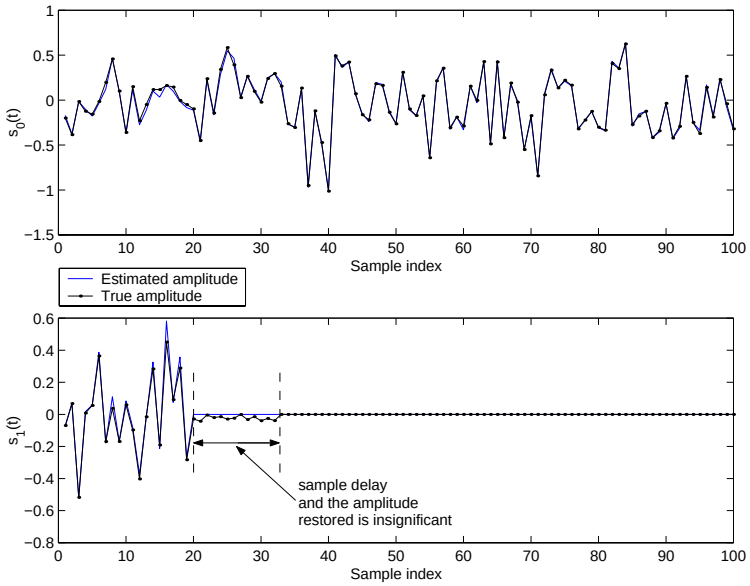

Figure 6.5: The signal recovery of $\boldsymbol{a}(t)_{|t=n}$ for scenario 2.

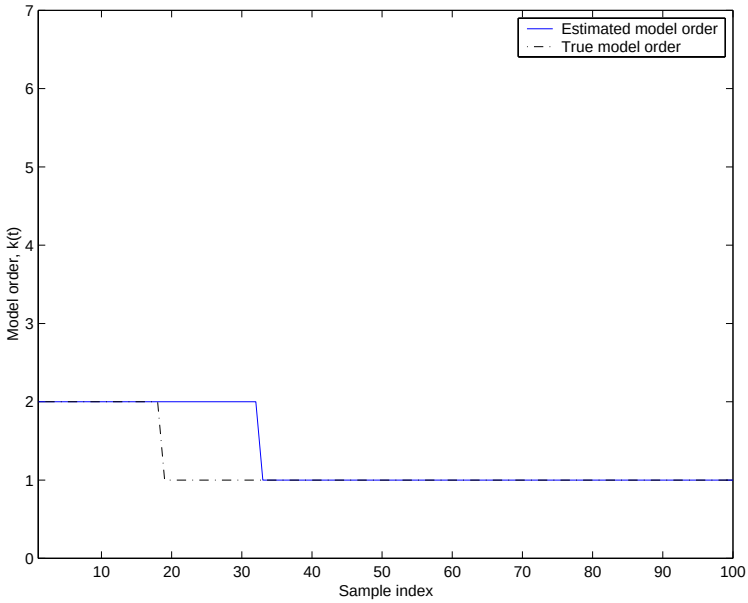

Figure 6.6: The tracking of the model order $k(t)_{|t=n}$ for scenario 2.

Time period	MSE (dB) ISD	MSE(dB) Amplitude
$n \in [1, 37]$	-29.4, -33.0	-22.6, -23.7
$n \in [38, 44]$	-33.1, –	-20.6, –
$n \in [45, 100]$	-34.7, –	-25.7, –

Table 6.3: The MSE between the true and estimated ISDs over 50 independent trials for the proposed method for the scenario 2.

6.6 Performance Evaluation

Since there appears to be no previous method which offers the same functionality as the proposed approach, it is difficult to compare performance with previous approaches. Therefore, to investigate the performance of the proposed algorithm, we resort to simulations to compare proximity of the results to lower bounds corresponding to optimum performance. Since the problem in question is nonstationary, we resort to the posterior Cramér-Rao bound (PCRB) (Tichavsky *et al.*, 1998), derived for discrete-time nonlinear dynamical systems. The development of the recursive update equation for the variances of the parameters can be found in Appendix E.

We now present the comparison between the performance of the proposed method in terms of the estimated variances of $\boldsymbol{\tau}(n)$ as a function of SNR and the PCRB. In this simulation, we use the parameter values listed in Table 6.1, except for SNR. We arbitrarily select a particular time sample at $n = 20$ for the evaluation of the variances of $\boldsymbol{\tau}(n)$. The algorithm is run for 100 independent trials over a range of SNR, from -5dB to 20 dB. Each trial uses 50 observations and the number of sources is assumed known. Figure 6.7 shows the comparision between the estimated variances and the PCRB. As shown in Fig. 6.7, for SNR levels lower than -2dB, the algorithm starts to break down (i.e., departs rapidly from the PCRB). However, it is seen that the variances approach the PCRB closely, above this level. The reasons why the variances do not come closer to the theoretical PCRB are: 1) interpolation errors due to a non-ideal interpolation function being used and 2) the procedure for estimating the source amplitudes. The errors introduced in the signal recovery have an impact on the estimation accuracy of the ISD parameters, since the source amplitudes $\boldsymbol{a}(n)$ affect the transformed data $\boldsymbol{z}(n)$, as evident from (3.47). The quantities $\boldsymbol{z}(n)$ are then used in estimating the $\boldsymbol{\tau}(n)$ through the posterior distribution (6.15).

We also include a few more figures to compare the performance of the algorithm with the PCRB for $n \in [1, 50]$ for different SNR levels, as shown in Figs. 6.8, 6.9, and 6.10. The trajectories of the PCRB and the estimated variances of the error in the estimated $\boldsymbol{\tau}(n)$ are obtained by running 1,000 trials. We see that as SNR levels decrease, the gap between

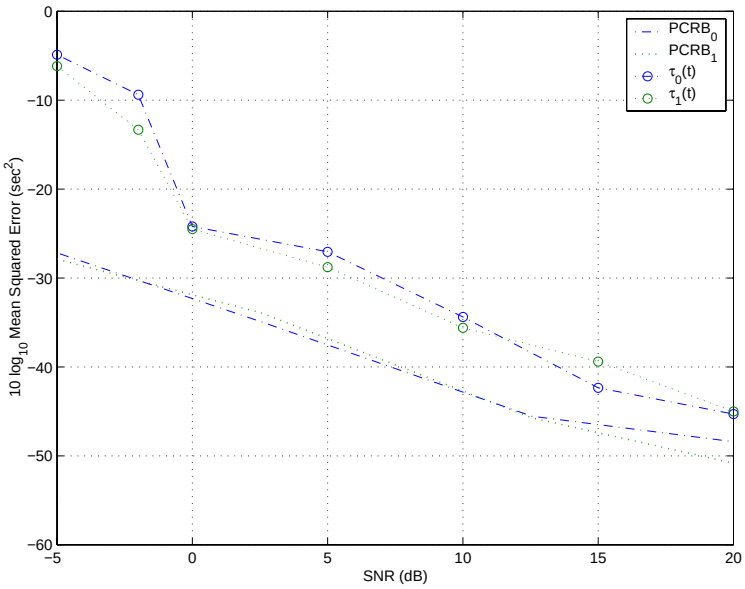

Figure 6.7: The performance of the algorithm as a function of SNR for the sample at $n = 20$.

the estimated variances and the PCRB widens, which is consistent with the findings in Fig. 6.7.

We also investigate the detection performance of the algorithm as a function of SNR level. Fig. 6.11 clearly reveals that the higher the SNR level, the higher is the probability of correct detection of the number of sources. In addition, given an SNR level, the performance is dependent on the number of particles used in the particle filter. The larger the number of particles, the better is the performance in model order detection. In other words, in order to have a consistent tracking peformance that heavily relies on the consistent model order detection, one needs to use a large number of particles in the algorithm.

6.7 Conclusion

A new approach to wideband array signal processing is proposed that is based on an interpolation process to approximate wideband signals. A Bayesian approach is used for estimation of the parameters of interest, where a marginalized posterior density function is formulated.

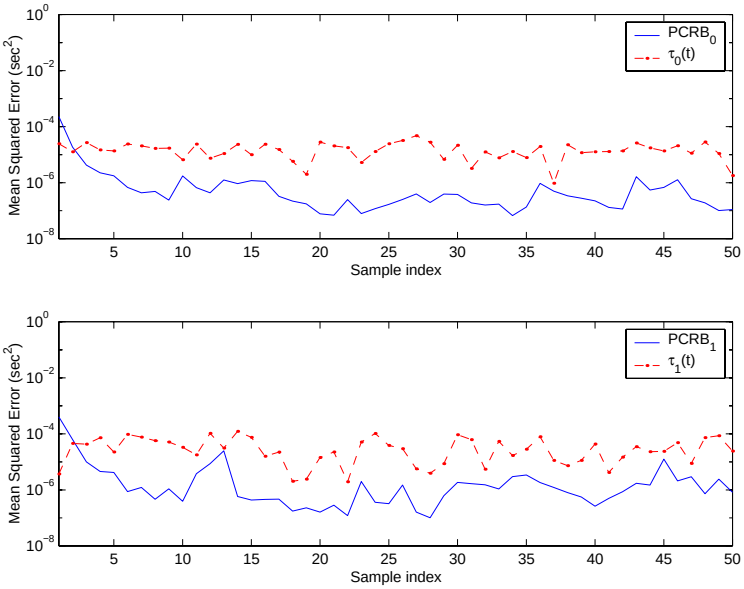

Figure 6.8: A comparison between the estimate variances of $\boldsymbol{\tau}(t)_{|t=n}$ and the PCRB for SNR = 15 dB.

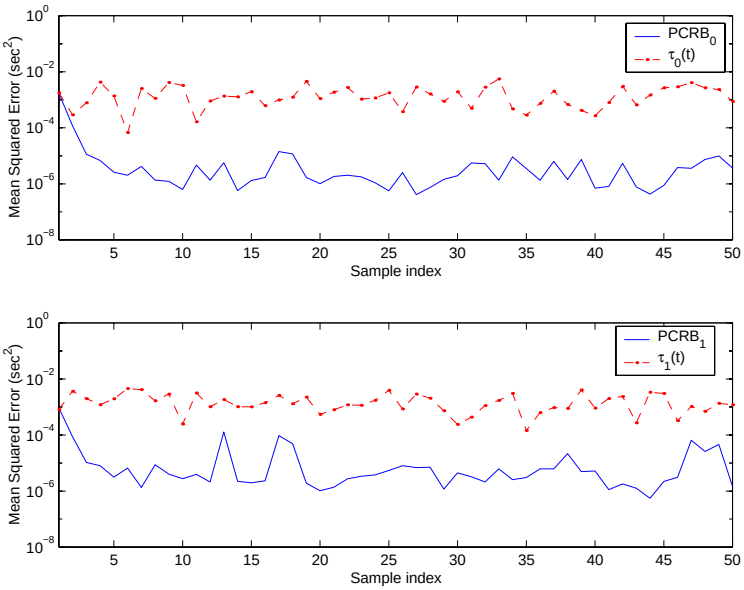

Figure 6.9: A comparison between the estimate variances of $\boldsymbol{\tau}(t)_{|t=n}$ and the PCRB for SNR = 0 dB.

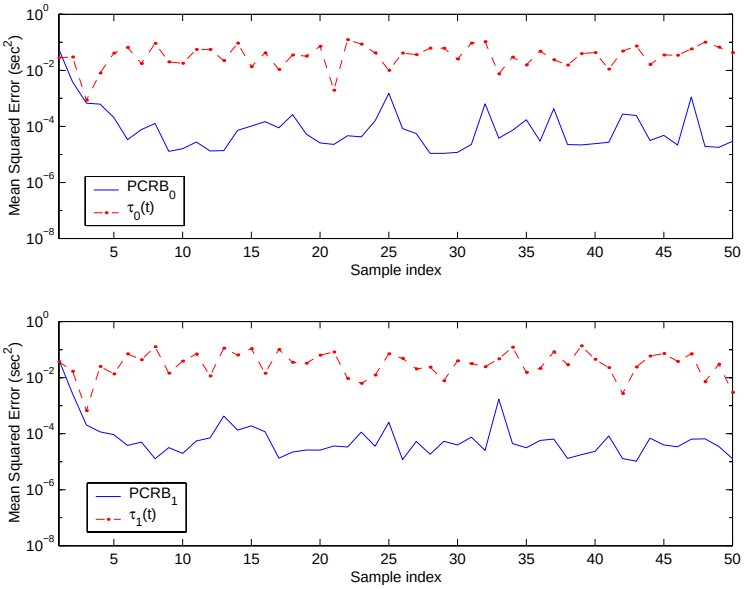

Figure 6.10: A comparison between the estimate variances of $\boldsymbol{\tau}(t)_{|t=n}$ and the PCRB for SNR $= -5$ dB.

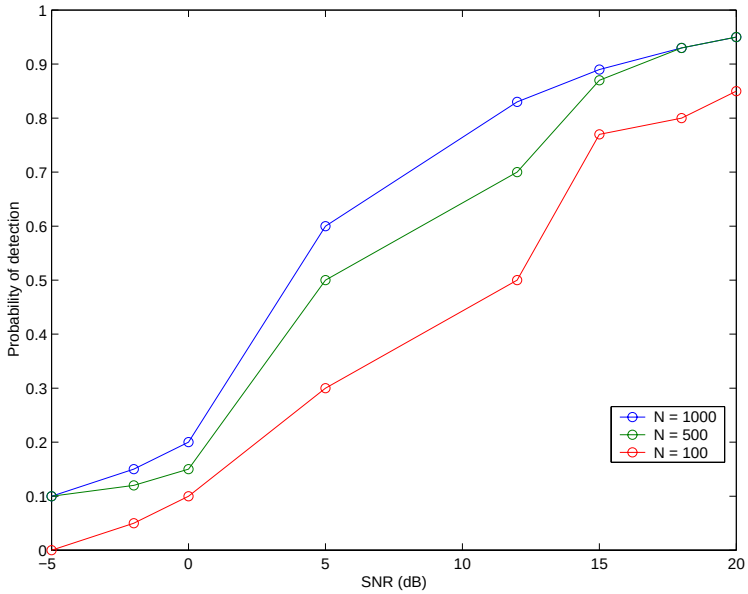

Figure 6.11: The probability of detection of the number of unknown sources versus SNR levels for different numbers of particles.

The time-varying ISDs are tracked by particle filters, for which a novel second-order importance sampling function is proposed. The model order is determined by a hypothesis testing procedure. The source amplitudes are estimated using a MAP estimate. Simulation results support the effectiveness of the method, and demonstrate reliable detection of the number of sources and estimation of their ISDs in white noise environments with a single linear array. As a result, the source signals are reliably restored. Furthermore, a theoretical lower bound on error variances (the posterior Cramér-Rao bound, PCRB) is developed to evaluate the performance of the proposed algorithm. It has been shown that the performance of the proposed method is comparable to that of the PCRB.

Chapter 7

Conclusions and Suggestions for Future Work

This thesis presented new approaches to various classical problems in array signal processing, using modern numerical Bayesian methods. We showed how the *Markov Chain Monte Carlo* (MCMC) and the *Sequential Importance Sampling* (SIS) methods present new outlooks and offer many advantages to problems in the field. In particular, we developed a new data structure that uses an interpolation function and the signal samples to approximate wideband signals. The transformed data from this data model allows the wideband signal processing to be performed in the time-domain with a fewer number of observations.

This thesis focused on joint detection and estimation problems in array signal processing context in an additive white noise scenario. When the parameters of interest are static throughout the observation, an *off-line* MCMC approach is employed, whereas an *on-line* approach using sequential MC methods is employed in the event that parameters of interest are time-varying. No existing methods have tackled this joint problem of detection of the model order and estimation of the parameters, particularly in wideband scenario, due to the difficult nature of the problem.

Four problems were addressed in the course of this thesis.

1. A novel data model that uses an interpolation function and the signal samples to

approximate both narrowband and wideband signals without change of structure or parameters was developed. The transformed observations from this model share analogous features as the narrowband data model. Also, all quantities in the model, including the data, are purely real, unlike other existing models which require complex quantities. This latter point leads to significant savings in hardware, since quadrature mixing to IF frequencies is no longer required, and computational requirements are reduced.

This data model is the basis on which the off-line and on-line algorithms for joint detection and estimation for wideband signals using MCMC and sequential MC techniques were developed, respectively.

2. The first algorithm of this thesis focused on the implementation of sequential MC methods for advanced beamforming for narrowband signals. Unlike conventional beamforming techniques that assume knowledge of the number of sources, and the sources to be static, the proposed algorithm is able to use sequential MC techniques to simultaneously track the time-varying number of sources and the motions of the incident angles of the sources. Based on the instantaneous estimates of the incident angles, the algorithm separates and restores the signal amplitudes.

3. The second algorithm of this thesis focused on solving a more difficult and cumbersome array signal processing problem for wideband signals. Unlike other conventional approaches that solve the problem in the frequency domain and require a huge number of observations to sustain the computational accuracy, the proposed algorithm applies the reversible jump MCMC method on the novel data model developed in this thesis in the time domain.

   The method uses a single array of sensors and analytically integrates out the signal amplitudes and the unknown noise variances to leave a marginalized posterior distribution that is only a function of the parameters of interest.

The performance of the method was successfully compared with the theoretical Cramér-Rao bound and other statistical tests.

4. The final algorithm was an extension of the first and the second approaches in jointly detecting the number of wideband sources and tracking their motions. The proposed method first applied rational statistical testing to determine the number of unknown sources, and then applied the sequential MC technique on the modified novel data model for tracking the time-varying parameter.

   The performance of the method was successfully compared with the posterior Cramér-Rao bound.

7.1 Contributions to the scientific literature

The work presented in Chapters 3 to 6 has been published in various conferences and on a contributed chapter. In addition, four journal papers have been submitted for publication.

- **Journal Papers**
  - *Wideband Array Signal Processing Using Sequential MCMC Methods* (Ng *et al.*, 2003e)
  - *Sequential MCMC for Signal recovery of Multiple Nonstationary Targets Using an Array of Sensors* (Ng *et al.*, 2003a)
  - *Wideband Array Signal Processing Using MCMC Methods* (Ng *et al.*, 2002a)
  - *Particle filter for tracking an unknown number of sources* (Larocque *et al.*, 2002) IEEE Transactions on Signal Processing, December, 2002.
- **Conference Papers**
  - *A Bayesian Approach to Tracking Wideband Targets Using Sensor Arrays and Particle Filters.* (Ng, Reilly and Kirubarajan) (Ng *et al.*, 2003c) SSP 2003.

  - *Application of Particle Filters for Tracking Moving Receivers in Wireless Communication Systems* (Ng, Reilly and Kirubarajan) (Ng *et al.*, 2003b) SPAWC 2003.
  - *Wideband Array Signal Processing Using MCMC Methods.* (Ng, Reilly, Kirubarajan and Larocque) (Ng *et al.*, 2003d) ICASSP 2003.
  - *Wideband Array Signal Processing Using MCMC Methods.* (Ng, Reilly, Kirubarajan and Larocque) (Ng *et al.*, 2002b) SAM 2002.
  - *Sequential Monte Carlo for Spatial Signal Separation and Restoration.* (Ng, Reilly and Larocque) (Ng *et al.*, 2001b) 2001 Workshop on Maximum Entropy and Bayesian Methods in Science and Engineering.
  - *On the Implementation of Particle Filters for DOA Tracking.* (Ng *et al.*, 2001a) Lecture at ICASSP 2001.

- **Contributed Chapters**
  - *Sequential MCMC for Spatial Signal Separation and Restoration From An Array of Sensors.* (Ng *et al.*, 2001b)

7.2 Future Work

In this thesis, the focus is to apply numerical Bayesian techniques to solve array signal processing problems in both static and nonstatic scenarios. There are a few areas that are worth being explored in the future. They are

- Verification of the algorithms with the real arrays and real-life data
- Improvment of the performance of the algorithms

7.3 Verification of the algorithms with the real arrays and real-life data

Throughout the entire research, only simulated data were used in the development of the algorithms and all results were based on these simulated data. While the performance of these algorithms was consistent with the theoretical tests, including the Cramér-Rao lower bound and other statistical tests, real-life data should be collected and used to verify their practical performance.

In addition, the array sensor elements in the entire research were assumed appropriately calibrated and identical. Robustness of the algorithms to these assumptions should be conducted as well.

7.4 Improvment of the performance of the algorithms

There are a few areas to improve the algorithms in this thesis. For the MCMC algorithm for wideband array signal processing, a half-sinc function was used to enable zero-delay processing. According to Figure 5.11, there was some performance gap between the algorithm and the theoretical CRLB. One possible area that contributes to the gap is the use of non-ideal interpolation functions and the half-sinc approach as described in Chapters 3 and 5. Modification of the algorithm that can address the buffering of the extra samples should be explored, and results can be compared.

For the sequential approach, one of the bottlenecks in using the algorithm in practice is the number of particles used. It is an issue of trading-off between accuracy of tracking the time-varying posterior distribution function of the states and the time needed to respond to the state changes. Throughout the research, no efforts were spent on finding an intelligent important sampling procedure that may require much fewer number of particles in the process. Therefore, investigation in this area could help reduce this bottleneck.

Appendix A

Probability Density Functions

Definitions of selected probability functions that appear in the thesis (Bernardo and Smith, 1994).

Name	Symbol	Functional Form	c
Real Normal	$\mathcal{N}(\mathbf{x}\|\mathbf{m},\mathbf{\Sigma})$	$c\exp(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{m}))$	$\|2\pi\mathbf{\Sigma}\|^{-1/2}$
Inverted Gamma	$\mathcal{IG}(x\|\alpha,\beta)$	$cx^{-\alpha-1}\exp(-\beta/x)\mathbb{I}_{[0,+\infty)}(x)$	$\frac{\beta^\alpha}{\Gamma(\alpha)}$
Poisson	$\mathcal{P}(x\|\lambda)$	$c\frac{\lambda^x}{x!}\mathbb{I}_{\mathbb{N}}(x)$	$\exp(-\lambda)$
Chi-squared	$\chi^2(x\|\nu)$	$cx^{\nu/2-1}\exp(-x/2)\mathbb{I}_{[0,+\infty)}(x)$	$\frac{2^{-\nu/2}}{\Gamma(\nu/2)}$
F	$\mathcal{F}(x\|\alpha,\beta)$	$c\frac{x^{\alpha/2-1}}{(\beta+\alpha x)^{(\alpha+\beta)/2}}\mathbb{I}_{(0,+\infty)}(x)$	$\frac{\Gamma((\alpha+\beta)/2)}{\Gamma(\alpha/2)\Gamma(\beta/2)}\alpha^{\alpha/2}\beta^{\beta/2}$

Table A.1: Definition of selected probability density functions

Appendix B

Recursivity of Particle Filters

This appendix shows how to obtain the recursive form of the posterior equation (2.37)

$$p(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})\frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_{1:t})}.$$

We start with the application of Bayes' theorem,

$$p(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1}) = \frac{p(\boldsymbol{y}_{1:t+1}|\boldsymbol{x}_{1:t+1})p(\boldsymbol{x}_{1:t+1})}{p(\boldsymbol{y}_{1:t+1})}.$$

Since the observation noise is assumed *iid*, the conditional terms are assumed independent and the total likelihood term can be expanded one step into

$$p(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1}) = \frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{1:t})p(\boldsymbol{x}_{1:t+1})}{p(\boldsymbol{y}_{1:t+1})}.$$

Applying once more Bayes' theorem on the diminished total likelihood term, we get,

$$p(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})\frac{p(\boldsymbol{y}_{1:t})p(\boldsymbol{x}_{1:t+1})}{p(\boldsymbol{y}_{1:t+1})p(\boldsymbol{x}_{1:t})}.$$

Expanding the total density function of $\boldsymbol{x}$ as (also for $\boldsymbol{y}$),

$$p(\boldsymbol{x}_{1:t+1}) = p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{1:t})p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1})\cdots p(\boldsymbol{x}_o),$$

and simplifying the common terms in the numerator and denominator, we obtain

$$p(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})\frac{p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{1:t})}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_{1:t})}.$$

Applying the first order Markovian property of the model, the previous equation simplifies to

$$p(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})\frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_t)},$$

which is the desired recursive form. QED.

When the proposal function satisfies the following recursivity condition

$$q(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1}) = q(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t+1}),$$

the importance weights can be evaluated recursively. The importance weights are defined as

$$\tilde{\boldsymbol{w}}(t+1) = \frac{p(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1})}{q(\boldsymbol{x}_{1:t+1}|\boldsymbol{y}_{1:t+1})}.$$

Replacing the distributions by their corresponding definitions, we get

$$\tilde{\boldsymbol{w}}(t+1) = \frac{p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_t)q(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t+1})}.$$

Recognizing that

$$\tilde{\boldsymbol{w}}(t) = \frac{p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})}{q(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})},$$

we can obtain the recursive form of the unnormalized importance weights as,

$$\tilde{\boldsymbol{w}}(t+1) = \boldsymbol{w}(t)\frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t+1})}.$$

Appendix C

Derivation of The CRLB

Let $\boldsymbol{\theta}$ be the parameter vector defined as follows:

$$\boldsymbol{\theta} \triangleq \{\tau_k, \boldsymbol{s}_k(n)\}, \tag{C.1}$$

where $k = 0, ..., K-1$ and $n = 1, ..., N$. Given $\boldsymbol{Y} = \{\boldsymbol{y}(n), n = 1, ..., N\}$, where $\boldsymbol{y}(n)$ is defined as in (3.38), we define the likelihood function, $l(\boldsymbol{\theta})$, as:

$$\begin{aligned} l(\boldsymbol{\theta}) &= p\left(\boldsymbol{Y}|\boldsymbol{\theta}\right), \\ &= \prod_{n=1}^{N} \frac{1}{(2\pi\sigma_w^2)^{M/2}} \exp\left\{ \frac{-1}{2\sigma_w^2} \left(\boldsymbol{y}(n) - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}) \boldsymbol{a}(n-l) \right)^T \left(\boldsymbol{y}(n) - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}) \boldsymbol{a}(n-l) \right) \right\}, \end{aligned} \tag{C.2}$$

where $\tilde{\boldsymbol{H}}_l(\tau_k) \in \mathcal{R}^{M \times 1}$ is

$$\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}) \triangleq \left[\tilde{\boldsymbol{H}}_l(\tau_0), \tilde{\boldsymbol{H}}_l(\tau_1), \ldots, \tilde{\boldsymbol{H}}_l(\tau_{K-1}) \right], \tag{C.3}$$

and $\boldsymbol{a}(n-l) \in \mathcal{R}^{K \times 1}$ is

$$\boldsymbol{a}(n-l) \triangleq \left[s_0(n-l), s_1(n-l), \ldots, s_{K-1}(n-l) \right]^T. \tag{C.4}$$

Defining $\boldsymbol{u}(n)$ as follows

$$\boldsymbol{u}(n) \triangleq \boldsymbol{y}(n) - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}) \boldsymbol{a}(n-l) \tag{C.5}$$

and taking the natural logarithm of (C.2) yields

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &\triangleq \ln\left[l(\boldsymbol{\theta})\right], \\ &= -\frac{1}{2\sigma_w^2}\sum_{n=1}^{N}\boldsymbol{u}^T(n)\boldsymbol{u}(n) - \frac{MN}{2}\ln\left(2\pi\sigma_w^2\right). \end{aligned} \tag{C.6}$$

To obtain the CRLB, we need the Fisher Information Matrix (Bar-Shalom *et al.*, 2001), $\boldsymbol{\mathcal{J}}(\boldsymbol{\theta}) \in \mathcal{R}^{(KN+K)\times(KN+K)}$, which is defined as

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\theta}) = -\begin{bmatrix} E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_0\partial\boldsymbol{\tau}}\right] & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_0\partial\boldsymbol{s}_0(n)}\right] & \cdots & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_0\partial\boldsymbol{s}_{K-1}(n)}\right] \\ \vdots & \vdots & & \vdots \\ E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_{K-1}\partial\boldsymbol{\tau}}\right] & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_{K-1}\partial\boldsymbol{s}_0(n)}\right] & \cdots & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_{K-1}\partial\boldsymbol{s}_{K-1}(n)}\right] \\ E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\boldsymbol{s}_0(n)\partial\boldsymbol{\tau}}\right] & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\boldsymbol{s}_0(n)\partial\boldsymbol{s}_0(n)}\right] & \cdots & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\boldsymbol{s}_0(n)\partial\boldsymbol{s}_{K-1}(n)}\right] \\ \vdots & \vdots & & \vdots \\ E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\boldsymbol{s}_{K-1}(n)\partial\boldsymbol{\tau}}\right] & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\boldsymbol{s}_{K-1}(n)\partial\boldsymbol{s}_0(n)}\right] & \cdots & E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\boldsymbol{s}_{K-1}(n)\partial\boldsymbol{s}_{K-1}(n)}\right] \end{bmatrix}, \tag{C.7}$$

where $E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_k\partial\boldsymbol{\tau}}\right] \in \mathcal{R}^{1\times K}$, $E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial\boldsymbol{s}_p(n)}\right] \in \mathcal{R}^{1\times N}$, and $E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_p\partial\boldsymbol{s}_k(n)}\right] \in \mathcal{R}^{1\times N}$ are defined, respectively, as

$$E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_k\partial\boldsymbol{\tau}}\right] \triangleq \left[E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_k\partial\tau_0}\right], E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_k\partial\tau_1}\right], \ldots, E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_k\partial\tau_{K-1}}\right]\right], \tag{C.8}$$

$$E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial\boldsymbol{s}_p(n)}\right] \triangleq \left[E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial s_p(1)}\right], E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial s_p(2)}\right], \ldots, E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial s_p(N)}\right]\right], \tag{C.9}$$

$$E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_p\partial\boldsymbol{s}_k(n)}\right] \triangleq \left[E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_p\partial s_k(1)}\right], E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_p\partial s_k(2)}\right], \ldots, E\left[\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial\tau_p\partial s_k(N)}\right]\right]. \tag{C.10}$$

The derivations of the elememts in $\boldsymbol{\mathcal{J}}(\boldsymbol{\theta})$ will be presented below.

C.1 Derivation of $\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\tau_k}$

$$\begin{aligned}\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\tau_k} &= -\frac{1}{2\sigma_w^2}\left\{\frac{\partial}{\partial\tau_k}\sum_{n=1}^{N}\boldsymbol{u}^T(n)\boldsymbol{u}(n)\right\}, \\ &= -\frac{1}{2\sigma_w^2}\left\{\sum_{n=1}^{N}\left(\frac{\partial\boldsymbol{u}(n)}{\partial\tau_k}\right)^T\boldsymbol{u}(n) + \sum_{n=1}^{N}\boldsymbol{u}^T(n)\left(\frac{\partial\boldsymbol{u}(n)}{\partial\tau_k}\right)\right\}, \end{aligned} \tag{C.11}$$

where

$$\begin{aligned}
\frac{\partial \boldsymbol{u}(n)}{\partial \tau_k} &= \frac{\partial}{\partial \tau_k}\left(\boldsymbol{y}(n) - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l)\right), \\
&= -\sum_{l=0}^{L-1} \frac{\partial \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})}{\partial \tau_k}\boldsymbol{a}(n-l), \\
&= -\sum_{l=0}^{L-1} \frac{\partial}{\partial \tau_k}\left[\tilde{\boldsymbol{H}}_l(\tau_0), \ldots, \tilde{\boldsymbol{H}}_l(\tau_k), \ldots, \tilde{\boldsymbol{H}}_l(\tau_{K-1})\right] \begin{bmatrix} s_0(n-l) \\ \vdots \\ s_k(n-l) \\ \vdots \\ s_{K-1}(n-l) \end{bmatrix}, \\
&= -\sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l^{'}(\tau_k) s_k(n-l), \\
&= -\tilde{\boldsymbol{H}}^{'}(\tau_k)\boldsymbol{s_k}(n), \qquad \text{(C.12)}
\end{aligned}$$

where

$$\tilde{\boldsymbol{H}}_l^{'}(\tau_k) \triangleq \frac{\partial \tilde{\boldsymbol{H}}_l(\tau_k)}{\partial \tau_k}, \qquad \text{(C.13)}$$

$$\begin{aligned}
\tilde{\boldsymbol{H}}^{'}(\tau_k) &= \left[\tilde{\boldsymbol{H}}_0^{'}(\tau_k), \tilde{\boldsymbol{H}}_1^{'}(\tau_k), \ldots, \tilde{\boldsymbol{H}}_{L-1}^{'}(\tau_k)\right], \qquad \text{(C.14)} \\
\boldsymbol{s}_k(n) &= [s_k(n), s_k(n-1), \ldots, s_k(n-L+1)]^T, \qquad \text{(C.15)}
\end{aligned}$$

and $\mathbf{0} \in \mathcal{R}^{M\times 1}$ is a column vector of zeros. If sinc function is used as the interpolation function in $\tilde{\boldsymbol{H}}_l(\tau_k)$, the m, lth element in $\tilde{\boldsymbol{H}}^{'}(\tau_k)$ is defined as in (C.39). Accordingly, (C.11) becomes:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \tau_k} &= -\frac{1}{2\sigma_w^2}\left\{\sum_{n=1}^{N}\left(\frac{\partial \boldsymbol{u}(n)}{\partial \tau_k}\right)^T \boldsymbol{u}(n) + \sum_{n=1}^{N} \boldsymbol{u}^T(n)\left(\frac{\partial \boldsymbol{u}(n)}{\partial \tau_k}\right)\right\}, \\
&= -\frac{1}{\sigma_w^2}\sum_{n=1}^{N}\left(-\tilde{\boldsymbol{H}}^{'}(\tau_k)\boldsymbol{s_k}(n)\right)^T \boldsymbol{u}(n), \\
&= \frac{1}{\sigma_w^2}\sum_{n=1}^{N} \boldsymbol{s}_k^T(n)\boldsymbol{g}_k(n), \qquad \text{(C.16)}
\end{aligned}$$

where

$$\boldsymbol{g}_k(n) \triangleq \left(\tilde{\boldsymbol{H}}^{'}(\tau_k)\right)^T \boldsymbol{u}(n). \tag{C.17}$$

C.2 Derivation of $\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial s_k(n)}$

$$\begin{aligned}
\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial s_k(n)} &= -\frac{1}{2\sigma_w^2}\left\{\frac{\partial}{\partial s_k(n)}\sum_{n=1}^{N}\boldsymbol{u}^T(n)\boldsymbol{u}(n)\right\}, \\
&= -\frac{1}{2\sigma_w^2}\left\{\sum_{n=1}^{N}\left(\frac{\partial \boldsymbol{u}(n)}{\partial s_k(n)}\right)^T\boldsymbol{u}(n) + \sum_{n=1}^{N}\boldsymbol{u}^T(n)\left(\frac{\partial \boldsymbol{u}(n)}{\partial s_k(n)}\right)\right\},
\end{aligned} \tag{C.18}$$

where

$$\begin{aligned}
\frac{\partial \boldsymbol{u}(n)}{\partial s_k(n)} &= \frac{\partial}{\partial s_k(n)}\left(\boldsymbol{y}(n) - \sum_{l=0}^{L-1}\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l)\right), \\
&= -\frac{\partial}{\partial s_k(n)}\left\{\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n) + \sum_{l=1}^{L-1}\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau})\boldsymbol{a}(n-l)\right\}, \\
&= -\frac{\partial}{\partial s_k(n)}\left\{\tilde{\boldsymbol{H}}_0(\boldsymbol{\tau})\boldsymbol{a}(n)\right\}, \\
&= -1 \times \left[\tilde{\boldsymbol{H}}_0(\tau_0), \ldots, \tilde{\boldsymbol{H}}_0(\tau_k), \ldots, \tilde{\boldsymbol{H}}_0(\tau_{K-1})\right]\frac{\partial}{\partial s_k(n)}\left\{\begin{bmatrix} s_0(n) \\ \vdots \\ s_k(n) \\ \vdots \\ s_{K-1}(n) \end{bmatrix}\right\}, \\
&= -\tilde{\boldsymbol{H}}_0(\tau_k).
\end{aligned} \tag{C.19}$$

Substituting (C.19) into (C.18), we have:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial s_k(n)} &= -\frac{1}{2\sigma_w^2}\left\{\sum_{n=1}^{N}\left(\frac{\partial \boldsymbol{u}(n)}{\partial s_k(n)}\right)^T\boldsymbol{u}(n) + \sum_{n=1}^{N}\boldsymbol{u}^T(n)\left(\frac{\partial \boldsymbol{u}(n)}{\partial s_k(n)}\right)\right\}, \\
&= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\boldsymbol{u}(n).
\end{aligned} \tag{C.20}$$

C.3 Derivation of $\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial \tau_k \partial \tau_p}$

1. If $k = p$, we have

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial \tau_k^2} &= \frac{1}{\sigma_w^2} \frac{\partial}{\partial \tau_k} \sum_{n=1}^{N} \boldsymbol{s}_k^T(n) \boldsymbol{g}_k(n), \\
&= \frac{1}{\sigma_w^2} \sum_{n=1}^{N} \boldsymbol{s}_k^T(n) \left(\frac{\partial \boldsymbol{g}_k(n)}{\partial \tau_k} \right), \\
&= \frac{1}{\sigma_w^2} \sum_{n=1}^{N} \sum_{l=0}^{L-1} \frac{\partial g_k^l(n)}{\partial \tau_k} s_k(n-l), \qquad \text{(C.21)}
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial g_k^l(n)}{\partial \tau_k} &= \left(\frac{\partial \tilde{\boldsymbol{H}}_l^{'}(\tau_k)}{\partial \tau_k} \right)^T \boldsymbol{u}(n) + \left(\tilde{\boldsymbol{H}}_l^{'}(\tau_k) \right)^T \frac{\partial \boldsymbol{u}(n)}{\partial \tau_k}, \\
&= \left(\tilde{\boldsymbol{H}}_l^{''}(\tau_k) \right)^T \boldsymbol{u}(n) - \left(\tilde{\boldsymbol{H}}_l^{'}(\tau_k) \right)^T \tilde{\boldsymbol{H}}^{'}(\tau_k) \boldsymbol{s}_k(n), \qquad \text{(C.22)}
\end{aligned}$$

where

$$\tilde{\boldsymbol{H}}_l^{''}(\tau_k) \triangleq \frac{\partial^2 \tilde{\boldsymbol{H}}_l(\tau_k)}{\partial \tau_k^2}, \qquad \text{(C.23)}$$

where the mth element in $\tilde{\boldsymbol{H}}_l^{''}(\tau_k)$ is defined as in (C.40) if sinc function is used. Accordingly, (C.21) becomes:

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial \tau_k^2} &= \frac{1}{\sigma_w^2} \sum_{n=1}^{N} \sum_{l=0}^{L-1} \left\{ \left(\tilde{\boldsymbol{H}}_l^{''}(\tau_k) \right)^T \boldsymbol{u}(n) - \left(\tilde{\boldsymbol{H}}_l^{'}(\tau_k) \right)^T \tilde{\boldsymbol{H}}^{'}(\tau_k) \boldsymbol{s}_k(n) \right\} s_k(n-l), \\
&= \frac{1}{\sigma_w^2} \sum_{n=1}^{N} \sum_{l=0}^{L-1} \left(\tilde{\boldsymbol{H}}_l^{''}(\tau_k) \right)^T \boldsymbol{u}(n) s_k(n-l) - \\
&\quad \frac{1}{\sigma_w^2} \sum_{n=1}^{N} \sum_{l=0}^{L-1} s_k(n-l) \left(\tilde{\boldsymbol{H}}_l^{'}(\tau_k) \right)^T \tilde{\boldsymbol{H}}^{'}(\tau_k) \boldsymbol{s}_k(n), \\
&= \frac{1}{\sigma_w^2} \sum_{n=1}^{N} \left\{ \boldsymbol{u}^T(n) \tilde{\boldsymbol{H}}^{''}(\tau_k) \boldsymbol{s}_k(n) - \boldsymbol{s}_k(n) \left(\tilde{\boldsymbol{H}}^{'}(\tau_k) \right)^T \tilde{\boldsymbol{H}}^{'}(\tau_k) \boldsymbol{s}_k(n) \right\}, \\
&= \frac{1}{\sigma_w^2} \left\{ \sum_{n=1}^{N} \boldsymbol{u}^T(n) \tilde{\boldsymbol{H}}^{''}(\tau_k) \boldsymbol{s}_k(n) - \mathrm{Tr} \left[\mathcal{H}^{'}(\tau_k, \tau_k) \boldsymbol{R}_{kk}(n) \right] \right\}, \qquad \text{(C.24)}
\end{aligned}$$

where

$$\mathrm{tr}\left[\mathcal{H}^{'}(\tau_k,\tau_k)\boldsymbol{R}_{kk}(n)\right] \triangleq \sum_{n=1}^{N}\boldsymbol{s}_k(n)\left(\tilde{\boldsymbol{H}}^{'}(\tau_k)\right)^T\tilde{\boldsymbol{H}}^{'}(\tau_k)\boldsymbol{s}_k(n), \tag{C.25}$$

and

$$\begin{aligned}\mathcal{H}^{'}(\tau_k,\tau_k) &= \left(\tilde{\boldsymbol{H}}^{'}(\tau_k)\right)^T\tilde{\boldsymbol{H}}^{'}(\tau_k), &\text{(C.26)}\\ \boldsymbol{R}_{kk}(n) &= \sum_{n=1}^{N}\boldsymbol{s}_k(n)\boldsymbol{s}_k^T(n). &\text{(C.27)}\end{aligned}$$

2. If $k \neq p$, we have

$$\begin{aligned}\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial\tau_k\partial\tau_p} &= \frac{1}{\sigma_w^2}\frac{\partial}{\partial\tau_p}\sum_{n=1}^{N}\boldsymbol{s}_k^T(n)\boldsymbol{g}_k(n),\\ &= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\boldsymbol{s}_k^T(n)\left(\frac{\partial\boldsymbol{g}_k(n)}{\partial\tau_p}\right),\\ &= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\sum_{l=0}^{L-1}\frac{\partial g_k^l(n)}{\partial\tau_p}s_k(n-l), &\text{(C.28)}\end{aligned}$$

where

$$\begin{aligned}\frac{\partial g_k^l(n)}{\partial\tau_p} &= \left(\tilde{\boldsymbol{H}}_l^{'}(\tau_k)\right)^T\frac{\partial\boldsymbol{u}(n)}{\partial\tau_p},\\ &= -\left(\tilde{\boldsymbol{H}}_l^{'}(\tau_k)\right)^T\tilde{\boldsymbol{H}}^{'}(\tau_p)\boldsymbol{s}_p(n). &\text{(C.29)}\end{aligned}$$

Substituting (C.29) into (C.28) yields:

$$\begin{aligned}\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial\tau_k\partial\tau_p} &= \frac{-1}{\sigma_w^2}\sum_{n=1}^{N}\sum_{l=0}^{L-1}s_k(n-l)\left(\tilde{\boldsymbol{H}}_l^{'}(\tau_k)\right)^T\tilde{\boldsymbol{H}}^{'}(\tau_p)\boldsymbol{s}_p(n),\\ &= \frac{-1}{\sigma_w^2}\sum_{n=1}^{N}\boldsymbol{s}_k(n)\left(\tilde{\boldsymbol{H}}^{'}(\tau_k)\right)^T\tilde{\boldsymbol{H}}^{'}(\tau_p)\boldsymbol{s}_p(n),\\ &= \frac{-1}{\sigma_w^2}\mathrm{tr}\left[\mathcal{H}^{'}(\tau_k,\tau_p)\boldsymbol{R}_{kp}(n)\right], &\text{(C.30)}\end{aligned}$$

where

$$\mathrm{tr}\left[\mathcal{H}^{'}(\tau_k,\tau_p)\boldsymbol{R}_{kp}(n)\right] \triangleq \sum_{n=1}^{N}\boldsymbol{s}_k(n)\left(\tilde{\boldsymbol{H}}^{'}(\tau_k)\right)^T\tilde{\boldsymbol{H}}^{'}(\tau_p)\boldsymbol{s}_p(n), \tag{C.31}$$

and

$$\mathcal{H}^{'}(\tau_k,\tau_p) = \left(\tilde{\boldsymbol{H}}^{'}(\tau_k)\right)^T \tilde{\boldsymbol{H}}^{'}(\tau_p), \tag{C.32}$$

$$\boldsymbol{R}_{kp}(n) = \sum_{n=1}^{N} \boldsymbol{s}_p(n)\boldsymbol{s}_k^T(n). \tag{C.33}$$

C.4 Derivation of $\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial s_k(n)\partial s_p(n)}$

1. If $k = p$

$$\begin{aligned}
\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial s_k(n)} &= \frac{1}{\sigma_w^2}\frac{\partial}{\partial s_k(n)}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\boldsymbol{u}(n),\\
&= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\frac{\partial\boldsymbol{u}(n)}{\partial s_k(n)},\\
&= \frac{-1}{\sigma_w^2}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\tilde{\boldsymbol{H}}_0(\tau_k),\\
&= \frac{-N}{\sigma_w^2}\tilde{\boldsymbol{H}}_0^T(\tau_k)\tilde{\boldsymbol{H}}_0(\tau_k).
\end{aligned} \tag{C.34}$$

2. If $k \neq p$

$$\begin{aligned}
\frac{\partial\mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial s_p(n)} &= \frac{1}{\sigma_w^2}\frac{\partial}{\partial s_p(n)}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\boldsymbol{u}(n),\\
&= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\frac{\partial\boldsymbol{u}(n)}{\partial s_p(n)},\\
&= \frac{-1}{\sigma_w^2}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\tilde{\boldsymbol{H}}_0(\tau_p),\\
&= \frac{-N}{\sigma_w^2}\tilde{\boldsymbol{H}}_0^T(\tau_k)\tilde{\boldsymbol{H}}_0(\tau_p),
\end{aligned} \tag{C.35}$$

C.5 Derivation of $\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial s_k(n)\partial \tau_p(n)}$

1. $k = p$

$$\begin{aligned}\frac{\partial \mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial \tau_k} &= \frac{1}{\sigma_w^2}\frac{\partial}{\partial \tau_k}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\boldsymbol{u}(n),\\ &= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\left\{\left(\frac{\partial \tilde{\boldsymbol{H}}_0(\tau_k)}{\partial \tau_k}\right)^T\boldsymbol{u}(n)+\tilde{\boldsymbol{H}}_0^T(\tau_k)\frac{\partial \boldsymbol{u}(n)}{\partial \tau_k}\right\},\\ &= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\left\{\left(\tilde{\boldsymbol{H}}_0^{'}(\tau_k)\right)^T\boldsymbol{u}(n)-\tilde{\boldsymbol{H}}_0^T(\tau_k)\tilde{\boldsymbol{H}}^{'}(\tau_k)\boldsymbol{s_k}(n)\right\}. \quad \text{(C.36)}\end{aligned}$$

2. $k \neq p$

$$\begin{aligned}\frac{\partial \mathcal{L}^2(\boldsymbol{\theta})}{\partial s_k(n)\partial \tau_p} &= \frac{1}{\sigma_w^2}\frac{\partial}{\partial \tau_p}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\boldsymbol{u}(n),\\ &= \frac{1}{\sigma_w^2}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\frac{\partial \boldsymbol{u}(n)}{\partial \tau_p},\\ &= \frac{-1}{\sigma_w^2}\sum_{n=1}^{N}\tilde{\boldsymbol{H}}_0^T(\tau_k)\tilde{\boldsymbol{H}}^{'}(\tau_p)\boldsymbol{s_p}(n). \quad \text{(C.37)}\end{aligned}$$

Substituting equations (C.24), (C.30), (C.34), (C.35), (C.36), and (C.37), respectively, for $k, p = 0, 1, \ldots, K + KN - 1$ into the matrix $\boldsymbol{\mathcal{J}}(\boldsymbol{\theta})$ in (C.7), and defining the inverse of the resulting matrix, we can obtain the Cramér-Rao Lower Bound of the estimates in $\boldsymbol{\theta}$.

C.6 Derivatives of the interpolation function

Given the sampling instant, T_s, and cutoff frequency, f_c, the sinc function with L taps is defined as

$$h_l(m\tau_k) \triangleq \frac{\sin \pi f_c(t_l - m\tau_k)}{\pi f_c(t_l - m\tau_k)}, \qquad l = 0, 1, ..., L-1, \quad \text{(C.38)}$$

where $t_l = lT_s$ and $m\tau_k$ is the time delay. The first derivative of $h_l(m\tau_k)$ with respect to τ_k is given as follows

$$\begin{aligned} h_l^{'}(m\tau_k) &\triangleq \frac{dh_l(m\tau_k)}{d\tau_k}, \\ &= \frac{d}{d\tau_k}\left\{\frac{\sin \pi f_c(t_l - m\tau_k)}{\pi f_c(t_l - m\tau_k)}\right\}, \\ &= \frac{-m\pi f_c(t_l - m\tau_k)\cos \pi f_c(t_l - m\tau_k) + m \sin \pi f_c(t_l - m\tau_k)}{\pi f_c(t_l - m\tau_k)^2}. \end{aligned} \tag{C.39}$$

Accordingly, the second derivative of $h_l(m\tau_k)$ with respect to τ_k is given as follows

$$\begin{aligned} h_l^{''}(m\tau_k) &\triangleq \frac{d}{d\tau_k} h_l^{'}(m\tau_k), \\ &= \frac{\left(2 - (\pi f_c)^2 (t_l - m\tau_k)^2\right) m^2 \sin \pi f_c(t_l - m\tau_k) - 2m^2 \pi f_c (t_l - m\tau_k)\cos \pi f_c(t_l - m\tau_k)}{\pi f_c(t_l - m\tau_k)^4}. \end{aligned} \tag{C.40}$$

Appendix D

The Optimal Importance Sampling Function

D.1 Derivation of the importance sampling function for narrowband scenario

Given the observation vector $\boldsymbol{y}(t)$ described in (4.6) in Chapter 4, we now present the development of the optimal importance function $q_{optimal}(\cdot)$ described in (4.23). The observation vector in (4.6) and the state update equation in (4.4) are reproduced here for convenience

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}(t-1) + \sigma_w \boldsymbol{w}(t), \tag{D.41}$$

$$\boldsymbol{y}(t) = \boldsymbol{S}(\boldsymbol{\phi}(t))\boldsymbol{a}(t) + \sigma_v \boldsymbol{v}(t). \tag{D.42}$$

To approximate the optimal importance function, a first-order approximation is used that is a local linearization or Taylor expansion of the observation vector $\boldsymbol{y}(t)$ as follows

$$\boldsymbol{y}(t) \approx \boldsymbol{S}(\boldsymbol{\phi}(t-1))\boldsymbol{a}(t-1) + \nabla_\phi(t) \times \Big(\boldsymbol{\phi}(t) - \boldsymbol{\phi}(t-1)\Big) + \sigma_w \boldsymbol{w}(t), \tag{D.43}$$

where $\nabla_\phi(t)$ is the gradient vector defined as

$$\nabla_\phi(t) = \frac{\partial \boldsymbol{S}(\boldsymbol{\phi}(t))\boldsymbol{a}(t)}{\partial \boldsymbol{\phi}(t)}\bigg|_{\binom{\boldsymbol{\phi}(t)=\boldsymbol{\phi}(t-1)}{\boldsymbol{a}(t)=\boldsymbol{a}(t-1)}}. \tag{D.44}$$

Solving the above for $\boldsymbol{\phi}(t)$, from the assumptions on the model noise, yields the desired form of distribution function for $\boldsymbol{\phi}(t) given \boldsymbol{\phi}(t-1) and \boldsymbol{y}(t)$ that is linear and Gaussian as follows

$$q(\boldsymbol{\phi}^{(i)}(t)|\boldsymbol{\phi}^{(i-1)}(t-1), \boldsymbol{y}(t)) \sim \mathcal{N}(\boldsymbol{m}^{(i)}(t), \boldsymbol{\Sigma}^{(i)}(t)), \tag{D.45}$$

where, for each particle,

$$\boldsymbol{\Sigma}^{-1}(t) = \sigma_v^{-2}\boldsymbol{I}_{k(t)} + \nabla_\phi^H(t)\left(\sigma_w^{-2}\boldsymbol{I}_M\right)\nabla_\phi(t), \tag{D.46}$$

$$\boldsymbol{m}(t) = \boldsymbol{\Sigma}(t)\left(\sigma_v^{-2}\boldsymbol{I}_{k(t)}\boldsymbol{\phi}(t-1) + \nabla_\phi^H(t)\left(\sigma_w^{-2}\boldsymbol{I}_M\right)\left[\boldsymbol{y}(t) - \boldsymbol{S}(\boldsymbol{\phi}(t-1))\boldsymbol{a}(t-1) + \nabla_\phi(t)\boldsymbol{\phi}(t-1)\right]\right). \tag{D.47}$$

The distribution function in (D.45) is a first-order optimal importance function in which only the gradient vector is used.

D.1.1 Derivation of the Gradient Vectors

Let $\nabla_\phi(t) \in \mathcal{C}^{M\times K}$ be defined as a vector as follows

$$\nabla_\phi(t) = [\nabla_{\phi_0}(t), \nabla_{\phi_1}(t), ..., \nabla_{\phi_{k(t)-1}}(t)], \tag{D.48}$$

where $\nabla_{\phi_k}(t) \in \mathcal{C}^{M\times 1}$ is defined as

$$\begin{aligned}
\nabla_{\phi_k}(t) &= \frac{\partial \boldsymbol{S}(\boldsymbol{\phi}(t))\boldsymbol{a}(t)}{\partial \phi_k(t)}, \\
&= \frac{\partial \boldsymbol{s}(\phi_k(t))a_k(t)}{\partial \phi_k(t)}, \\
&= \boldsymbol{s}'(\phi_k(t))a_k(t), \\
&= a_k(t)\begin{bmatrix} 0 \\ -j\frac{2\pi d\cos\phi_k(t)}{\lambda}\exp\left(-j\frac{2\pi d\sin\phi_k(t)}{\lambda}\right) \\ \vdots \\ -j\frac{2(M-1)\pi d\cos\phi_k(t)}{\lambda}\exp\left(-j\frac{2(M-1)\pi d\sin\phi_k(t)}{\lambda}\right) \end{bmatrix}.
\end{aligned} \tag{D.49}$$

D.2 Derivation of the importance sampling function for wide-band scenario

Given the observation vector $\boldsymbol{z}(n)$ described in (3.47) in Chapter 3, we now present the development of the optimal importance function $q_{optimal}(\cdot)$ described in (6.27). The observation vector in (6.1) and the state update equation in (6.2) are reproduced here for convenience

$$\boldsymbol{\tau}(n) = \boldsymbol{\tau}(n-1) + \sigma_v \boldsymbol{v}(n), \tag{D.50}$$

$$\boldsymbol{z}(n) = \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}(n))\boldsymbol{a}(n) + \sigma_w \boldsymbol{w}(n). \tag{D.51}$$

We select the optimal importance function to be proportional to the terms the likelihood function and the prior function of $\boldsymbol{\tau}$. Let $\mathcal{L}(\boldsymbol{\tau}_n)$ be the logarithm of the optimal importance function, i.e.,

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\tau}_n\right) &\triangleq \mathcal{L}_z(\boldsymbol{\tau}_n) + \mathcal{L}_\tau(\boldsymbol{\tau}_n), \\ &= \log p\left(\boldsymbol{z}_n|\boldsymbol{\tau}_{1:n}, \boldsymbol{z}_{1:n-1}\right) + \log p\left(\boldsymbol{\tau}_n|\boldsymbol{\tau}_{1:n-1}, \boldsymbol{z}_{1:n}\right).\end{aligned} \tag{D.52}$$

We use a second-order Taylor expansion on $\mathcal{L}\left(\boldsymbol{\tau}_n\right)$ about the sensibly chosen point $\boldsymbol{\tau}_{n-1}$ to give

$$\mathcal{L}\left(\boldsymbol{\tau}_n\right) \approx \mathcal{L}\left(\boldsymbol{\tau}_{n-1}\right) + \nabla^T \mathcal{L}\left(\boldsymbol{\tau}_n\right) \times \left(\boldsymbol{\tau}_n - \boldsymbol{\tau}_{n-1}\right) + \frac{1}{2}\left(\boldsymbol{\tau}_n - \boldsymbol{\tau}_{n-1}\right)^T \nabla^2 \mathcal{L}\left(\boldsymbol{\tau}_n\right)\left(\boldsymbol{\tau}_n - \boldsymbol{\tau}_{n-1}\right), \tag{D.53}$$

where $\nabla\mathcal{L}(\boldsymbol{\tau}_n) \in \mathcal{R}^{k\times 1}$ and $\nabla^2\mathcal{L}(\boldsymbol{\tau}_n) \in \mathcal{R}^{k\times k}$ are the gradient vectors and the Hessian matrix of $\mathcal{L}(\boldsymbol{\tau}_n)$, respectively, defined as follows

$$\nabla\mathcal{L}(\boldsymbol{\tau}_n) = \frac{\partial}{\partial \boldsymbol{\tau}_n}\left(\mathcal{L}_z(\boldsymbol{\tau}_n) + \mathcal{L}_\tau(\boldsymbol{\tau}_n)\right)\Big|_{\substack{\boldsymbol{\tau}_n=\boldsymbol{\tau}_{n-1} \\ \boldsymbol{a}_n=\boldsymbol{a}_{n-1}}}, \tag{D.54}$$

$$\nabla^2\mathcal{L}(\boldsymbol{\tau}_n) = \frac{\partial^2}{\partial \boldsymbol{\tau}_n^2}\left(\mathcal{L}_z(\boldsymbol{\tau}_n) + \mathcal{L}_\tau(\boldsymbol{\tau}_n)\right)\Big|_{\substack{\boldsymbol{\tau}_n=\boldsymbol{\tau}_{n-1} \\ \boldsymbol{a}_n=\boldsymbol{a}_{n-1}}}. \tag{D.55}$$

D.2.1 Derivation of the Gradient Vectors

The derivation of the gradient vector in (D.54) will be described in detail in this section. The gradient vector $\nabla\mathcal{L}(\boldsymbol{\tau}_n)$ in (D.54) is composed of two components, given as follows

$$\nabla\mathcal{L}_z(\boldsymbol{\tau}_n) = \frac{\partial}{\partial\boldsymbol{\tau}_n}\log\left(p(\boldsymbol{z}_n|\boldsymbol{\alpha}_n^{(i)})\right), \tag{D.56}$$

$$\nabla\mathcal{L}_\tau(\boldsymbol{\tau}_n) = \frac{\partial}{\partial\boldsymbol{\tau}_n}\log\left(p(\boldsymbol{\tau}_n^{i}|\boldsymbol{\tau}_{n-1}^{(i)}, \boldsymbol{z}_n)\right). \tag{D.57}$$

We first present the derivation of $\nabla\mathcal{L}_z(\boldsymbol{\tau}_n)$ and then that of $\nabla\mathcal{L}_\tau(\boldsymbol{\tau}_n)$. Some details of the derivation can be found in Appendix C with the replacement of $\boldsymbol{\tau}$ by $\boldsymbol{\tau}_n$,

D.2.1.1 Derivation of $\nabla\mathcal{L}_z(\boldsymbol{\tau}_n)$

$$\nabla\mathcal{L}_z(\boldsymbol{\tau}_n) = [\nabla\mathcal{L}_z(\tau_{n,0}), \nabla\mathcal{L}_z(\tau_{n,1}), ..., \nabla\mathcal{L}_z(\tau_{n,k_n-1})]^T, \tag{D.58}$$

where

$$\begin{aligned}
\nabla\mathcal{L}_z(\tau_{n,k}) &= \frac{\partial}{\partial\tau_{n,k}}\log\left(p(\boldsymbol{z}_n|\boldsymbol{\alpha}_n)\right), \\
&= \frac{\partial}{\partial\tau_{n,k}}\left\{\frac{-1}{2\sigma_w^2}\left(\boldsymbol{z}_n - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_n)\boldsymbol{a}_n\right)^T\left(\boldsymbol{z}_n - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_n)\boldsymbol{a}_n\right) + \kappa_{\sigma_w^2}\right\}, \\
&= \frac{1}{\sigma_w^2}\sum_{l=0}^{L-1}\left(\boldsymbol{z}_n - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_n)\boldsymbol{a}_n\right)^T\tilde{\boldsymbol{H}}_l'(\tau_{n,k})\boldsymbol{s}_k(t-l), \\
&= \frac{1}{\sigma_w^2}\boldsymbol{\varepsilon}_n^T\tilde{\boldsymbol{H}}'(\tau_{n,k})\boldsymbol{s}_k(n),
\end{aligned} \tag{D.59}$$

where $\boldsymbol{\varepsilon}_n$ is given by

$$\begin{aligned}
\boldsymbol{\varepsilon}_n &= \boldsymbol{z}_n - \tilde{\boldsymbol{H}}_0(\boldsymbol{\tau}_n)\boldsymbol{a}_n, \\
&= \boldsymbol{y}_n - \sum_{l=0}^{L-1}\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}_n)\boldsymbol{a}_{n-l},
\end{aligned}$$

where $\kappa_{\sigma_w^2}$ is a function of the noise variance σ_w^2, and for $k = 0, ..., k_n - 1$

$$\tilde{\boldsymbol{H}}_l'(\tau_{n,k}) \triangleq \frac{\partial\tilde{\boldsymbol{H}}_l(\tau_{n,k})}{\partial\tau_{n,k}}, \tag{D.60}$$

$$\begin{aligned}\tilde{\boldsymbol{H}}'(\tau_{n,k}) &= \frac{\partial}{\partial \tau_{k,t}} \tilde{\boldsymbol{H}}(\tau_{n,k}), \\ &= \left[\tilde{\boldsymbol{H}}'_0(\tau_{n,k}), \tilde{\boldsymbol{H}}'_1(\tau_{n,k}), \ldots, \tilde{\boldsymbol{H}}'_{L-1}(\tau_{n,k})\right], \end{aligned} \tag{D.61}$$

and $\boldsymbol{s}_k(n)$ is the signal amplitude for the kth source, defined as

$$\boldsymbol{s}_k(n) \quad = \quad [s_k(n), s_k(n-1), \ldots, s_k(t-L+1)]^T. \tag{D.62}$$

D.2.1.2 Derivation of $\nabla \mathcal{L}_\tau(\boldsymbol{\tau}_n)$

$$\nabla \mathcal{L}_\tau(\boldsymbol{\tau}_n) = [\nabla \mathcal{L}_\tau(\tau_{t,0}), \nabla \mathcal{L}_\tau(\tau_{t,1}), ..., \nabla \mathcal{L}_\tau(\tau_{t,k_n-1})]^T, \tag{D.63}$$

where

$$\begin{aligned}\nabla \mathcal{L}_\tau(\tau_{n,k}) &= \frac{\partial}{\partial \tau_{n,k}} \log\left(p(\boldsymbol{\tau}_n|\boldsymbol{\tau}_{n-1})\right), \\ &= \frac{\partial}{\partial \tau_{n,k}} \left\{ \frac{-1}{2\sigma_v^2} \left(\boldsymbol{\tau}_n - \boldsymbol{\tau_{n-1}}\right)^T \left(\boldsymbol{\tau}_n - \boldsymbol{\tau_{n-1}}\right) + \kappa_{\sigma_v^2} \right\}, \\ &= \frac{-1}{\sigma_v^2}(\tau_{n,k} - \tau_{n-1,k}), \end{aligned} \tag{D.64}$$

where $\kappa_{\sigma_w^2}$ is a function of the noise variance σ_v^2.

As a result, the kth element of the gradient vector $\nabla \mathcal{L}(\boldsymbol{\tau}_n)$ can be expressed as

$$\begin{aligned}[\nabla \mathcal{L}(\boldsymbol{\tau}_n)]_k &= \nabla \mathcal{L}_z(\tau_{n,k}) + \nabla \mathcal{L}_\tau(\tau_{n,k}), \\ &= \frac{1}{\sigma_w^2} \boldsymbol{\varepsilon}_n^T \tilde{\boldsymbol{H}}'(\tau_{n,k}) \boldsymbol{s}_k(n) - \frac{1}{\sigma_v^2}(\tau_{n,k} - \tau_{n-1,k}). \end{aligned} \tag{D.65}$$

D.2.2 Derivation of the Hessian Matrices

The derivation of the Hessian matrices in (D.55) will be described in detail in this section. The gradient vector

$$\nabla^2 \mathcal{L}(\boldsymbol{\tau}_n) = \nabla^2 \mathcal{L}_z(\boldsymbol{\tau}_n) + \nabla^2 \mathcal{L}_\tau(\boldsymbol{\tau}_n), \tag{D.66}$$

where the k,pth elements of $\nabla^2 \mathcal{L}_z(\boldsymbol{\tau}_n)$ and $\nabla^2 \mathcal{L}_\tau(\boldsymbol{\tau}_n)$ are given by

$$\left[\nabla^2 \mathcal{L}_z(\boldsymbol{\tau}_n)\right]_{k,p} = \frac{\partial^2}{\partial \tau_{n,k} \partial \tau_{n,p}} \log\left(p(\boldsymbol{z}_n|\boldsymbol{\alpha}_n^{(i)})\right), \quad k,p = 0, ..., k_n - 1 \tag{D.67}$$

$$\left[\nabla^2 \mathcal{L}_\tau(\boldsymbol{\tau}_n)\right]_{k,p} = \frac{\partial^2}{\partial \tau_{n,k} \partial \tau_{n,p}} \log\left(p(\boldsymbol{\tau}_n^{i)}|\boldsymbol{\tau}_{n-1}^{(i)}, \boldsymbol{z}_n)\right), \quad k,p = 0, ..., k_n - 1. \tag{D.68}$$

D.2.2.1 Derivation of $\left[\nabla^2\mathcal{L}_z(\boldsymbol{\tau_n})\right]_{k,p}$

1. If $k \neq p$

$$\begin{aligned}\left[\nabla^2\mathcal{L}_z(\boldsymbol{\tau}_n)\right]_{k,p} &= \frac{\partial}{\partial\tau_{n,p}}\left\{\frac{\partial}{\partial\tau_{n,k}}\log\left(p(\boldsymbol{z}_n|\boldsymbol{\alpha}_n^{(i)})\right)\right\},\\ &= \frac{\partial}{\partial\tau_{n,p}}\nabla\mathcal{L}_z(\tau_{n,k}),\\ &= \frac{\partial}{\partial\tau_{n,p}}\left\{\frac{1}{\sigma_w^2}\boldsymbol{\varepsilon}_n^T\tilde{\boldsymbol{H}}'(\tau_{n,k})\boldsymbol{s}_k(n)\right\},\\ &= \frac{-1}{\sigma_w^2}\boldsymbol{s}_p^T(n)\left(\tilde{\boldsymbol{H}}'(\tau_{n,p})\right)^T\tilde{\boldsymbol{H}}'(\tau_{n,k})\boldsymbol{s}_k(n),\end{aligned} \tag{D.69}$$

2. If $k = p$

$$\begin{aligned}\left[\nabla^2\mathcal{L}_z(\boldsymbol{\tau}_n)\right]_{k,k} &= \frac{\partial^2}{\partial\tau_{n,k}^2}\log\left(p(\boldsymbol{z}_n|\boldsymbol{\alpha}_n^{(i)})\right),\\ &= \frac{\partial}{\partial\tau_{n,k}}\nabla\mathcal{L}_z(\tau_{n,k}),\\ &= \frac{\partial}{\partial\tau_{n,k}}\left\{\frac{1}{\sigma_w^2}\boldsymbol{\varepsilon}_n^T\tilde{\boldsymbol{H}}'(\tau_{n,k})\boldsymbol{s}_k(n)\right\},\\ &= \frac{1}{\sigma_w^2}\left\{\boldsymbol{\varepsilon}_n^T\tilde{\boldsymbol{H}}''(\tau_{n,k})\boldsymbol{s}_k(n) - \boldsymbol{s}_p^T(n)\left(\tilde{\boldsymbol{H}}'(\tau_{n,p})\right)^T\tilde{\boldsymbol{H}}'(\tau_{n,k})\boldsymbol{s}_k(n)\right\},\end{aligned} \tag{D.70}$$

where

$$\tilde{\boldsymbol{H}}''(\tau_{n,k}) = \frac{\partial}{\partial\tau_{n,k}}\tilde{\boldsymbol{H}}'(\tau_{n,k}). \tag{D.71}$$

D.2.2.2 Derivation of $\left[\nabla^2\mathcal{L}_\tau(\boldsymbol{\tau_n})\right]_{k,p}$

1. If $k \neq p$

$$\begin{aligned}\left[\nabla^2\mathcal{L}_\tau(\boldsymbol{\tau}_n)\right]_{k,p} &= \frac{\partial}{\partial\tau_{n,p}}\left\{\frac{\partial}{\partial\tau_{n,k}}\log\left(p(\boldsymbol{\tau}_n|\boldsymbol{\tau}_{n-1})\right)\right\},\\ &= \frac{\partial}{\partial\tau_{n,p}}\nabla\mathcal{L}_\tau(\tau_{n,k}),\\ &= \frac{\partial}{\partial\tau_{n,p}}\left\{\frac{1}{\sigma_v^2}(\tau_{n,k}-\tau_{n-1,k})\right\}\\ &= 0. \end{aligned} \tag{D.72}$$

2. If $k = p$

$$\begin{aligned}\left[\nabla^2\mathcal{L}_\tau(\boldsymbol{\tau}_n)\right]_{k,k} &= \frac{\partial^2}{\partial\tau_{n,k}^2}\log\left(p(\boldsymbol{\tau}_n|\boldsymbol{\tau}_{n-1})\right),\\ &= \frac{\partial}{\partial\tau_{n,k}}\nabla\mathcal{L}_\tau(\tau_{n,k}),\\ &= \frac{\partial}{\partial\tau_{n,k}}\left\{\frac{-1}{\sigma_v^2}(\tau_{n,k}-\tau_{n-1,k})\right\},\\ &= \frac{-1}{\sigma_v^2}. \end{aligned} \tag{D.73}$$

Therefore, the k,pth element of the Hessian matrix $\nabla^2\mathcal{L}(\boldsymbol{\tau}_n)$ in (D.55) can be expressed as follows

$$\begin{aligned}\left[\nabla^2\mathcal{L}(\boldsymbol{\tau}_n)\right]_{k,p} &= \left[\nabla^2\mathcal{L}_z(\boldsymbol{\tau}_n)\right]_{k,p} + \left[\nabla^2\mathcal{L}_\tau(\boldsymbol{\tau}_n)\right]_{k,p},\\ &= \begin{cases} -\sigma_w^{-2}\boldsymbol{s}_k^T(n)\left(\tilde{\boldsymbol{H}}'(\tau_{n,k})\right)^T\tilde{\boldsymbol{H}}'(\tau_{n,k})\boldsymbol{s}_k(n), & \text{if } k=p,\\ \sigma_w^{-2}\left\{\boldsymbol{\varepsilon}_n^T\tilde{\boldsymbol{H}}''(\tau_{n,k})\boldsymbol{s}_k(n) - \boldsymbol{s}_p^T(n)\left(\tilde{\boldsymbol{H}}'(\tau_{n,p})\right)^T\tilde{\boldsymbol{H}}'(\tau_{n,k})\boldsymbol{s}_k(n)\right\} - \sigma_v^{-2}, & \text{if } k\neq p. \end{cases} \end{aligned} \tag{D.74}$$

Appendix E

Derivation of The PCRB

Let $\boldsymbol{y}$ be an observation vector, $\boldsymbol{\theta}$ be an k_0-dimensional parameter vector, $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ be a function of $\boldsymbol{y}$, which represents an estimate of $\boldsymbol{\theta}$, and $p(\boldsymbol{y}, \boldsymbol{\theta})$ be the joint probability density of $(\boldsymbol{y}, \boldsymbol{\theta})$. The PCRB on the estimation error on $\boldsymbol{\theta}$ has the form

$$P = E\left\{[\hat{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{\theta}][\hat{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{\theta}]^T\right\} \geq \boldsymbol{\mathcal{J}}^{-1}, \tag{E.75}$$

where $\boldsymbol{\mathcal{J}}$ is the $k_0 \times k_0$ Fisher Information matrix with elements

$$[\boldsymbol{\mathcal{J}}]_{i,j} = E\bigg[-\frac{\partial^2 \log p(\boldsymbol{y}, \boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\bigg], \quad i, j = 0, ..., k_0 - 1, \tag{E.76}$$

provided that the derivatives and expectations in (E.75) and (E.76) exist. Let $\nabla_{\boldsymbol{\theta}}$ be the operator of the first-order partial derivative as follows

$$\nabla_{\boldsymbol{\theta}} = \left[\frac{\partial}{\partial\theta_0}, ..., \frac{\partial}{\partial\theta_{k_0-1}}\right]^T, \tag{E.77}$$

such that we can express $\boldsymbol{\mathcal{J}}$ as

$$\begin{aligned}
\boldsymbol{\mathcal{J}} &= -E\left[\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T \log p(\boldsymbol{y}, \boldsymbol{\theta})\right], & \text{(E.78)} \\
&= -E\left[\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T \Big\{ \log p(\boldsymbol{y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \Big\}\right], & \text{(E.79)} \\
&= \boldsymbol{\mathcal{J}}_D + \boldsymbol{\mathcal{J}}_P, & \text{(E.80)}
\end{aligned}$$

where $\boldsymbol{\mathcal{J}}_D$, also recognized as the standard Fisher Information matrix (Li *et al.*, 2001), represents the information obtained from the data, defined as

$$\boldsymbol{\mathcal{J}}_D = -E\left[\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T \log p(\boldsymbol{y}|\boldsymbol{\theta})\right], \tag{E.81}$$

and $\boldsymbol{\mathcal{J}}_P$ is the information obtained from the *a prior* information, defined as

$$\boldsymbol{\mathcal{J}}_P = -E\left[\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T \log p(\boldsymbol{\theta})\right]. \tag{E.82}$$

E.1 Introduction to information submatrix

Assume that the parameter vector can be partitioned as follows

$$\boldsymbol{\theta} = \left[\boldsymbol{\theta}_a^T, \boldsymbol{\theta}_b^T\right]^T, \quad \boldsymbol{\theta}_a \in \mathcal{R}^{k_a \times 1}, \boldsymbol{\theta}_b \in \mathcal{R}^{k_b \times 1} \tag{E.83}$$

where $k_a + k_b = k_0$, and the information matrix $\boldsymbol{\mathcal{J}}$ can be partitioned into blocks as follows

$$\boldsymbol{\mathcal{J}} = \begin{bmatrix} \boldsymbol{\mathcal{J}}_{aa} & \boldsymbol{\mathcal{J}}_{ab} \\ \boldsymbol{\mathcal{J}}_{ba} & \boldsymbol{\mathcal{J}}_{bb} \end{bmatrix}, \tag{E.84}$$

where $\boldsymbol{\mathcal{J}}_{ba} = \boldsymbol{\mathcal{J}}_{ab}^T$. The inverses of the submatrices in $\boldsymbol{\mathcal{J}}$ in (E.84) are the corresponding covariance matrices with other parameters fixed. It can be shown that (Tichavsky *et al.*, 1998) that the covariance of estimation of $\boldsymbol{\theta}_b$, P_b, is lower bounded by the righn-lower block ofof $\boldsymbol{\mathcal{J}}^{-1}$ as follows

$$\begin{aligned} P_b &= E\left\{ \left[\hat{\boldsymbol{\theta}}_b(\boldsymbol{y}) - \boldsymbol{\theta}_b\right] \left[\hat{\boldsymbol{\theta}}_b(\boldsymbol{y}) - \boldsymbol{\theta}_b\right]^T \right\}, && \text{(E.85)} \\ &\geq [\boldsymbol{\mathcal{J}}_{bb} - \boldsymbol{\mathcal{J}}_{ba}\boldsymbol{\mathcal{J}}_{aa}^{-1}\boldsymbol{\mathcal{J}}_{ab}]^{-1}, && \text{(E.86)} \\ &= \boldsymbol{\mathcal{J}}^{-1}(\boldsymbol{\theta}_b), && \text{(E.87)} \end{aligned}$$

provided that $\boldsymbol{\mathcal{J}}_{aa}^{-1}$ exists. The matrix $\boldsymbol{\mathcal{J}}(\boldsymbol{\theta}_b) \in \mathcal{R}^{k_b \times k_b}$, known as the *information submatrix* for parameter $\boldsymbol{\theta}_b$, is given by

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\theta}_b) = \boldsymbol{\mathcal{J}}_{bb} - \boldsymbol{\mathcal{J}}_{ba}\boldsymbol{\mathcal{J}}_{aa}^{-1}\boldsymbol{\mathcal{J}}_{ab}. \tag{E.88}$$

E.2 Sequential update for the information submatrix

Given that the state-space model as follows

$$\boldsymbol{\tau}(n) = \boldsymbol{\tau}(n-1) + \sigma_v \boldsymbol{v}(n), \tag{E.89}$$

$$\boldsymbol{y}(n) = \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}(n))\boldsymbol{a}(n-l) + \sigma_w \boldsymbol{w}(n), \tag{E.90}$$

and that both $\boldsymbol{v}(n)$ and $\boldsymbol{w}(n)$ are *iid* Gaussian random variables with zero mean and unit variance, the total joint probability density function $p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n)$, where $\boldsymbol{\mathcal{Y}}_n = \boldsymbol{y}_{1:n}$ and $\boldsymbol{\mathcal{T}}_n = \boldsymbol{\tau}_{1:n}$, can be given as follows

$$p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n) = p(\boldsymbol{\tau}_0) \prod_{j=1}^{n} p(\boldsymbol{y}_j|\boldsymbol{\tau}_j) \prod_{j=1}^{n} p(\boldsymbol{\tau}_j|\boldsymbol{\tau}_{j-1}), \tag{E.91}$$

where $p(\boldsymbol{\tau}_0)$ is assumed known. According to (E.78), we can derive an $nk_0 \times nk_0$ information matrix $\boldsymbol{\mathcal{J}}(\boldsymbol{\mathcal{T}}_n)$ from $p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n)$. However, instead of computing the information matrix $\boldsymbol{\mathcal{J}}(\boldsymbol{\mathcal{T}}_n)$, we are more interested in computing the $k_0 \times k_0$ instantaneous information submatrix as in (E.88) for the parameter $\boldsymbol{\tau}_n$.

Let $\boldsymbol{\mathcal{T}}_n$ be partitioned as $\left[\boldsymbol{\mathcal{T}}_{n-1}^T, \boldsymbol{\tau}_n^T\right]^T$. Following (E.83)-(E.84), we can express $\boldsymbol{\mathcal{J}}(\boldsymbol{\mathcal{T}}_n) \in \mathcal{R}^{nk_0 \times nk_0}$ as

$$\begin{aligned}
\boldsymbol{\mathcal{J}}(\boldsymbol{\mathcal{T}}_n) &= \begin{bmatrix} \boldsymbol{A}_n & \boldsymbol{B}_n \\ \boldsymbol{B}_n^T & \boldsymbol{C}_n \end{bmatrix}, \\
&= \begin{bmatrix} -E\left[\nabla_{\boldsymbol{\mathcal{T}}_{n-1}} \nabla_{\boldsymbol{\mathcal{T}}_{n-1}}^T \log p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n)\right] & -E\left[\nabla_{\boldsymbol{\tau}_n} \nabla_{\boldsymbol{\mathcal{T}}_{n-1}}^T \log p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n)\right] \\ -E\left[\nabla_{\boldsymbol{\mathcal{T}}_{n-1}} \nabla_{\boldsymbol{\tau}_n}^T \log p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n)\right] & -E\left[\nabla_{\boldsymbol{\tau}_n} \nabla_{\boldsymbol{\tau}_n}^T \log p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n)\right] \end{bmatrix},
\end{aligned} \tag{E.92}$$

provided that the derivatives and the expectations exist. As a result, according to (E.88), we obtain an expression of $\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_n) \in \mathcal{R}^{k_0 \times k_0}$ as follows

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_n) = \boldsymbol{C}_n - \boldsymbol{B}_n^T \boldsymbol{A}_n^{-1} \boldsymbol{B}_n. \tag{E.93}$$

In order to get a recursive update equation of $\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_{n+1})$, given $\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_n)$ and $\boldsymbol{y}_{n+1}$, we need

to first consider the joint probability function $p(\boldsymbol{\mathcal{Y}}_{n+1}, \boldsymbol{\mathcal{T}}_{n+1})$ as follows

$$\begin{aligned} p(\boldsymbol{\mathcal{Y}}_{n+1}, \boldsymbol{\mathcal{T}}_{n+1}) &= p(\boldsymbol{y}_{n+1}|\boldsymbol{\mathcal{T}}_{n+1}, \boldsymbol{\mathcal{Y}}_n) p(\boldsymbol{\tau}_{n+1}|\boldsymbol{\mathcal{T}}_n, \boldsymbol{\mathcal{Y}}_n) p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n), \\ &= p(\boldsymbol{y}_{n+1}|\boldsymbol{\tau}_{n+1}) p(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n) p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n), \end{aligned} \tag{E.94}$$

where we use the fact that $\boldsymbol{\mathcal{T}}_n$ is independent of $\boldsymbol{y}_{n+1}$ and that innovations of $\boldsymbol{\tau}_n$ are independent.

Accordingly, the information matrix $\boldsymbol{\mathcal{J}}(\mathcal{T}_{n+1})$ with $\boldsymbol{\mathcal{T}}_{n+1}$ partitioned as $\left[\boldsymbol{\mathcal{T}}_n^T, \boldsymbol{\tau}_{n+1}^T\right]^T$ can be shown to be

$$\begin{aligned} \boldsymbol{\mathcal{J}}(\mathcal{T}_{n+1}) &= -E\left[\nabla_{\boldsymbol{\mathcal{T}}_{n+1}} \nabla^T_{\boldsymbol{\mathcal{T}}_{n+1}} \log p(\boldsymbol{\mathcal{Y}}_{n+1}, \boldsymbol{\mathcal{T}}_{n+1})\right], \\ &= -E\left[\nabla_{\boldsymbol{\mathcal{T}}_{n+1}} \nabla^T_{\boldsymbol{\mathcal{T}}_{n+1}} \left\{ \log p(\boldsymbol{y}_{n+1}|\boldsymbol{\tau}_{n+1}) + \log p(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n) + \log p(\boldsymbol{\mathcal{Y}}_n, \boldsymbol{\mathcal{T}}_n) \right\}\right], \\ &= \begin{bmatrix} \boldsymbol{A}_{n+1} & \boldsymbol{B}_{n+1} \\ \boldsymbol{B}_{n+1}^T & \boldsymbol{C}_{n+1} \end{bmatrix}, \end{aligned} \tag{E.95}$$

where the terms $\boldsymbol{A}_{n+1}$, $\boldsymbol{B}_{n+1}$, and $\boldsymbol{C}_{n+1}$ are given by

$$\boldsymbol{A}_{n+1} = \begin{bmatrix} \boldsymbol{A}_n & \boldsymbol{B}_n \\ \boldsymbol{B}_n^T & \boldsymbol{C}_n + \boldsymbol{D}_n^{11} \end{bmatrix}, \tag{E.96}$$

$$\boldsymbol{B}_{n+1} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{D}_n^{12} \end{bmatrix}, \tag{E.97}$$

$$\boldsymbol{C}_{n+1} = \boldsymbol{D}_n^{22}, \tag{E.98}$$

and the terms $\boldsymbol{D}_n^{11} \in \mathcal{R}^{k_0 \times k_0}$, $\boldsymbol{D}_n^{12} \in \mathcal{R}^{k_0 \times k_0}$, $\boldsymbol{D}_n^{21} \in \mathcal{R}^{k_0 \times k_0}$, and $\boldsymbol{D}_n^{22} \in \mathcal{R}^{k_0 \times k_0}$ are defined as follows

$$\boldsymbol{D}_n^{11} = E\left[-\nabla_{\boldsymbol{\tau}_n} \nabla^T_{\boldsymbol{\tau}_n} \log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right)\right], \tag{E.99}$$

$$\boldsymbol{D}_n^{12} = E\left[-\nabla_{\boldsymbol{\tau}_{n+1}} \nabla^T_{\boldsymbol{\tau}_n} \log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right)\right], \tag{E.100}$$

$$\boldsymbol{D}_n^{21} = \left[\boldsymbol{D}_n^{12}\right]^T, \tag{E.101}$$

$$\begin{aligned} \boldsymbol{D}_n^{22} =& E\left[-\nabla_{\boldsymbol{\tau}_{n+1}} \nabla^T_{\boldsymbol{\tau}_{n+1}} \log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right)\right] + \\ & E\left[-\nabla_{\boldsymbol{\tau}_{n+1}} \nabla^T_{\boldsymbol{\tau}_{n+1}} \log p\left(\boldsymbol{y}_{n+1}|\boldsymbol{\tau}_{n+1}\right)\right]. \end{aligned} \tag{E.102}$$

Thus the information submatrix $\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_{n+1})$ can be given by the inverse of the righn-lower submatrix of $\boldsymbol{\mathcal{J}}^{-1}(\boldsymbol{\mathcal{T}}_{n+1})$ as in (E.88) by

$$\begin{aligned}\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_{n+1}) &= \boldsymbol{C}_{n+1} - \boldsymbol{B}_{n+1}^T \boldsymbol{A}_{n+1}^{-1} \boldsymbol{B}_{n+1}, \\ &= \boldsymbol{D}_n^{22} - \begin{bmatrix} \mathbf{0}, \boldsymbol{D}_n^{21} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_n & \boldsymbol{B}_n \\ \boldsymbol{B}_n^T & \boldsymbol{C}_n + \boldsymbol{D}_n^{11} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{D}_n^{12} \end{bmatrix}, \\ &= \boldsymbol{D}_n^{22} - \boldsymbol{D}_n^{21} \Big[\boldsymbol{D}_n^{11} + \boldsymbol{C}_n - \boldsymbol{B}_n^T \boldsymbol{A}_n^{-1} \boldsymbol{B}_n \Big]^{-1} \boldsymbol{D}_n^{12}, \\ &= \boldsymbol{D}_n^{22} - \boldsymbol{D}_n^{21} \Big[\boldsymbol{D}_n^{11} + \boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_n) \Big]^{-1} \boldsymbol{D}_n^{12},\end{aligned} \tag{E.103}$$

which is the desired recursive update equation of the information submatrix for $\boldsymbol{\tau}_{n+1}$. The initial information submatrix $\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_0)$ can be computed from the *a priori* probability function $p(\boldsymbol{\tau}_0)$ as follows

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\tau}_0) = -E\Big[\nabla_{\boldsymbol{\tau}_0} \nabla_{\boldsymbol{\tau}_0}^T \log p(\boldsymbol{\tau}_0) \Big]. \tag{E.104}$$

E.3 Derivation of $\boldsymbol{D}_n^{11}$, $\boldsymbol{D}_n^{12}$, $\boldsymbol{D}_n^{21}$, and $\boldsymbol{D}_n^{22}$

Given the state-space model in (E.90) and that both $\boldsymbol{v}(n)$ and $\boldsymbol{w}(n)$ are *iid* Gaussian random variables with zero mean and unit variance, the functions $\log p(\boldsymbol{y}_{n+1}|\boldsymbol{\tau}_{n+1})$ and $\log p(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n)$ are given as follows

$$\begin{aligned}&\log p\left(\boldsymbol{y}_{n+1}|\boldsymbol{\tau}_{n+1}\right) = \kappa_{\sigma_w} - \\ &\frac{1}{2\sigma_w^2} \left(\boldsymbol{y}_{n+1} - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}(\boldsymbol{\tau}_{n+1}) \boldsymbol{a}_{n-l+1} \right)^T \left(\boldsymbol{y}_{n+1} - \sum_{l=0}^{L-1} \tilde{\boldsymbol{H}}(\boldsymbol{\tau}_{n+1}) \boldsymbol{a}_{n-l+1} \right),\end{aligned} \tag{E.105}$$

$$\log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right) = \kappa_{\sigma_v} - \frac{1}{2\sigma_v^2} \left(\boldsymbol{\tau}_{n+1} - \boldsymbol{\tau}_n\right)^T \left(\boldsymbol{\tau}_{n+1} - \boldsymbol{\tau}_n\right), \tag{E.106}$$

where κ_{σ_w} and κ_{σ_v} are a function of σ_w^2 and σ_v^2, respectively. Next we present the derivations of $\boldsymbol{D}_n^{11}$, $\boldsymbol{D}_n^{12}$, $\boldsymbol{D}_n^{21}$, and $\boldsymbol{D}_n^{22}$ in sequel, where some results are taken from Appendix D.

E.3.1 Derivation of $\boldsymbol{D}_n^{11}$

The term $\boldsymbol{D}_n^{11}$ is defined as

$$E\left\{-\nabla_{\boldsymbol{\tau}_n}\nabla_{\boldsymbol{\tau}_n}^T \log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right)\right\}, \tag{E.107}$$

whose i,jth element of $\boldsymbol{D}_n^{11}$ is defined as follows

$$\begin{aligned}
[\boldsymbol{D}_n^{11}]_{i,j} &= \frac{\partial}{\partial\tau_{t,j}}\left\{\frac{-\partial}{\partial\tau_{t,i}}\log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right)\right\}, \quad i,j=0,...,k_0-1 \\
&= \frac{-1}{\sigma_v^2}\frac{\partial}{\partial\tau_{t,j}}\left\{\left(\tau_{n+1,i}-\tau_{t,i}\right)\right\}, \\
&= \begin{cases} 0, & \text{if } i\neq j \\ \frac{1}{\sigma_v^2}, & \text{if } i=j \end{cases}.
\end{aligned} \tag{E.108}$$

In other words, the matrix $\boldsymbol{D}_n^{11}$ is a diagonal matrix defined as

$$\boldsymbol{D}_n^{11} = \frac{1}{\sigma_v^2}\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix}. \tag{E.109}$$

E.3.2 Derivation of $\boldsymbol{D}_n^{12}$

The i,jth element of $\boldsymbol{D}_n^{12}$ is defined as follows

$$\begin{aligned}
[\boldsymbol{D}_n^{12}]_{i,j} &= \frac{\partial}{\partial\tau_{n+1,j}}\left\{\frac{-\partial}{\partial\tau_{t,i}}\log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right)\right\}, \quad i,j=0,...,k_0-1 \\
&= \frac{-1}{\sigma_v^2}\frac{\partial}{\partial\tau_{n+1,j}}\left\{\left(\tau_{n+1,i}-\tau_{t,i}\right)\right\}, \\
&= \begin{cases} 0, & \text{if } i\neq j \\ \frac{-1}{\sigma_v^2}, & \text{if } i=j \end{cases}.
\end{aligned} \tag{E.110}$$

In other words, the matrix $\boldsymbol{D}_n^{12}$ and $\boldsymbol{D}_n^{21} = \left[\boldsymbol{D}_n^{12}\right]^T$ are a diagonal matrix defined as

$$\boldsymbol{D}_n^{12} = \boldsymbol{D}_n^{21} = \frac{-1}{\sigma_v^2}\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix}. \tag{E.111}$$

E.3.3 Derivation of $\boldsymbol{D}_n^{22}$

The i,jth element of $\boldsymbol{D}_n^{22}$ is defined as follows

$$\begin{aligned}
[\boldsymbol{D}_n^{22}]_{i,j} &= \frac{\partial}{\partial \tau_{n+1,j}}\left\{\frac{-\partial}{\partial \tau_{n+1,i}}\log p\left(\boldsymbol{\tau}_{n+1}|\boldsymbol{\tau}_n\right) + \frac{-\partial}{\partial \tau_{n+1,i}}\log p\left(\boldsymbol{y}_{n+1}|\boldsymbol{\tau}_{n+1}\right)\right\}, \quad i,j = 0,...,k_0-1, \\
&= \frac{\partial}{\partial \tau_{n+1,j}}\left\{\frac{1}{\sigma_v^2}(\tau_{n+1,i} - \tau_{t,i}) - \frac{1}{\sigma_w^2}\boldsymbol{\epsilon}_{n+1}^T\tilde{\boldsymbol{H}}'(\tau_{n+1,i})\boldsymbol{s}_i(n+1)\right\}, \\
&= \begin{cases} \frac{1}{\sigma_w^2}f_{i,j}(n+1), & \text{if } i \neq j \\ \frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2}\left[\boldsymbol{\epsilon}_{n+1}^T\tilde{\boldsymbol{H}}''(\tau_{n+1,i})\boldsymbol{s}_i(n+1) + f_{i,i}(n+1)\right], & \text{if } i = j \end{cases},
\end{aligned} \tag{E.112}$$

where

$$\boldsymbol{\epsilon}_{n+1} = \boldsymbol{y}_{n+1} - \sum_{l=0}^{L-1}\tilde{\boldsymbol{H}}_l(\boldsymbol{\tau}_{n+1})\boldsymbol{a}_{n-l+1}, \tag{E.113}$$

$$\tilde{\boldsymbol{H}}'(\tau_{n+1,i}) = \frac{\partial}{\partial \tau_{n+1,i}}\tilde{\boldsymbol{H}}(\tau_{n+1,i}), \tag{E.114}$$

$$\tilde{\boldsymbol{H}}''(\tau_{n+1,i}) = \frac{\partial}{\partial \tau_{n+1,i}}\tilde{\boldsymbol{H}}'(\tau_{n+1,i}). \tag{E.115}$$

$$f_{i,j}(n+1) = -\boldsymbol{s}_j^T(n+1)\left(\tilde{\boldsymbol{H}}'(\tau_{n+1,j})\right)^T\tilde{\boldsymbol{H}}'(\tau_{n+1,i})\boldsymbol{s}_i(n+1) \tag{E.116}$$

Bibliography

Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716–723.

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ.

Andrieu, C. (1997). *Méthodes MCMC pour l'analyse Bayesienne de modèles de régression paramétrique non-linéaire. Application à l'analyse de raies et à la déconvolution impulsionnelle*. Ph.D. thesis, Université de Cergy-Pontoise, France.

Andrieu, C. (1998). On sequential simulation-based methods for Bayesian filtering. Technical report. Download from http://www.cs.ubc.ca/ nando/smc/smcpapers.html.

Andrieu, C. and Doucet, A. (1999). Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing*, **47**(10), 2667–2676.

Andrieu, C., Doucet, A., Fitzgerald, W., and Godsill, S. (1998). An introduction to the theory and applications of simulation based computational methods in Bayesian signal processing. In *Tutorial from Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA.

Andrieu, C., Freitas, N. D., and Doucet, A. (1999). Sequential MCMC for Bayesian model selection. In *Proceedings of the Internationl Workshop on Higher Order Statistics*, Ceasarea, Israel.

Andrieu, C., Djuric, P., and Doucet, A. (2001). Model selection by MCMC computation. *Signal Processing*, **81**(1), 19–37.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, **50**(2), 174–188.

Aspach, D. and Sorenson, H. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transaction Automatic Control*, **17**(4), 439–448.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with Appliations to Tracking and Navigation*. John Wiley & Sons, New York.

Barabell, A. (1983). Improving the resolution performance of eigenstructure-based direction-finding algorithm. In *Proceedings Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 336–339, Boston, MA.

Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Ph.D. thesis, Linkoping University, Linkoping, Sweden.

Bernardo, J. and Smith, A. (1994). *Bayesian Theory*. John Wiley & Sons, New York.

Boehme, J. (1986). Separated estimation of wave parameters and spectral parameters. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 2819–2822, Tokyo, Japan.

Boehme, J. (1989). Location and spectrum estimation by approximate maximum likelihood. In *Proceedings SPIE*, volume 1, pages 326–337, Orlando, Florida.

Box, G. and Tiao, G. (1973). *Bayesian Inference in Statistical Analysis*. Addison Wesley, Reading, M.A.

Buckley, K. and Griffiths, L. (1988). Broadband signal subspace spatial spectrum (BASS-ALE) estimation. *IEEE Transaction on Signal Processing*, **36**, 953–964.

Bucy, R. and Senne, K. (1971). Digital sysnthesis of nonlinear filters. *Automatica*, **7**, 287–298.

Chen, W., Reilly, J., and Wong, K. (1996). Detection of the number of signals in noise with banded covariance matrices. *IEE Proceedings on Radar, Sonar and Navigation*, **143**, 289–294.

Chen, W.-G. (1991). *Detection of the Number of Signals in Array Signal Processing.* Ph.D. thesis, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada.

Djuric, P. M. (1996). A model selection rule for sinusoids in white gaussian noise. *IEEE Transactions on Signal Processing*, **44**(7), 1744–1751.

Doron, M. A. and Weiss, A. J. (1992). On focusing matrices for wide-band array processing. *IEEE Transactions on Signal Processing*, **40**(6), 1295–1302.

Doron, M. A., Weiss, A. J., and Messer, H. (1993). Maximum-likelihood direction finding of wide-band sources. *IEEE Transactions on Signal Processing*, **41**(1), 411–414.

Doucet, A. (1998). On sequential simulation-based methods for Bayesian filtering. Technical Report TR.310, University of Cambridge, Department of Engineering, Signal Processing Group, England.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, **10**, 197–208.

Doucet, A., de Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo in Practice.* Springer-Verlag, New York.

Fuchs, J. (1992). Estimation of the number of signals in the presence of unknown correlated sensor noise. *IEEE Transactions on Signal Processing*, **40**, 1053–1061.

Gamerman, D. (1997). *Markov Chain Monte Carlo. Stochastic Simulation for Bayesian Inference.* Chapman and Hall, Suffolk, England.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(6), 721–732.

Gershman, A. (1999). Robust adaptive beamforming in sensor arrays. In *Int. Journal of Electronics and Communications*, volume 53, pages 305–314.

Gilks, W. and Berzuini, C. (1998). Following a moving target - Monte Carlo inference for dynamic Bayesian models. Technical report, University of Pavia, Department of Computer Sciences and Systems, Italy.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1998). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, New York.

Godsill, S. (1998). On the relationship between MCMC model uncertainty methods. Technical Report TR.305, Cambridge University, Engineering Department, England.

Godsill, S. (2001). On the relationship between MCMC model uncertainty methods. *Journal of Computational and Graphical Statistics*, **10**(2).

Golub, G. H. and Loan, C. F. V. (1993). *MATRIX Computations*. The Johns Hopkins University Press, Baltimore, MA, 2nd edition.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, **140**(2), 107–113.

Green, P. (1995). Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**(4), 711–732.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1), 97–109.

Haykin, S. (1985). *Array Signal Processing*. Prentice Hall, Englewood Cliffs, NJ.

Haykin, S. (2000). *Adaptive Filter Theory*. Prentice Hall, Englewood Cliffs, NJ, 4th edition.

Hung, H. and Kaveh, M. (1988). Focusing matrices for coherent signal-subspace processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **36**, 1272–1281.

Jauffret, C. and Bar-Shalom, Y. (1990). Track formation with bearing and frequency measurements in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, **26**(4), 999–1010.

Jeffreys, H. (1961). *Theory of Probability*. Oxford University Press, London, U.K., 3rd edition.

Johnson, D. (1982). The application of spectral estimation to bearing estimation problem. *Proceedings of the IEEE*, **70**(1), 1018–1028.

Johnson, D. and Dudgeon, D. (1993). *Array Signal Processing: Concepts and Techniques*. PTR Prentice Hall, Englewood Cliff, NJ.

Katkovnik, V. and Gershman, A. B. (2000). A local polynomial approximation based beamforming for source localization and tracking in nonstationary environments. **7**(1), 3–5.

Katkovnik, V. and Gershman, A. B. (2002). Performance study of the local polynomial approximation based beamforming in the presence of moving sources. *IEEE Transactions on Signal Processing*, **50**(8), 1151–1157.

Kurien, T. and Washburn, R. (1985). Multiobject tracking using passive sensors. In *Proceedings of the American Control Conference*, Boston, MA.

Larocque, J.-R., Reilly, J. P., and Ng, W. (2002). Particle filter for tracking an unknown number of sources. *IEEE Transactions on Signal Processing*, **50**(12), 2926–2937.

Lee, Y. W. (1960). *Statistical Theory of Communication*. John Wiley & Sons, New York.

Li, X.-R., Bar-Shalom, Y., and Kirubarajan, T. (June, 2001). *Estimation, Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, New York.

Liu, J. S. and Chen, R. (1999). Sequential Monte Carlo Methods for dynamic systems. *Journal of the American Statistical Association*, **93**, 1032–1044.

MacEachern, S., Clyde, M., and Liu, J. (1999). Sequential importance sampling for non-parametric Bayes models: The next generation. *The Canadian Journal of Statistics*, **27**, 251–267.

Nagesha, V. and Kay, S. (1996). Maximum likelihood estimation for array processing in coloured noise. *IEEE Transactions on Signal Processing*, **44**(2), 169–180.

Ng, W., Larocque, J.-R., and Reilly, J. P. (2001a). On the implementation of particle filters for DOA tracking. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2821–2824, Salt Lake City, UT.

Ng, W., Larocque, J.-R., and Reilly, J. P. (2001b). Sequential MCMC for spatial signal separation and restoration from an array of sensors. In *The annual Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, pages 89–108, Johns Hopkins University, Baltimore, MA.

Ng, W., Reilly, J. P., Kirubarajan, T., and Larocque, J.-R. (2002a). Wideband array signal processing using MCMC methods. Submitted to *IEEE Transactions on Signal Processing*, Jan. 2003, also available at http://www.ece.mcmaster.ca/∼reilly.

Ng, W., Reilly, J. P., and Kirubarajan, T. (2002b). Wideband array signal processing using MCMC methods. In *Sensor Array and Multichannel Signal Processing Workshop Proceedings*, pages 350–354, Washington, D.C.

Ng, W., Reilly, J. P., Kirubarajan, T., and Larocque, J.-R. (2003a). Advanced beamforming for narrowband signals using sequential MCMC methods. Submitted to *IEEE Transactions Aerospace and Electronic Systems*, July 2003, also available at http://www.ece.mcmaster.ca/∼reilly.

Ng, W., Reilly, J. P., Kirubarajan, T., and Larocque, J.-R. (2003b). Application of particle

filters for tracking moving receivers in wireless communication systems. In *Proceedings of the Signal Processing Advances In Wireless Communications*, Rome, Italy.

Ng, W., Reilly, J., and Kirubarajan, T. (2003c). A bayesian approach to tracking wideband targets using sensor arrays and particle filters. In *Proceedings of the Statistical Signal Processing*, St. Louis, MO.

Ng, W., Reilly, J. P., and Kirubarajan, T. (2003d). Wideband Array Signal Processing Using MCMC Methods. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 189–192, Hong Kong, China.

Ng, W., Reilly, J. P., and Kirubarajan, T. (2003e). Wideband array signal processing using sequential MCMC methods. Submitted to IEEE Transactions on Signal Processing, Sept. 2003, also available at http://www.ece.mcmaster.ca/∼reilly.

Orton, M. and Fitzgerald, W. (2002). A Bayesian apporach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, **50**(2), 216–223.

Pattipati, K., Deb, S., Bar-Shalom, Y., and Washburn, R. (1992). A new relaxation algorithm and passive sensor data association. *IEEE Transactions on Automatic Control*, **37**(2), 198–213.

Pillai, S. U. (1989). *Array Signal Processing*. Springer-Verlag, New York.

Pisarenko, V. F. (1973). The retrieval of harmonics from a covariance function. *Geophysical Journal Royal of Astronomical Society*, **33**, 347–366.

Rao, B. D. and Hari, K. (1989). Performance analysis of root-music. *IEEE Transactions on Signal Processing*, **37**(12), 1939–1949.

Rao, C., Sastry, C., and Zhou, B. (1994). Tracking the direction of arrival of multiple moving targets. *IEEE Transactions on Signal Processing*, **42**(5), 1133–1144.

Reed, J., Krizman, K., Woerner, B., and Rappaport, T. (1988). An overview of the challenges and progress in meeting the E-911 requirement for location service. **36**, 30–37.

Reilly, J. and Haykin, S. (1982). Maximum likelihood receiver for low-angle tracking radar, parts I and II. *Proceedings of the Institute of Electrical Engineering*, **129 Pt. F**, 261–272.

Reilly, J. P. (1981). *Nonlinear array processing techniques with applications to correlated multipath.* Ph.D. thesis, McMaster University, Hamilton, Ontario, Canada.

Ripley, B. (1987). *Stochastic Simulation.* John Wiley & Sons, New York.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, **14**, 465–471.

Robert, C. and Casella, G. (1999). *Monte Carlo Statistical Methods.* Springer-Verlag, New York.

Ruanaidh, J. O. and Fitzgerald, W. (1996). *Numerical Bayesian Methods Applied to Signal Processing.* Springer-Verlag, New York.

Rubin, B. (1988). *Bayesian Statistics 3: Using the SIR Algorithm to Simulate Posterior Distributions (with discussions).* Oxford University Press, London.

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, **34**, 284–280.

Snedecor, G. W. and Cochran, W. G. (1980). *Statistical Methods.* Iowa State University Press, Ames, Iowa, 7th edition.

Swales, S., Maloney, J., and Stevenson, J. (1999). Locating mobile phones and the US wireless E-911 mandate. In *IEE Colloquium on Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications*, pages 2/1 – 2/6.

Tichavsky, P., Muravchik, H., and Nehorai, A. (1998). Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, **46**(5), 1386–1396.

Tranter, W. H., Rappaport, T., Woerner, B., and Reed, J., editors (1999). *Wireless Personal Communications. Emerging Technologies for Enhanced Communications.* Kluwer Academic Publishers, Norwell, MA.

Troughton, P. and Godsill, S. (1997). Bayesian model selection for time series using Markov Chain Monte Carlo. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany.

Troughton, P. and Godsill, S. (1998). A reversible jump sampler for autoregressive time series. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA.

Valaee, S. and Kabal, P. (1992). A unitary transformation algorithm for wideband array processing. In *Proceedings of Sixth IEEE SP Workshop on Statistical Signal & Array Processing*, pages 300–303, Victoria, BC.

Valaee, S. and Kabal, P. (1995). Wideband array processing using a two-sided correlation transformation. *IEEE Transactions on Signal Processing*, **44**(1), 160–172.

Valaee, S., Champagne, B., and Kabal, P. (1999). Localization of wideband signals using least-squares and total least-squares approaches. *IEEE Transactions On Signal Processing*, **47**(5), 1213–1222.

Van Trees, H. L. (2002). *Optimum Array Processing.* John Wiley & Sons, New York.

Veen, B. D. V. and Buckley, K. M. (1988). Beamforming: A versatile approach to spatial filtering.

Viberg, M. and Ottersen, B. (1991). Sensor array processing based on subspace fitting. *IEEE Transactions on Signal Processing*, **39**(5), 1110–1121.

Viberg, M., Ottersten, B., and Kailath, T. (1991). Detection and estimation in sensor arrays using weighted subspace fitting. *IEEE Transactions on Signal Processing*, **39**(11), 2436–2449.

Wang, H. and Kaveh, M. (1985). Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **33**, 823–831.

Wax, M. and Kailath, T. (1985). Detection of signals by information theoretic criteria. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **33**(2), 387–392.

Wax, M., Shan, T., and Kailath, T. (1984). Spatio-temporal spectral analysis by eigenstructure methods. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **32**(40), 817–827.

Wigren, T. and Eriksson, A. (1995). Accuracy aspects of DOA and angular velocity estimation in sensor array processing. **38**, 60–62.

Wong, K.-M., Reilly, J. P., Wu, Q., and Qiao, S. (1992). Estimation of the directions of arrival of signals in unknown correlated noise, Part 1: The MAP approach and its implementation. *IEEE Transactions on Signal Processing*, **40**(8), 2007–2017.

Wu, Q. and Wong, K.-M. (1994). UN-MUSIC and UN-CLE: An application of generalized correlation analysis to the estimation of the direction of arrival of signals in unknown correlated noise. *IEEE Transactions on Signal Processing*, **42**(9), 2331–2343.

Yang, B. (1995). An extension of the PASTd algorithm to both rank and subspace tracking. *IEEE Signal Processing Letters*, **2**(9), 179–182.

Zhao, L., Krishnaiah, P., and Bai, Z. (1987). Remarks on certain criteria for detection of the number of signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **35**(2), 129–132.