PARTICLE METHODS FOR BAYESIAN MULTI-OBJECT TRACKING AND PARAMETER ESTIMATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE ÖZKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2009

Approval of the thesis:

PARTICLE METHODS FOR BAYESIAN MULTI-OBJECT TRACKING AND PARAMETER ESTIMATION

submitted by **EMRE ÖZKAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Mübeccel Demirekler
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Mustafa Kuzuoğlu
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Hitay Özbay
Electrical and Electronics Engineering Dept., Bilkent University _____

Assist. Prof. Dr. Emre Tuna
Electrical and Electronics Engineering Dept., METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: EMRE ÖZKAN

Signature :

ABSTRACT

PARTICLE METHODS FOR BAYESIAN MULTI-OBJECT TRACKING AND PARAMETER ESTIMATION

Özkan, Emre

Ph.D., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Mübeccel Demirekler

August 2009, 103 pages

In this thesis a number of improvements have been established for specific methods which utilize sequential Monte Carlo (SMC), aka. Particle filtering (PF) techniques. The first problem is the Bayesian multi-target tracking (MTT) problem for which we propose the use of non-parametric Bayesian models that are based on time varying extension of Dirichlet process (DP) models. The second problem studied in this thesis is an important application area for the proposed DP based MTT method; the tracking of vocal tract resonance frequencies of the speech signals. Lastly, we investigate SMC based parameter estimation problem of nonlinear non-Gaussian state space models in which we provide a performance improvement for the path density based methods by utilizing regularization techniques.

Keywords: Particle Filter, Dirichlet Process, Parameter Estimation, Target Tracking

ÖZ

PARÇACIK METODLARI İLE ÇOKLU NESNE İZLEME VE PARAMETRE KESTİRİMİ

Özkan, Emre
Doktora, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Mübeccel Demirekler

Ağustos 2009, 103 sayfa

Bu tezde belli problemlerin çözümünde parçacık filtresi olarak da bilinen sıralı Monte Carlo (SMC) tekniklerini kullanan yöntemlerde iyileştirmeler yapılmıştır. Ele alınan ilk problem olan Bayes yaklaşımlı çoklu hedef izleme (ÇHİ) problemi için Dirichlet süreci (DS) temelli parametrik olmayan Bayes modellerinin kullanımı önerilmiştir. İkinci problem, önerilen DS temelli ÇHİ algoritması için önemli bir uygulama alanı teşkil eden, konuşma sinyallerinde ses yolu rezonans frekanslarını izleme problemidir. Son olarak, doğrusal ve Gauss olmayan durum uzay modellerinde parametre kestirimi amaçlı kullanılan SMC temelli bir algoritma incelenmiştir. Bu çalışma kapsamında iz yörüngesini temel olarak alan algoritmalar için regülerizasyon teknikleri kullanılarak iyileştirme sağlanmıştır.

Anahtar Kelimeler: Parçacık Filtresi, Dirichlet Süreci, Parametre Kestirimi, Hedef İzleme

In memory of my father,

ACKNOWLEDGMENTS

It is easy to get lost in one’s journey to PhD, and I could not have made it to the end without the help, encouragement, guidance and insight of my supervisor, Mübeccel Demirekler, whose influence as a mentor goes far beyond my academic life. I hope that I could be as inspiring.

I am so grateful to Eren Akdemir for all his support and friendship throughout my years in METU. He was the one I can always trust.

I am deeply indebted to Umut Orguner who never hesitated to lend a hand whenever I needed and made the hard times bearable.

I also would like to thank to all of my friends İ. Yücel Özbek, Evren İmre, Umut Özertem, Onur Yüce Gün, Ömer Köktürk, Evren Ekmekçi, Alper Koz, Ruth Yasemin Erol, Tolga Gürol, Çağatay Ata, Özge Aktukan, Eren Karaca, Suat Idil, Özgür Özgan, Soner Yeşil, Sezen Yeşil, Ayşegul Özgan, Serkan Temizel, Ayşegül Yeniaras for making my life worth living.

Lastly, I would like to thank to my family as they have always supported me by all means in my whole life. I owe them so much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Solution to many real life problems requires processing a series of observations which carry information about the unknowns. The model used in describing the problem plays an important role as it defines the implicit relation between the observations and the unknowns. The state space models have been used widely in several applications which aim to estimate a hidden state that can only be observed, maybe partly, through a set of measurements. The examples for the application areas vary in a wide range such as target tracking, communications, econometrics, biometrics etc.. The widespread use of the state space models has created a need of efficient estimation algorithms which require reasonable computation power while producing an estimate of the unknown state with an acceptable error range. The compromise in between kept the subject matter popular among both the practitioners and theoreticians and a number of algorithms have been developed over the last fifty years.

The Kalman filter (KF) is probably the most famous estimation technique in this context. Under certain linearity and Gaussian noise assumptions, it produces the optimal estimates of the state vector, in the mean square sense, with minimum variance. The increasing popularity of the KF was only confined by its inapplicability to larger class of models. The use of KF could be broadened into non-Gaussian models, as it is still the best linear unbiased estimator (BLUE) if the system is linear. However many practical problems involve non-Gaussian nonlinear models which can describe more sophisticated system dynamics or the complex relations between the state and the observations. Numerous improvements were proposed which aim to adapt KF techniques to the estimation problem of complex system models involving nonlinearities. The first idea was the local linearization of the nonlinear equations which resulted the Extended Kalman filter (EKF) [23]. The use of EKF became standard for the nonlinear models. Unfortunately, if the degree of the nonlinearity is high EKF becomes

unstable or shows poor performance as it only considers the first order terms in the Taylor series expansion of the nonlinear functions. Under severe nonlinearities, where the higher order terms are not negligible, the EKF approximation fails. In the modifications of EKF, unscented transform was utilized which comprises the method known as Unscented Kalman Filter (UKF) [25] [26]. UKF uses carefully chosen sigma points to propagate the Gaussian approximation of the filtering density at each time step and it can outperform EKF. The main weakness of UKF and EKF is that the both algorithms approximate the posterior density with a single Gaussian which might fail to represent the true posterior density if it is bi-modal or, for example, a mixture of Gaussians. This common problem of EKF and UKF has limited their application areas. Until the 90's, the literature was still lacking an estimation technique to handle the complex non-linear systems. In 1993 Gordon et. al. [20] introduced the first practical sequential Monte Carlo (SMC) based algorithm. A number of algorithms which share the similar ideas were proposed during the 90's in different fields and in different names such as condensation [22], bootstrap filters [20], particle filters [7], survival of the fittest [27], Monte Carlo filters [29], etc. These algorithms are now referred to as SMC methods or particle filters (PF). Being applicable to a very large class of models, SMC algorithms provided a powerful tool for the solution of the complex nonlinear non-Gaussian estimation problem. Unlike UKF or EKF SMC methods do not rely on functional approximations or local linearizations. The relevant distributions are approximated by discrete random samples, namely the particles, and their weights. As an example, let $p(x)$ be a probability density function to be approximated by N particles. The particle approximation to the probability density function $p(x)$ is in the form.

$$\tilde{p}(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i) \tag{1.1}$$

where N is the number of particles used in the approximation, $\{x^i, i = 1 : N\}$ is a set of support points with associated weights $\{w^i, i = 1 : N\}$ and $\delta(.)$ is the Dirac delta function. A graphical example is given in Figure 1.1 and Figure 1.2. The discrete approximation to continuous density enjoys several properties. For example, the computation of expectations is simplified to summations.

$$\mathcal{E}\{f(x)\} = \int f(x)p(x)dx \tag{1.2}$$

is approximated by

$$\widetilde{\mathcal{E}}\{f(x)\} = \sum_{i=1}^{N} w^i f(x^i). \tag{1.3}$$

where $f(.)$ is some useful function for estimation.

Furthermore, the SMC methods are quite easy to implement but computationally quite demanding. As the computational power got more available and cheaper the application areas of SMC methods got wider. Now the SMC methods are standard methods for the problems which involve complex nonlinear non-Gaussian models. The amazing increase in the computational power made the use of SMC possible even for the real time applications. Moreover SMC algorithms can be used to make inference from complex distributions.

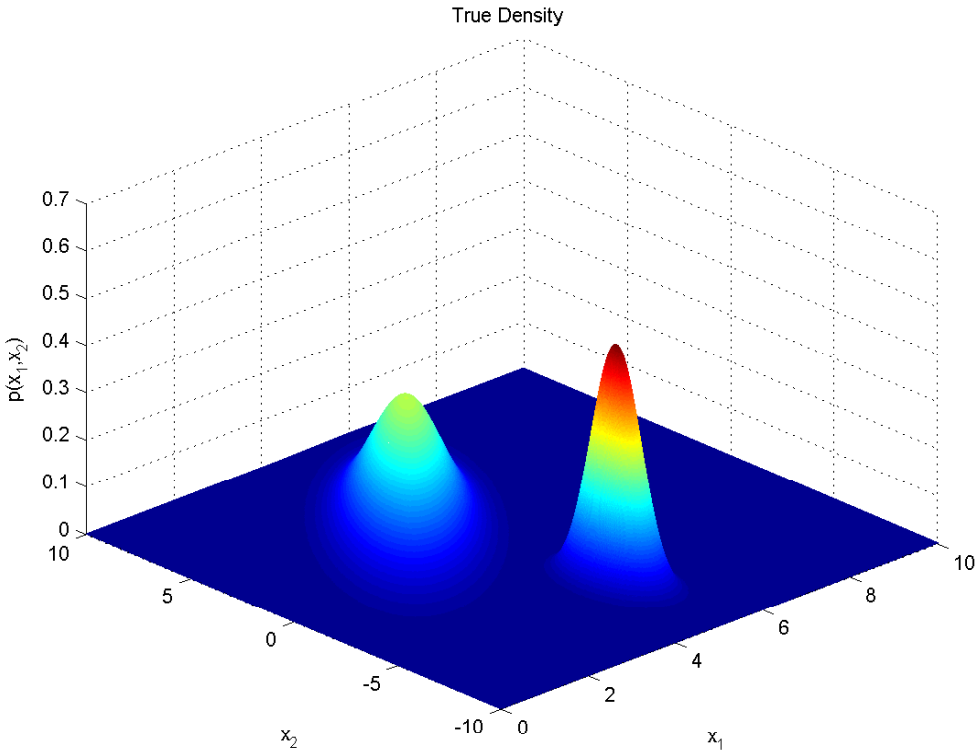


Figure 1.1: True probability density function.

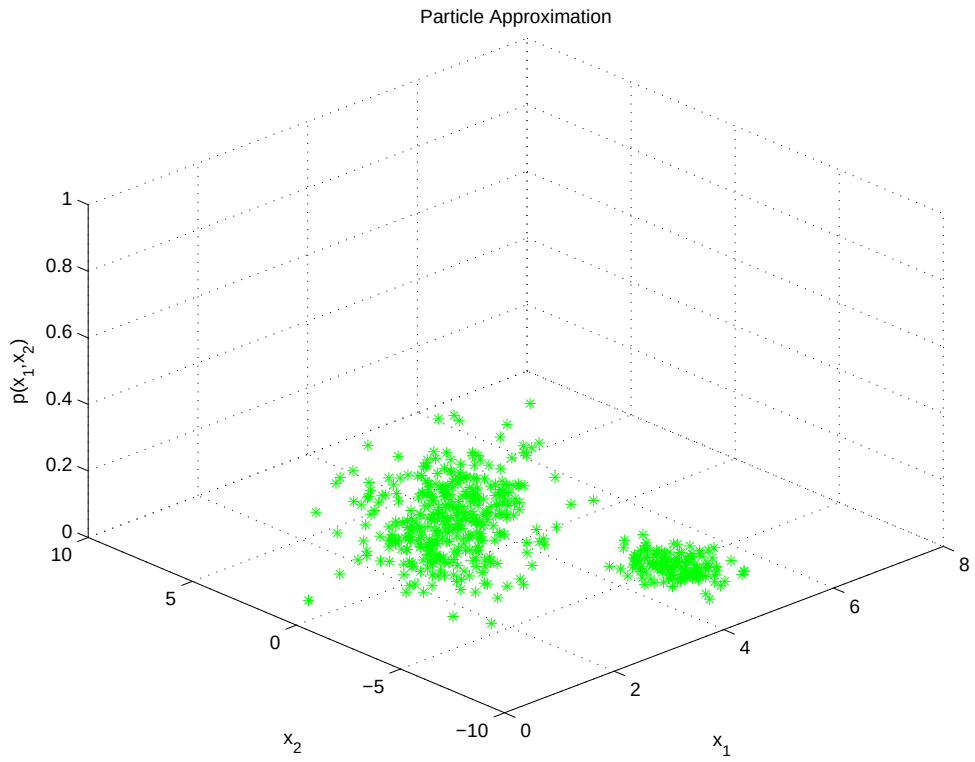


Figure 1.2: Approximated probability density function represented by particles having equal weights

This thesis presents solutions to problems of the statistical signal processing algorithms that utilize particle filtering methods in approximating complex distributions and the functions of these distributions. The first problem we concentrate on is the problem of unknown number of components in a mixture density estimation. This type of scenario appears in many practical signal processing examples. The specific applications that we will consider here is full Bayesian multi-target tracking. The second application that we will focus on is tracking of formant frequencies in speech processing. The third problem is the estimation of unknown parameters of general state space models which is encountered in most signal processing applications. We apply our particle filter based solution to the parameter estimation of general state space models. A brief summary of the topics covered within this dissertation are as follows.

1.1 Bayesian Nonparametric Models for Multi-target Tracking

In Chapter 2 we propose a Dirichlet process based multi-target tracking algorithm which aims to track unknown number of targets in a surveillance region. Dirichlet processes are widely used in classification, clustering and mixture density estimation problems in statistics [13, 24]. The proposed model relies on an extension of Dirichlet processes, namely time varying Dirichlet processes. The model naturally handles the track initiation/deletion tasks of the multi-target tracking problem whereas the existing algorithms use either some ad-hoc logic or their probabilistic variants. We define a user presentation logic which keeps the identity of the targets by merging the best hypothesis produced by the algorithm in consecutive time instants. In the Bayesian approach to multi-target tracking, probabilistic models are used in order to cast the multi-target problem into a Bayesian estimation problem. Consequently, the resulting posterior distributions are too complex to be expressed analytically hence the sequential Monte Carlo methods are utilized to approximate the relevant distributions.

1.2 Dynamic Speech Spectrum Representation and Tracking Variable Number of Vocal Tract Resonance Frequencies with Time Varying Dirichlet Process Mixture Models

In this research, we extend our previous study about multi-target tracking to the formant tracking problem in speech signal processing literature. Formants, being the resonance frequencies of the vocal tract, carry important information about uttered speech and the speaker. Existing formant tracking algorithms aim to track fixed number of formants. The resulting model is incapable of representing and adapting to the varying structure of the formants in speech signals. We develop a new approach for tracking vocal tract resonance (VTR) frequencies which is based on representing the spectral density of the speech signals with time varying Dirichlet process mixture models. The method involves modeling the speech signal spectrum by an unknown number of mixture of Gaussians, for which the Dirichlet process mixture model is utilized. The vocal tract resonance frequencies are detected from the estimated spectral density and tracking of the resonance frequencies is performed for speech utterances. In this work we assume that the number of mixture components in the spectral density of the speech signals varies in time as the vocal tract resonance frequencies appear/disappear due to the pole zero cancelations and observability issues. Therefore, we aimed to establish a method which is flexible enough to allow varying number of mixture components in the estimated spectral density. Dirichlet process defines a distribution over probability measures with possibly infinite number of mixtures and is capable of adapting the number of mixture components with the incoming data. Consequently the prior knowledge of the number of mixture components is obviated.

1.3 Regularized Particle Methods for Filter Derivative Approximation

Estimation of static parameters in non-linear non-Gaussian general state space models via the particle methods have remained a long standing problem in the literature. Various attempts are made to achieve the difficult task of jointly estimating both the state and the model parameters. Here we consider a gradient based stochastic approximation algorithm which aims to find the maximum-likelihood estimate of the unknown static parameters. The method proposed

here approximates the path-density and its derivative by a set of particles and utilize kernel smoothing techniques to prevent the degeneracy of the algorithm which would cause error accumulation and leads the algorithm to diverge in time.

CHAPTER 2

BAYESIAN NONPARAMETRIC MODELS FOR MULTI-TARGET TRACKING

2.1 Introduction

This research is mainly focused on Dirichlet process and its application to Multi-target tracking (MTT) problem. Dirichlet processes are a very popular class of models in non-parametric Bayesian statistics and are widely used in statistics, population genetics, machine learning, etc. for density estimation and clustering. In this research, we aimed to adapt Dirichlet process models to the multi-target tracking problem. In a full Bayesian approach to multi-target tracking, the time-varying number of targets and the dynamics of these targets are modeled using probabilistic models. Although this kind of approach might seem neat and well structured, the resulting posterior distributions are too complex to be expressed analytically therefore they are generally intractable. The ability of particle filtering methods to approximate the complex distributions made the realization of full Bayesian approach to the multi-target tracking problem possible. In this context, the probabilistic models for the targets dynamics are well-established. On the contrary, the models used to model the time-varying number of targets have been overlooked. The existing models are either based on ad-hoc logic or their probabilistic variations. The classical (M/N) logic oversimplifies the track initiation/deletion problem in MTT. Existing probabilistic models used for track initiation/deletion procedures are also somehow deficient as they may define non-stationary priors. We propose the use of a new class of models relying on time-varying Dirichlet processes which has attractive properties. The measurement to track association task and track initiation/deletion procedures are handled naturally by Dirichlet process models. In most of the target tracking applications, target identities are required to be maintained as new targets appear/disappear in the surveil-

lance region. The previous methods also lack the output presentation stage. In the proposed algorithm we define a user presentation logic which aims to keep a unique identity for each of the active targets in the region by combining the best hypothesis at consecutive time instants. The resulting algorithm is novel in many ways and it defines a complete tracking system which is able to initiate/delete tracks in full Bayesian framework. The inference is done using Rao-Blackwellized particle filtering technique so that the algorithm can be implemented efficiently and can be run real-time and on-line. In our experiments we show that the algorithm performs better than joint probabilistic data association (JPDA) and global nearest neighborhood (GNN) algorithms which use standard (M/N) ad-hoc logic for track initiation and deletion procedures. In addition to its capability of defining a mathematical model for track deletion/initiation tasks, the proposed method can keep multiple hypotheses for track to measurement/clutter association and it is able to outperform both JPDA and GNN algorithms under heavy clutter. The chapter is organized as follows. In the first section the background information for the Bayesian Inference and the Dirichlet processes are given. In the second section, multi-target tracking problem is introduced, in the following sections the description of the proposed algorithm is given and the chapter is concluded with the simulations and discussions sections.

This research was conducted jointly with François Caron and Arnaud Doucet of University of British Columbia (UBC), Vancouver Canada. The author would like to thank for their collaboration.

2.2 Background

2.2.1 Dirichlet Processes

In the Bayesian context, Dirichlet processes (DP) are known to be a specific prior used for mixture models. Among all the properties which makes DP applicable to wide range of areas, probably the most important one is its capability to model the infinite mixtures. The ability of modeling the infinite mixtures, makes DP a good choice as a prior distribution, which is meant to be flexible enough to capture the various structures in mixture models. Before proceeding to the description of DP, some basics of Bayesian inference is reviewed here.

2.2.2 Bayesian Inference

Bayesian inference is a statistical inference, in which the observed data and the prior information are used together to infer the probability of a belief or a hypothesis. The inference is based on Bayes Rule which is formulated as follows:

$$p(G|D) = \frac{p(D|G) \times p(G)}{p(D)} \tag{2.1}$$

In equation (2.1), $p(G)$ is called the *prior distribution*; $p(D|G)$ is called the *likelihood function*; $p(D)$ is called the *marginal distribution* of the observed data, and $p(G|D)$ is called the *posterior distribution* of G given the data D.

In Bayesian approach one first defines a prior distribution which is intended to represent our prior beliefs about the events before observing the data. After observing some data, Bayes Rule is applied to obtain the posterior distribution which takes both the prior information and the data into account. From this posterior distribution one can also compute predictive distributions for future observations.

$$posterior \propto prior \times likelihood$$

Here we present a simple example, an extended version of which will be related to Dirichlet Processes later in this chapter. This example is the same problem considered by Bayes in proposition (9) of his essay published in 1764[1].

In this example we consider the computation of the posterior distribution of a binomial parameter. We are given n observed success out of m binomial trials and we want to find an estimate for the probability of success p.

In frequentiest approach a reasonable estimate for p would be $\frac{n}{m}$ which is also equal to the maximum likelihood (ML) estimate. Suppose that the prior information about the parameter p is given by a Beta distribution.

$$Beta(p;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\times\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{2.2}$$

Where $0 \leq p \leq 1, \quad \alpha,\beta > 0, \quad \Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ is the standard Gamma function, and α and β are called the shape parameters. The fractional term is independent of p which acts as

[1] Although Thomas Bayes lost his life in 1761, his works were published posthumously by his friend Richard Price in the Philosophical Transactions of the Royal Society of London in 1764.

a normalizing constant such that the probability density function sums up to 1. The posterior distribution of p is calculated as follows.

$$\begin{aligned} p(p|D) &= \frac{\binom{m}{n} p^n (1-p)^{m-n} \times \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\times\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}}{p(D)} \\ &= c \times p^{n+\alpha-1}(1-p)^{m-n+\beta-1}, \quad c \to constant \\ &\propto p^{n+\alpha-1}(1-p)^{m-n+\beta-1} \\ &= Beta(p; n+\alpha, m-n+\beta) \end{aligned} \tag{2.3}$$

The posterior distribution takes its maximum value for $p = \frac{n+\alpha}{m-n+\beta}$, that is also equal to the maximum a posteriori (MAP) estimate of p. The MAP estimate of p can be used to predict the $(n+1)^{th}$ trial, based on the first n observations and the prior. Note that the posterior distribution is also a Beta distribution with modified parameters which results from the conjugacy between the binomial and Beta distributions. A family of prior distributions is conjugate to a particular likelihood function if the posterior distribution belongs to the same family as the prior. Beta family is conjugate to the Binomial likelihood.

At this point it is important to notice that the repeated Bernoulli trials of this form can also be modeled using Hierarchical Bayesian models.

Let y_i be the random variable indicating success for the i^{th} trial.

$$y_i = \left\{ \begin{array}{ll} 1 & \text{with probability p} \\ 0 & \text{with probability (1-p)} \end{array} \right\} \tag{2.4}$$

Combining this definition with the prior distribution of p results:

$$p \sim \quad Beta(\alpha, \beta) \tag{2.5}$$

$$y_i \sim \quad Binomial(p) \tag{2.6}$$

or equivalently,

$$p \sim \quad Beta(\alpha, \beta) \tag{2.7}$$

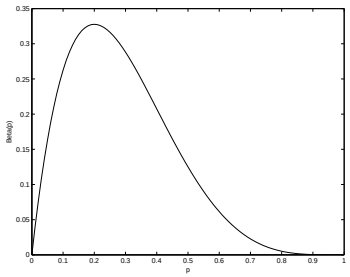
$$y_i \sim \quad G(.|p) \tag{2.8}$$

Where

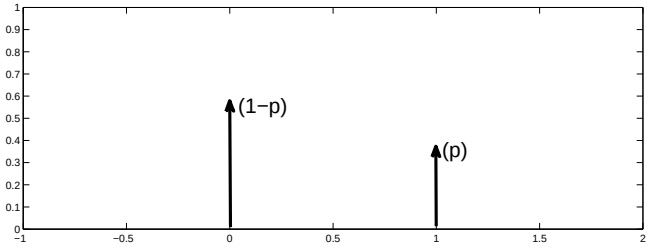
$$G(.|\pi, \phi) = \sum_{i=1}^{2} \pi_i \delta_{\phi_i}(.) \tag{2.9}$$

and $\pi_1 = p$, $\pi_2 = 1-p$, $\phi_1 = 0$, $\phi_2 = 1$ and $\delta_\phi(.)$ is the delta dirac function at point ϕ.

Such a representation will be beneficial later in working with the mixture models. Our aim is to consider the variable p as the probability that y_i is a member of group 1, and $(1-p)$ as the



(a)



(b)

Figure 2.1: Illustration of the Hierarchical Bayesian model. (a) The prior distribution of p, Beta(p;1,4). (b) The distribution $G(.|p)$ from which the y_i's are sampled.

probability that y_i is a member of group 0.

2.2.3 Dirichlet - Multinomial Model

In the next example, the classification of a number of measurements, y_i's, which are originated from K different classes will be investigated. The structure will be formulated as a Hierarchical Bayesian Model and the Bayesian inference will be applied with appropriate prior distributions.

The model for the classification problem will be constructed as follows. There exist K classes which are of interest. Given a specific class and its relevant parameters, the distribution of the measurements originating from that class is assumed to be known.

$$y_i|\phi, c_i \sim f(.|\phi_{c_i}) \tag{2.10}$$

Where ϕ_{c_i} stands for the parameters of the c_i^{th} class, c_i is called the allocation variable and it takes the values $1, \ldots, K$. The allocation variable indicates the identity of the class associated

to the measurement y_i. Each measurement belongs to one of the K classes with some unknown probabilities denoted by the vector π. Probability that the i^{th} observation is originated from the class j is equal to π_j. i.e. $p(c_i = j|\pi_j) = \pi_j$. π_j's are called the mixing coefficients or the weights which sum up to 1.

$$\begin{aligned} c_i &\sim G(.|\pi) \\ \text{where } G(.|\pi) &= \sum_{j=1}^{K} \pi_j \delta_j(.), \qquad \pi = [\pi_1, \pi_2, \ldots, \pi_K]. \end{aligned} \tag{2.11}$$

We will assume a Dirichlet distribution as a prior distribution for these unknown mixing coefficients π. *Remember the variable p and the Beta distribution in Bernoulli trials.* The Dirichlet distribution is a higher dimensional version of the Beta distribution. The Beta distribution is defined on 2-dimensional simplex, and it corresponds to the special case of the Dirichlet distribution where the dimension is equal to 2. The general form of a K-dimensional Dirichlet distribution with parameters $\alpha_1, \ldots, \alpha_k$ is given by

$$Dirichlet(p_1, \ldots, p_k|\alpha_1, \ldots, \alpha_k) = \frac{1}{\frac{\prod_{j=1}^{k}\Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{k}\alpha_j)}} \prod_{j=1}^{k} p_j^{\alpha_j - 1} \tag{2.12}$$

The fractional term $\frac{\prod_{j=1}^{k}\Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{k}\alpha_j)}$ is constant and equal to $\int \prod_{j=1}^{k} p_j^{\alpha_j - 1} dp$, assuring that the distribution integrates to 1. In standard applications, symmetric dirichlet distribution in which the α parameters are equal is used, i.e. $\alpha_1 = \alpha_2 = \ldots = \alpha_K = \frac{\alpha}{K}$.

The parameters, ϕ_j's are sampled independently from the *base distribution*, $G_0(.)$

$$\phi_{c_i} \sim G_0(.). \tag{2.13}$$

The base distribution provides our prior knowledge over the parameters which are likely to occur. With the above definitions, the resulting Hierarchical Model becomes

$$\pi|\alpha, K \sim Dirichlet(\frac{\alpha}{K}), \tag{2.14}$$

$$c_i|\pi \sim Multinomial(\pi), \tag{2.15}$$

$$\phi_j|G_0 \sim G_0(.), \tag{2.16}$$

$$y_i|\phi, c_i \sim f(.|\phi_{c_i}). \tag{2.17}$$

We can also write $c_j|\pi \sim G(.|\pi)$,(see: equation (2.11)), instead of the Multinomial distribution. Our next aim is to compute the prediction density of the allocation variable for the future observations. Given the association of the first $m-1$ measurements to the classes (shown by

$c_{-m} \triangleq \{c_i\}_{i=1}^{m-1}$), we want to compute the conditional probability $p(c_m = j|c_1, \ldots, c_{m-1})$. For that purpose we want to integrate out the mixing coefficients π, in the joint distribution as follows.

$$p(c_m = j|c_{-m}, \alpha, K) = \int p(c_m = j|\pi)p(\pi|c_{-m}, \alpha, K)d\pi \tag{2.18}$$

Note that $p(c_i = j|\pi) = \pi_j$. The second term in the integral is the posterior distribution of the mixing coefficients π, which is equal to

$$p(\pi|c_{-m}, \alpha, K) = \frac{p(c_{-m}|\pi)p(\pi|\alpha, K)}{p(c_{-m}|\alpha, K)} \tag{2.19}$$

Denominator term is constant therefore,

$$\begin{aligned}
p(\pi|c_{-m}, \alpha, K) &\propto p(c_{-m}|\pi)p(\pi|\alpha, K) \\
&= (\prod_{i=1}^{m-1} \pi_{c_i})\, p(\pi|\alpha, K) \\
&= (\prod_{i=1}^{m-1} \pi_{c_i})\, Dirichlet(\pi; \alpha, K) \\
&= (\prod_{i=1}^{m-1} \pi_{c_i})\, \frac{\Gamma(\alpha)}{\prod_{l=1}^{K} \Gamma(\frac{\alpha}{K})} \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1} \\
&= \frac{\Gamma(\alpha)}{\prod_{l=1}^{K} \Gamma(\frac{\alpha}{K})} (\prod_{i=1}^{m-1} \pi_{c_i}) \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1}
\end{aligned}$$

Let n_i indicate the number of measurements previously assigned to the class i. Then,

$$\begin{aligned}
p(\pi|c_{-m}, \alpha, K) &\propto \frac{\Gamma(\alpha)}{\prod_{l=1}^{K} \Gamma(\frac{\alpha}{K})} (\prod_{j=1}^{K} \pi_j^{n_j}) \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1} \\
p(\pi|c_{-m}, \alpha, K) &\propto \frac{\Gamma(\alpha)}{\prod_{l=1}^{K} \Gamma(\frac{\alpha}{K})} \prod_{j=1}^{K} \pi_j^{n_j+\frac{\alpha}{K}-1}
\end{aligned} \tag{2.20}$$

which is a Dirichlet distribution with modified parameters $n_j + \frac{\alpha}{K}$. The result follows from the multinomial Dirichlet conjugacy. Returning back to equation (2.18),

$$\begin{aligned}
p(c_m = j|c_{-m}, \alpha, K) &= \int \pi_j \frac{\Gamma(\sum_{i=1}^{K}(\frac{\alpha}{K} + n_i))}{\prod_{i=1}^{K} \Gamma(\frac{\alpha}{K} + n_i)} \prod_{z=1}^{K} \pi_z^{\frac{\alpha}{K}+n_z-1} d\pi \\
&= \frac{\Gamma(\sum_{i=1}^{K}(\frac{\alpha}{K} + n_i))}{\prod_{i=1}^{K} \Gamma(\frac{\alpha}{K} + n_i)} \int \pi_j \prod_{z=1}^{K} \pi_z^{\frac{\alpha}{K}+n_z-1} d\pi \\
&= \frac{\Gamma(\sum_{i=1}^{K}(\frac{\alpha}{K} + n_i))}{\prod_{i=1}^{K} \Gamma(\frac{\alpha}{K} + n_i)} \times \frac{\prod_{i=1, i\neq j}^{K} \Gamma(\frac{\alpha}{K} + n_i)\Gamma(\frac{\alpha}{K} + n_j + 1)}{\Gamma(\sum_{i=1, i\neq j}^{K}(\frac{\alpha}{K} + n_i) + (\frac{\alpha}{K} + n_j + 1))} \\
&= \frac{n_j + \frac{\alpha}{K}}{m - 1 + \alpha}
\end{aligned} \tag{2.21}$$

Where the last line follows from the property $\Gamma(y + 1) = y\Gamma(y)$;

2.2.4 Extension to infinite dimensional priors

So far we have seen that, based on the association of the previous measurements, our prior for the future measurements to be associated with a class j, is proportional to the number of measurements previously assigned to that class plus $\frac{\alpha}{K}$ (2.21). The next step is to extend our model for infinite number of classes. Our aim is to loose the restrictions brought by breaking the assumption of fixed and known number of classes, which is not realistic for the practical cases. For the extension to the infinite dimensional case, our approach will be letting the unknown $K \to \infty$. In this case, the vector of mixing coefficients π is of infinite dimension but it is still possible to compute explicitly the marginal distribution of the allocation variables. Returning back to equation (2.21),

$$\begin{aligned} p(c_m = j | c_{-m}, \alpha) &= \lim_{K\to\infty} \left(\frac{n_j + \frac{\alpha}{K}}{m - 1 + \alpha}\right) \\ &= \frac{n_j}{m - 1 + \alpha} \end{aligned} \tag{2.22}$$

The next step is to calculate the probability that the new measurement to be originated from one of the infinite number of classes, which has not been associated to any of the measurements yet. Suppose that only n of the classes have been observed, and there are $K - n$ classes having no measurements. Then,

$$\begin{aligned} p(c_m = c_{new} | c_{-m}, \alpha) &= \lim_{K\to\infty} \sum_{z=n+1}^{K} \left(\frac{n_z + \frac{\alpha}{K}}{m - 1 + \alpha}\right) \\ &= \lim_{K\to\infty} \sum_{z=n+1}^{K} \left(\frac{0 + \frac{\alpha}{K}}{m - 1 + \alpha}\right) \\ &= \frac{\alpha}{m - 1 + \alpha} \lim_{K\to\infty} \left(\frac{K - n}{K}\right) \\ &= \frac{\alpha}{m - 1 + \alpha} \end{aligned} \tag{2.23}$$

The resulting scheme is known as the Chinese Restaurant Process (CRP) in which n customers sit down in a Restaurant with an infinite number of tables. CRP scheme can be summarized as follows.

The first customer sits at the first table. The subsequent customers either sit at one of the previously occupied tables or a new table. Suppose there are already n customers sitting at K tables. Let n_i, $i = 1, \ldots, K$ denote the number of customers sitting at table i. When $(n + 1)^{th}$ customer enters the restaurant he sits at,

- table 1 with probability $\frac{n_1}{n+\alpha}$

  $\vdots$

- table K with probability $\frac{n_K}{n+\alpha}$

- a new table with probability $\frac{\alpha}{n+\alpha}$

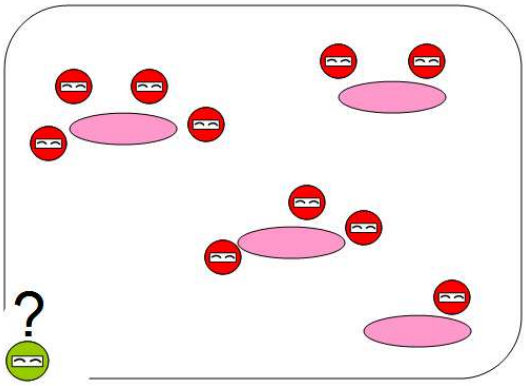
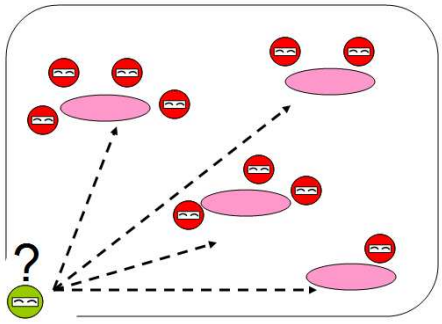
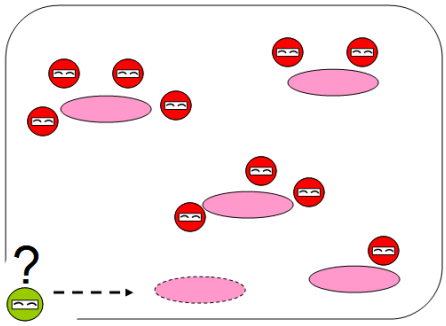


Figure 2.2: Chinese Restaurant Metaphor



(a)



(b)

Figure 2.3: Illustration of the Chinese Restaurant Process. (a) The customer sits one of the previously occupied tables, with probability proportional to the number of people sitting at that table. (b)The customer sits at a new table with probability proportional to α.

CRP is closely related to the predictive distribution of Dirichlet processes which is introduced in the following subsection.

2.2.5 Dirichlet Process Definition

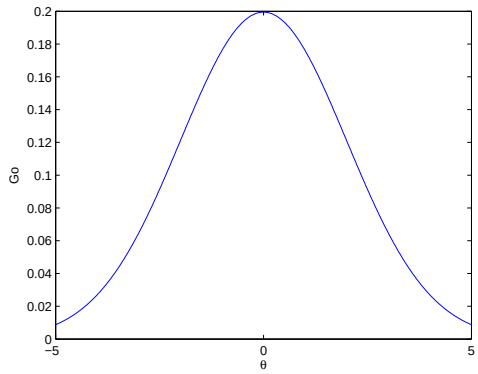
Formal definition of Dirichlet processes are given as follows.

Let (Ω, B) be a measurable space, with G_0 a probability measure on the space, and let α be a positive real number. A Dirichlet process is the distribution of a random probability measure G over (Ω, B) such that, for any finite partition $(A_1, A_2, ..., A_n)$ of Ω, the random

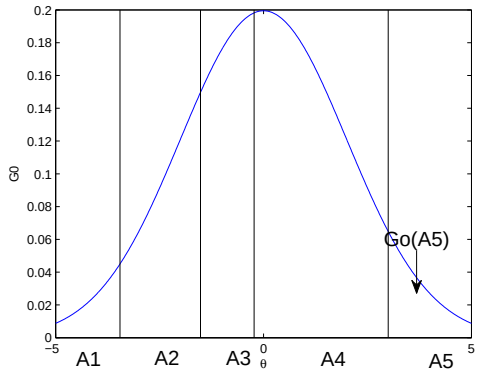
vector $(G(A_1), G(A_2), ..., G(A_n))$ is distributed as a finite-dimensional Dirichlet distribution:

$$(G(A_1), \ldots, G(A_n)) \sim Dir(\alpha G_0(A_1), \ldots, \alpha G_0(A_n)). \tag{2.24}$$

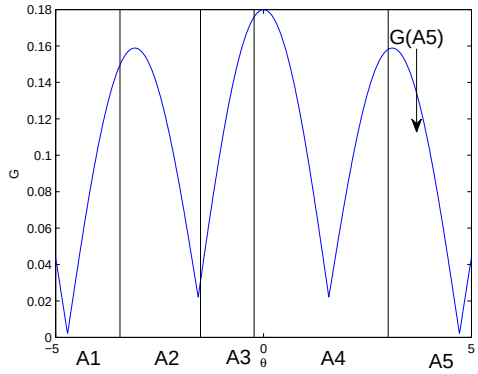
In order to illustrate the conditions implied by equation (2.24), an example is given as follows. Let Ω be $\mathbb{R}$, and B is defined in the usual way as the Borel Field including all open subsets of the real line. Let G_0 be a normal distribution with zero mean and variance σ_v^2. Then for any partitioning of Ω, G Measure of the partitions should be distributed according to a Dirichlet distribution with parameters $\alpha G_0(A_i)$. See Figure 2.4



(a)



(b)



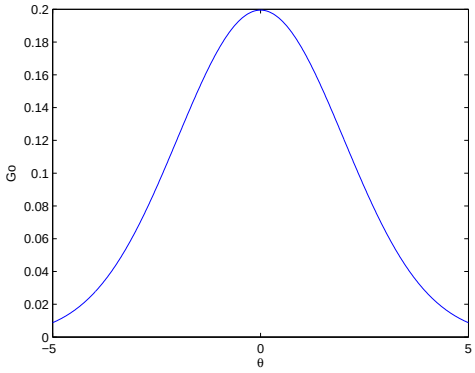
(c)

Figure 2.4: g. (a) The Base Distribution G_0. (b) One random partitioning of the real line. (c) G Measure of the partitions.

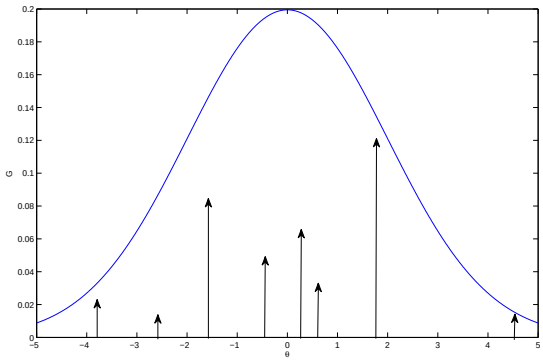
Since G itself is a probability measure, Dirichlet Process defines a distribution over probability distributions. More specifically Dirichlet Process defines the probability measure G to be discrete with probability 1 [43].

$$G(\Phi) = \sum_{k=1}^{\infty} \pi_k \delta(\phi_k), \qquad \text{where} \quad \Phi = [\phi_1, \phi_2, ...]. \tag{2.25}$$

Therefore a more realistic picture for a probability distribution G sampled from a Dirichlet Process would look like the one in Figure 2.5-b.



(a)



(b)

Figure 2.5: (a)The Base Distribution G_0. (b) G: One sample of a Dirichlet Process.

Returning back to our hierarchical model, for the infinite classes case, the model can be specified with the following equations.

$$G|G_0, \alpha \sim DP(G_0, \alpha) \tag{2.26}$$

$$\phi_i|G \sim G(.) \tag{2.27}$$

$$y_i|\phi_i \sim f(.|\phi_i). \tag{2.28}$$

What we are interested in is again the posterior distribution of G after observing a number of measurements. Similar to the finite dimensional case, because of the mutinomial-dirichlet conjugacy, the posterior distribution of G is again a Dirichlet Process with modified parameters.

$$\begin{aligned} G|G_0,\alpha &\sim DP(G_0,\alpha) \\ P(G) &= DP(G|G_0,\alpha) \\ P(G|\Phi) &= \frac{P(\Phi|G)P(G)}{P(\Phi)} \\ P(G|\phi_1,\ldots,\phi_n) &= DP(\bar{G_0},\bar{\alpha}) \\ \bar{G_0} &= \frac{\alpha}{\alpha+n}G_0 + \frac{1}{\alpha+n}\sum_{i=1}^{n}\delta(\phi_i), \\ \bar{\alpha} &= \alpha+n \end{aligned}$$

If we integrate out G, the prediction density for the $(n+1)^{th}$ observation will be

$$\begin{aligned} P(\phi_{n+1}|\phi_1,\ldots,\phi_n,G_0,\alpha) &\propto \int\prod_{i=1}^{n}P(\phi_i|G)p(G)dG \\ (\phi_{n+1}|\phi_1,\ldots,\phi_n,G_0,\alpha) &\sim \frac{\alpha}{\alpha+n}G_0 + \frac{1}{\alpha+n}\sum_{i=1}^{n}\delta(\phi_i). \end{aligned} \tag{2.29}$$

Notice that, the method for generating this sequence of random variables in (2.29) is exactly the same as the Chineese Restaurant Process, where the parameters of a new table is determined by the base distribution G_0.

A sequence of parameters generated according to (2.29) is also said to be sampled from a Polya urn, which is described in the following section.

2.2.6 Polya Urn Representation

Polya urn scheme [6] is analogous to the Chineese Restaurant Process, and is used to model the process of generating the samples from the predictive distribution of the Dirichlet process. The metaphor used for describing the process of sampling from the predictive distribution via Polya urn scheme is as follows:

Suppose there are n balls in an urn with m different colors (*colors are indicated by numbers 1 to m*). Let n_i denote the number of balls having the color i. Assume the m^{th} color is black, and the number of black balls in the urn (n_m) is equal to α. Each time, one ball is picked

from the urn. If that ball is black, we put a new ball with a new color into the urn, otherwise we put a ball having the same color into the urn. We put the ball back into the urn as well. Therefore, when we pick up a ball, the probability that its color will be one of the existing colors is proportional to the number of balls sharing the same color, and the probability of adding a new color to the urn is proportional to the number of black balls in the urn, i.e. α.

2.2.7 Stick Breaking Representation

Another realization of the Dirichlet Processes can be done via stick breaking representation of the random probability measure G. Sethuraman showed in his paper [43] a constructive way of representing the samples from the Dirichlet Processes as follows.

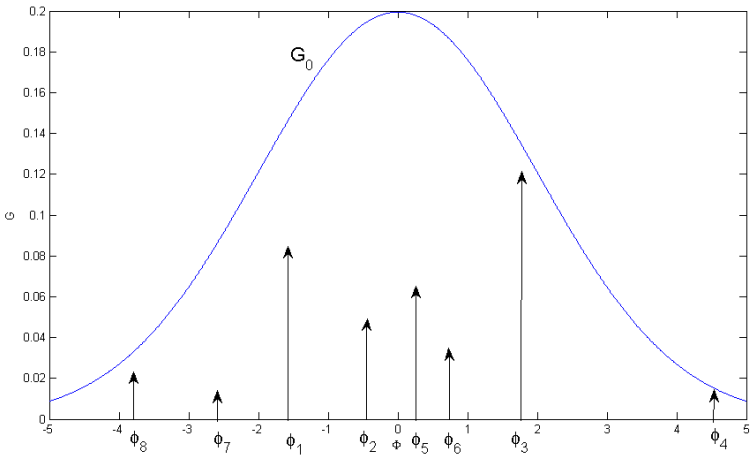
$$G = \sum_{k=1}^{\infty} \pi_k \delta(\phi_k) \tag{2.30}$$

$$\text{where} \quad \phi_k \sim G_0 \tag{2.31}$$

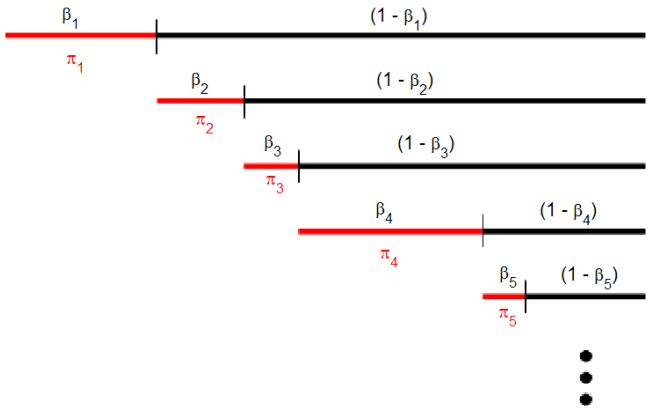
$$\pi_k = \beta_k \prod_{j=1}^{k-1} (1 - \beta_j) \tag{2.32}$$

$$\beta_k \sim Beta(1, \alpha) \tag{2.33}$$

The definition of the weights (π_i's) in equation (2.32) can be visualized as an analogy to a stick breaking process in the following way. We begin with a stick of length 1. Generate a random variable β_1 from the Beta distribution. Break the stick into two pieces such that the length of the resulting pieces are β_1 and $(1 - \beta_1)$. The first weight π_1, is equal to the length of the first piece, β_1. Take the other piece and break it into two pieces such that the length of the resulting pieces are proportional to β_2 and $(1 - \beta_2)$, where β_2 is another random sample from the Beta distribution. The length of the first piece is equal to π_2. Repeat the procedure with a countably infinite number of brakes and generate the sequence of weights π_i's. Note that, since we began with a stick of length 1, the sum of the weights generated in this process will be equal to 1. Therefore the resulting measure G, will be a valid probability measure. The pictorial representation of the stick breaking process is given in 2.6.



(a)



(b)

Figure 2.6: Stick Breaking Representation of $G = \sum_{k=1}^{\infty} \pi_k \delta(\phi_k)$ (a) The samples ϕ_k's sampled from G_0. (b) Generating the weights in stick breaking representation.

2.3 Multi-target Tracking

Multi-target tracking problem consists of estimating the states of a possibly unknown number of targets in a given surveillance region. The available sensors generally collect several measurements which may correspond to some clutter or the targets of interest if detected. The sensors typically lack the ability to associate each measurement to its originating target or the clutter. Hence, one has to solve simultaneously three problems: data association, estimating the number of targets in the surveillance region, and the estimation of the target states. Numerous algorithms have been proposed in the literature to address such problems and only the ones that are most relevant to our approach will be reviewed here; see [21] for a detailed review of the literature or [5] for pre-particle filter era.

Here, we propose a new class of stationary models for the time-varying number of targets. Proposed class of prior models relies on a time-varying extension of Dirichlet Processes. In scenarios where the number of targets is unknown, potentially infinite, but fixed, a DP prior has recently been used to perform Bayesian multitarget tracking [17]. In these papers, batch inference is performed using Markov chain Monte Carlo (MCMC). We propose here an original time-varying extension of DP to model the time-varying number of targets. Moreover, from a practical point of view, this process is easily interpretable and allows us to naturally handle birth/death of targets over time. We utilize Sequential Monte Carlo (SMC)-type algorithms to approximate on-line the resulting posterior distributions.

2.4 Known Number of Targets

Assume there are K targets of interest in the surveillance region. Let $x_{j,t}$ denote the state vector of target j at time t. The target dynamics are defined by state dynamics equation.

$$x_{j,t} \big| \, x_{j,t-1} \sim f\left(\cdot \big| \, x_{j,t-1}\right) \tag{2.34}$$

where $f(\cdot \,|\, x)$ is a probability density function for a given x. To define the measurement model, we again introduce the set of unobserved allocation variables $\{c_{k,t}\}$ where $k = 1, ..., m_t$ and m_t is the number of measurements at time t such that if $c_{k,t} = j$ then $y_{k,t}$ originates from target j. Formally, conditional upon $c_t = \{c_{k,t}\}$ and $x_t = \left\{x_{j,t}\right\}$, we assume that the measurements are

statistically independent and marginally distributed according to

$$y_{k,t}\big|\,(c_t, x_t) \sim g\left(\cdot|\, x_{c_{k,t},t}\right) \tag{2.35}$$

where $g(\cdot|\, x)$ is a probability density function for any x. To simplify the presentation, we do not include some clutter noise here. To complete this Bayesian model, we need to define a prior distribution for the allocation variables $\{c_{k,t}\}$. A simple assumption consists of assuming that, at any time t, we have a vector of prior probabilities $\pi_t = \{\pi_{j,t}\}$, i.e. $\pi_{j,t} \geq 0$ and $\sum_{j=1}^{K} \pi_{j,t} = 1$ and that

$$p(c_{1:T}|\,\pi_{1:T}) = \prod_{t=1}^{T} p(c_t|\,\pi_t) \tag{2.36}$$

where

$$p(c_t|\,\pi_t) = \prod_{k=1}^{m_k} p\left(c_{k,t}\big|\,\pi_t\right) \tag{2.37}$$

with

$$p\left(c_{k,t} = j\big|\,\pi_t\right) = \pi_{j,t}, \tag{2.38}$$

that is c_t follows a multinomial distribution of parameters (π_t). The PMHT algorithm uses the Expectation-Maximization algorithm to maximize the marginal distribution $p(x_{1:T}|\,y_{1:T}, \pi_{1:T})$ for a fixed T [44, 19]. In [21], this model is completed by introducing an additional prior distribution for $\pi_{1:T}$. The vectors π_t are assumed independent and identically distributed according to

$$\pi_t \sim Dirichlet\left(\frac{\alpha}{K}\right)$$

for some $\alpha > 0$. In this case, the target distribution of interest at time T is $p(x_{1:T}, \pi_{1:T}|\,y_{1:T})$. A combination of MCMC and SMC algorithms is proposed in [21] to sample from the sequence of these target distributions as T increases. The proposed algorithm is quite computationally intensive as it requires running an MCMC algorithm until convergence at each time step.

We consider here for the sake of illustration and so as to introduce later on the DP the static case where $\pi_t = \pi$ for any t and

$$\pi \sim Dirichlet\left(\frac{\alpha}{K}\right). \tag{2.39}$$

Since $c_{k,t}$'s follow multinomial distribution likewise the model defined by equations (2.14) and (2.15). The prediction distribution of the allocation variables can be computed according to the equation (2.21).

Then for any $j \in \{1, ..., K\}$

$$\Pr(c_{k,t} = j | c_{-k,t}) = \frac{n^j_{-k,t} + \alpha/K}{\sum_i n^i_{-k,t} + \alpha} \tag{2.40}$$

where $n^j_{-k,t}$ is the number of allocation variables with value j in $c_{-k,t}$. We define $c_{-1,1} = \varnothing$, $c_{-k,1} = c_{1:k-1,1}$ and for $t > 1$

$$c_{-k,t} = \begin{cases} \{c_{1:t-1}, c_{1:k-1,t}\} & \text{if } k > 1 \\ c_{1:t-1} & \text{if } k = 1 \end{cases}$$

The main drawback of using such a model is that the number of targets K is assumed to be known and constant. The DP based model presented in the next subsection allows us to remove this restrictive assumption.

2.4.1 Dirichlet Process Based Model

To deal with an unknown number of targets K, we again take the limit of the finite model defined by (2.40) as $K \to \infty$. The conditional distributions of the allocation variables will be in accordance with the Chinese Restaurant or equivalently Polya Urn scheme.

$$\Pr(c_{k,t} = j | c_{-k,t}) = \begin{cases} \frac{n^j_{-k,t}}{\sum_i n^i_{-k,t} + \alpha} & \text{if } j \in \mathbf{c}_{-k,t} \\ \frac{\alpha}{\sum_i n^i_{-k,t} + \alpha} & \text{if } j \notin \mathbf{c}_{-k,t} \end{cases} \tag{2.41}$$

This means that the measurement $y_{k,t}$ is associated to an existing target $j \in c_{-k,t}$ with probability $\frac{n^j_{-k,t}}{\sum_j n^j_{-k,t} + \alpha}$ and to a new target (with some arbitrary label $j \notin c_{-k,t}$) with probability $\frac{\alpha}{\sum_j n^j_{-k,t} + \alpha}$. Note that the labeling of the targets is arbitrary here and can be chosen for programming convenience. The parameter α tunes the prior distribution of the number of targets. When α tends to 0, the number of expected targets will tend to 1, while when α tends to infinity it will tend to infinity. Note that although the prior probability is infinite dimensional, the number of detected targets is at most equal to the number of measurements. By using this nonparametric approach, the number of targets does not have to be specified a priori, but is estimated from the measurements.

To complete the model, we initialize a target state only when this target is created and assume that the initial state is distributed according to a distribution $\nu(\cdot)$.

We consider in the next section that the vector π_t is time-varying. In analogy to the static case

(2.41), we present a modified Polya Urn scheme so that the infinite dimensional vector π_t is integrated out.

2.4.2 Time-Varying DP

We introduce here a modified Polya urn scheme for the allocation variables. The evolution model (2.41) defines a Polya urn scheme based on the whole set of previous allocation variables from time 1 to the current time step. This evolution model may be modified by only conditioning the new allocation variable on the allocation variables sampled from time $t-r$ to time t. Consider the following r-order Markov model

$$\begin{aligned}\Pr(c_{k,t} = j|c_{-k,t}) &= \Pr(c_{k,t} = j|c_{-k,t-r:t}) \\ &= \begin{cases} \frac{n^j_{-k,t-r:t}}{\sum_i n^i_{-k,t-r:t}+\alpha} & \text{if } j \in \mathbf{c}_{-k,t-r:t} \\ \frac{\alpha}{\sum_i n^i_{-k,t-r:t}+\alpha} & \text{if } j \notin \mathbf{c}_{-k,t-r:t} \end{cases}\end{aligned} \tag{2.42}$$

where $n^j_{-k,t-r:t}$ is the number of allocation variables in $c_{-k,t-r:t}$ taking the value j. The allocation variables at time t are thus only dependent on previous allocation variables up to time $t-r$. Note that with this model, targets may disappear over time. At each time t, we forget allocation variables c_{t-r-1}. If a target that was observed at time $t-r-1$ has not been observed anymore from time $t-r$ to time $t-1$, then the target "disappear", i.e. it is not tracked anymore.

2.4.3 Clutter Model

In practice, it is important to be able to model the clutter noise. We assume that the clutter noise is uniformly distributed on the surveillance area. For a measurement $y_{k,t}$ from the clutter, we set $c_{k,t} = 0$ and assume that $\Pr(c_{k,t} = 0|c_{-k,t}) = 1-\lambda$. Under these assumptions, the generalized Polya urn is given by

$$\begin{aligned}\Pr(c_{k,t} = j|c_{-k,t}) &= \Pr(c_{k,t} = j|c_{-k,t-r:t}) \\ &= \begin{cases} (1-\lambda) & \text{if } j = 0, \\ \lambda\frac{n^j_{-k,t-r:t}}{\sum_i n^i_{-k,t-r:t}+\alpha} & \text{if } j \in c_{-k,t}, \\ \lambda\frac{\alpha}{\sum_i n^i_{-k,t-r:t}+\alpha} & \text{if } j \notin c_{-k,t} \cup \{0\}. \end{cases}\end{aligned}$$

2.5 Sequential Bayesian Inference

As the number of possible associations increases exponentially over time, inference in the models described before cannot be performed exactly and we need to approximate the posterior distributions of interest. We present a simple deterministic approximation algorithm in the linear Gaussian scenario and an SMC method to handle non-linear non-Gaussian cases.

2.5.1 Linear Gaussian Model

We consider here the important case where the evolution and observation equations (2.34)-(2.35) of each target are linear and Gaussian; that is $\nu(x) = \mathcal{N}(x; x_0, \Sigma_0)$ and

$$f\left(x' \middle| x_{t-1}\right) = \mathcal{N}(x'; Fx_{t-1}, \Sigma_v), \tag{2.43}$$

$$g(y|\, x_t) = \mathcal{N}(y; Hx_t, \Sigma_w). \tag{2.44}$$

The target distributions we are interested in estimating are given by

$$p(c_{1:t}, x_{1:t}|y_{1:t}) = p(c_{1:t}|y_{1:t})\, p(x_{1:t}|y_{1:t}, c_{1:t}).$$

In this case, conditionally on the allocation variables, the model is linear and Gaussian so the distribution $p(x_{1:t}|y_{1:t}, c_{1:t})$ is a Gaussian whose statistics can be computed using Kalman filtering techniques.

In particular, $p(x_t|y_{1:t}, c_{1:t}) = \mathcal{N}(x_t; x_{t|t}(c_{1:t}), \Sigma_{t|t}(c_{1:t}))$ where $x_{t|t}(c_{1:t})$ and $\Sigma_{t|t}(c_{1:t})$ are given by the Kalman filter. Thus we only need to approximate the discrete distribution of the allocation variables which satisfies the following recursion

$$p(c_{1:t}|y_{1:t}) = p(c_{1:t-1}|y_{1:t-1})\, \frac{p(y_t|y_{1:t-1}, c_{1:t})\, p(c_t|c_{1:t-1})}{p(y_t|y_{1:t-1})}$$

with $p(y_t|y_{1:t-1}, c_{1:t}) = \mathcal{N}(y_t; y_{t|t-1}(c_{1:t}), S_{t|t-1}(c_{1:t}))$ where ; $y_{t|t-1}(c_{1:t})$ and $S_{t|t-1}(c_{1:t})$ are respectively the predictive mean and covariance of the innovation which can be computed by KF updates.

We consider a simple deterministic algorithm to approximate this posterior distribution. At each time t, we consider all possible values for the new allocation variable, compute the unnormalized posterior distribution $p(c_{1:t}|y_{1:t})$ which is known exactly for each possible value and then prune to keep the N most likely trajectories. It can be interpreted as variant of

the Rao-Blackwellised Particle Filter where we substitute the sampling step with a full exploration of the possible values and the resampling step with a deterministic selection of the more likely particles. It has been shown that in most scenarios this simple algorithm outperforms the RBPF.

The algorithm proceeds as follows. To initialise the algorithm, we set $w^{(i)}_{m_0,0} = 1/N$ for $i \in \{1, ..., N\}$.

$N-$best algorithm for Multitarget Tracking
At time $t \geq 1$

- Set $w^{(i)}_{0,t} = w^{(i)}_{m_{t-1},t-1}$
- For $k = 1, \ldots, m_t$
  - For each particle $i = 1, .., N$ do
    - For $j \in c^{(i)}_{-k,t} \cup \{0\} \cup \{c_{new}\}$, let $\widetilde{c}^{(i,j)}_{k,t} = j$ and compute the weight

$$\widetilde{w}^{(i,j)}_{k,t} = w^{(i)}_{k-1,t} p\left(y_{k,t} \middle| y_{-k:t}, c^{(i)}_{-k,t}, \widetilde{c}^{(i,j)}_{k,t}\right) p\left(\widetilde{c}^{(i,j)}_{k,t} \middle| c^{(i)}_{-k,t}\right) \tag{2.45}$$

  - Keep the N particles $\left(c^{(i)}_{-k,t}, \widetilde{c}^{(i,j)}_{k,t}\right)$ with highest weights, rename them $c^{(i)}_{-(k+1),t}$ and denote $w^{(i)}_{k,t}$ the associated weights.

The target distribution $p(c_{1:t}|y_{1:t})$ is approximated using

$$\widehat{p}(c_{1:t}|y_{1:t}) = \sum_{i=1}^{N} W^{(i)}_t \delta_{c^{(i)}_t}(c_{1:t})$$

where $W^{(i)}_t \propto w^{(i)}_{m_t,t}$, $\sum_{i=1}^{N} W^{(i)}_t = 1$ whereas $p(x_t|y_{1:t})$ is approximated through

$$\widehat{p}(x_t|y_{1:t}) = \sum_{i=1}^{N} W^{(i)}_t \mathcal{N}\left(x_t; x_{t|t}\left(c^{(i)}_{1:t}\right), \Sigma_{t|t}\left(c^{(i)}_{1:t}\right)\right).$$

2.5.2 Non-Linear Non-Gaussian Model

In scenarios where the target dynamics and/or the measurement model are non-linear and/or non-Gaussian, it is typically impossible to compute $p(x_{1:t}|y_{1:t}, c_{1:t})$ in closed-form contrary to the linear Gaussian case. We present here briefly a SMC method to approximate the distributions $p(x_{1:t}|y_{1:t}, c_{1:t})$.

In this approach the target posterior distributions of interest are approximated by a cloud of random samples named particles which evolve over time using a combination of importance sampling and resampling steps. To design an efficient algorithm, it is necessary to design an efficient importance distribution. The expression of the optimal importance distribution is well-known. It is typically impossible or too computationally intensive to use the exact form but it can be approximated through the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF); see e.g. [11, 10, 21, 12] for details. We have chosen here to sample sequentially the measurements $y_{1,t}, ..., y_{m_t,t}$ one-at-a time. In this case, the optimal importance distribution for $c_{k,t}$ is given by

$$q(c_{k,t}|y_{k,t}, c_{-k,t}, x_{t-1}) \propto p(y_{k,t}|c_{k,t}, x_{c_{k,t},t-1})p(c_{k,t}|c_{-k,t})$$

where

$$p(y_{k,t}|c_{k,t}, x_{c_{k,t},t-1}) = \begin{cases} \frac{1}{V} & \text{if } c_{k,t} = 0 \\ \int g\left(y_{k,t}\middle| x_{c_{k,t},t}\right) f\left(x_{c_{k,t},t}\middle| x_{c_{k,t},t-1}\right) dx_{c_{k,t},t} & \text{if } c_{k,t} \in c_{-k,t} \\ \int g\left(y_{k,t}\middle| x_{c_{k,t},t}\right) \nu\left(x_{c_{k,t},t}\right) dx_{c_{k,t},t} & \text{if } c_{k,t} = c_{new}. \end{cases}$$

Where the clutter density is assumed uniform over the surveillance region, V is the volume of the surveillance region, $\nu(.)$ is the distribution of the initial state. For $c_{k,t} \in c_{-k,t} \cup \{c_{new}\}$, we can build an approximation of both $p(y_{k,t}|c_{k,t}, x_{c_{k,t},t-1})$ and $p(x_{k,t}|y_{k,t}, c_{-(k+1),t}, x_{-k,t})$ if $c_{k,t} \in c_{-k,t}$ or $p(x_{k,t}|y_{k,t}, c_{-(k+1),t})$ if $c_{k,t} = c_{new}$ using EKF or UKF. Then the SMC method proceeds as follows. To initialise the algorithm, we set $w_0^{(i)} = 1/N$ for $i \in \{1, ..., N\}$.

SMC Algorithm for Multitarget Tracking

At time $t \geq 1$

- For each particle $i = 1, .., N$ do
  - For $k = 1, \ldots, m_t$, sample $\widetilde{c}_{k,t}^{(i)} \sim q\left(c_{k,t}\middle| y_{k,t}, c_{-k,t}^{(i)}, x_{t-1}^{(i)}\right)$
  - For $j \in \mathbf{c}_{t-r:t-1}^{(i)}$, sample $\widetilde{\mathbf{x}}_{j,t}^{(i)} \sim q(\mathbf{x}_{j,t}|\mathbf{x}_{j,t-1}^{(i)}, \mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)})$
  - For $j \in \overline{\mathbf{c}_{t-r:t-1}^{(i)}} \cap \widetilde{\mathbf{c}}_t^{(i)}$, sample $\widetilde{\mathbf{x}}_{j,t}^{(i)} \sim q(\mathbf{x}_{j,t}|\mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)})$
- For $i = 1, .., N$, update the weights as follows

$$\begin{aligned} \widetilde{w}_t^{(i)} \propto \; & w_{t-1}^{(i)} \frac{\prod_{k=1}^n p(\mathbf{y}_{k,t}|\widetilde{c}_{k,t}^{(i)}, \widetilde{\mathbf{x}}_t^{(i)}) \prod_{k=1}^n \Pr(\widetilde{\mathbf{c}}_{k,t}^{(i)}|\widetilde{\mathbf{c}}_{1:k-1,t}^{(i)}, \mathbf{c}_{t-r:t-1}^{(i)})}{\prod_{k=1}^n q(\widetilde{c}_{k,t}^{(i)}|\mathbf{y}_{k,t}, \widetilde{\mathbf{c}}_{k-1,t}^{(i)}, \mathbf{c}_{t-r:t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)})} \\ & \times \frac{\prod_{j \in \mathbf{c}_{t-r:t-1}^{(i)}} p(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{x}_{j,t-1}^{(i)}) \prod_{j \in \overline{\mathbf{c}_{t-r:t-1}^{(i)}} \cap \widetilde{\mathbf{c}}_t^{(i)}} p_0(\widetilde{\mathbf{x}}_{j,t}^{(i)})}{\prod_{j \in \mathbf{c}_{t-r:t-1}^{(i)}} q(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{x}_{j,t-1}^{(i)}, \mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)}) \prod_{j \in \overline{\mathbf{c}_{t-r:t-1}^{(i)}} \cap \widetilde{\mathbf{c}}_t^{(i)}} q(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)})} \end{aligned} \tag{2.46}$$

with $\sum_{i=1}^{N} \widetilde{w}_t^{(i)} = 1$.

• Compute $N_{\text{eff}} = \left[\sum \left(\widetilde{w}_t^{(i)}\right)^2\right]^{-1}$. If $N_{\text{eff}} \leq N/2$, duplicate the particles with large weights and remove the particles with small weights, resulting in a new set of particles denoted $\dot{}_t^{(i)}$ with weights $w_t^{(i)} = 1/N$. Otherwise, rename the particles and weights by removing the $\widetilde{\dot{}}$.

Details of the algorithm is given in Appendix section.

2.6 Target Identification

In most of the target tracking applications, target identities are required to be maintained as new targets appear/disappear in the surveillance region. This section describes a simple user presentation logic which aims to keep a unique identity for each of the active targets in the region. In this approach, a target-to-target association is made between the targets/clusters of the particles having the highest weight at the current and the previous time instants. For this purpose a target-to-target assignment matrix is formed as follows.

Construction of the Assignment Matrix
At time t

• Pick the particle with the highest weight at time $t - 1$

• Extrapolate the state vector/sufficient statistics of the particle to the current time $x_{t-1} \rightarrow \hat{x}_t$.

• Pick the particle with the highest weight at time t

• Construct the Assignment Matrix as

Targets	**Targets at time t-1**				**No Target**			
at time t	**1**	**2**	...	$\mathbf{N_{t-1}}$	**1**	**2**	...	$\mathbf{N_t}$
1	$d_{1,1}$	$d_{1,2}$		$d_{1,N_{t-1}}$	CN	X	X	X
2	$d_{2,1}$	$d_{2,2}$		$d_{2,N_{t-1}}$	X	CN	X	X
$\vdots$	$\vdots$	$\vdots$			X	X	$\ddots$	X
$\mathbf{N_t}$	$d_{N_t,1}$	$d_{N_t,2}$		$d_{N_t,N_{t-1}}$	X	X	X	CN

Where $d_{i,j} = \tilde{x}_{i,j}^T Q \tilde{x}_{i,j}$, is the distance between the targets i and j. $\tilde{x}_{i,j} = \hat{x}_i - \hat{x}_j$, is the state prediction difference vector, and Q is any positive definite weighting matrix. CN is the cost of not assigning any target from the previous time to a target at current time. X refers to unallowable assignment. Posterior to the construction of the assignment matrix, assignment problem must be solved to find the optimal association. The assignment problem here is defined as follows. Given the elements $a_{i,j}$ of an n x m matrix, find the particular mapping

$$i \mapsto \Xi(i), \qquad 1 \le i \le n, \quad 1 \le \Xi(i) \le m,$$
$$i \ne j \quad \Rightarrow \quad \Xi(i) \ne \Xi(j)$$

such that the total cost function

$$C_{total} = \sum_{i=1}^{n} a_{i,\Xi(i)}$$

is minimized over all permutations Ξ. Various algorithms are proposed to solve the problem (see [5]). The auction algorithm is used to find the optimal associations in our simulations. Notice that, in order to minimize the cost, the assignment of target i to target j can occur only if $d_{i,j} < CN$. Therefore CN defines a gate on the defined distance between the targets. If this gate is exceeded, it is better to assign a new identity to the target rather than associating it to the existing ones.

2.7 Simulations

2.7.1 Linear Gaussian Model

Consider the following linear dynamic model

$$\mathbf{x}_{j,t+1} = F\mathbf{x}_{j,t} + G\mathbf{v}_{j,t}$$

$$\mathbf{z}_{k,t} = H\mathbf{x}_{j,t} + \mathbf{w}_{k,t}$$

where $\mathbf{x}_{j,t} = \begin{pmatrix} X_{j,t} \\ \dot{X}_{j,t} \\ Y_{j,t} \\ Y_{j,t} \end{pmatrix}$ with $(X_{j,t}, Y_{j,t})$ and $(\dot{X}_{j,t}, \dot{Y}_{j,t})$ being respectively the position and first derivatives of the target j at time t in 2-D coordinates.

$$F = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix}, G = \begin{pmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{pmatrix}, H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$\mathbf{v}_{j,t} \sim \mathcal{N}(0, Q)$ and $\mathbf{w}_{k,t} \sim \mathcal{N}(0, R)$ with $Q = \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix}$ and $R = \begin{pmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{pmatrix}$. T is the sampling time.

The following values are set: $T = 1$, $\sigma_v^2 = 10$, $\sigma_w^2 = 5000$, $r = 8$, $\alpha = 1$. The mean number of clutter measurements per time index is set to 5. 3 target tracks are simulated. The number of active targets (i.e. targets that produce measurements) at each time step is represented in Figure 2.10. The true target trajectory and the measurements, accumulated over time steps, are represented in Figure 2.7 and Figure 2.8. The deterministic filter is iterated with 1000 particles. Decision is made using the particle with the highest log-likelihood. The estimated target trajectories are represented in Figure 2.9 and the number of estimated targets in Figure 2.10. One track is considered as detected when it has produced at least 7 measurements in the last 8 steps. Once it has been detected, it is deleted when no measurements appear in the last 8 time steps.

For the same set of simulated data, a Joint Probabilistic Data Association (JPDA) filter and the Global Nearest Neighbourhood (GNN) algorithm were run. A track is initiated when 7

measurements are associated to this track over the last 8 time steps, and is deleted if it has not produced measurements over the last 5 time steps. The estimated target tracks of the algorithms are represented in Figures 2.11 and 2.13. The number of estimated targets are depicted in Figures 2.12 and 2.14.

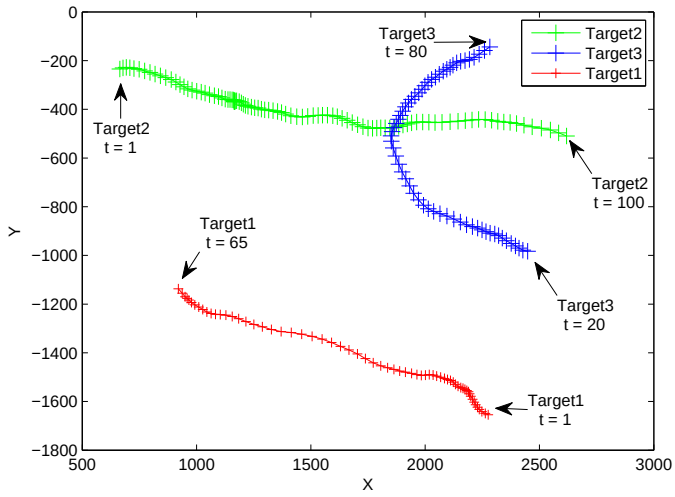


Figure 2.7: True Tracks.

2.7.2 Non-Linear Non-Gaussian Model

In the context of bearing only tracking, the target positions are not fully observable to the sensors, but a non-linear measurement equation exists which relates the state and the measurements non-trivially. We assumed a similar model to the linear gaussian case for the target dynamics. The resulting state space equations are as follows.

$$x_t = F_t x_{t-1} + v_t$$

$$y_t = tan^{-1}(\frac{y - s_y}{x - s_x}) + w_t$$

$$F_t = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix}, Q_t = 5 \times \begin{pmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{pmatrix}, R_t = \ 1 \times 10^{-4}$$

Where $T = 1$, Q_t and R_t are the process noise and measurement noise covariance matrices

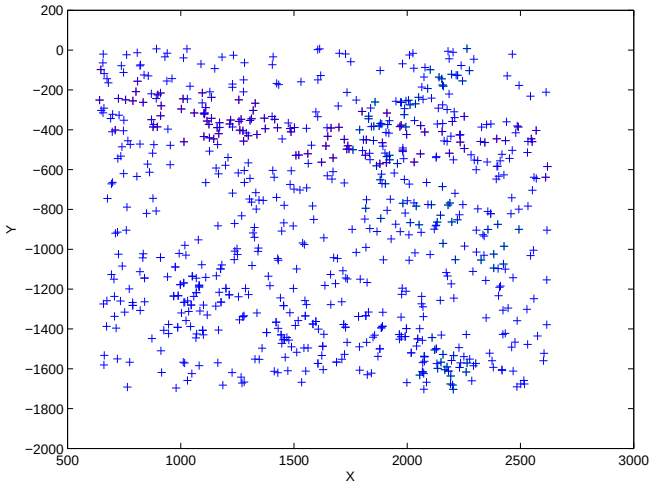


Figure 2.8: Measurements generated from time 1 to time T. Measurements are produced either by clutter or by a target.

respectively. In the scenario, there exist two targets whose true target trajectories are depicted in 2.15. There are two bearing only sensors, whose coordinates (s_x, s_y) are (-10000, 5000) and (10000, -5000). There are 5 false alarms per scan which are uniformly distributed over the region. The estimated target positions and estimated number of targets for the DP based algorithm is given in Figure 2.16 and 2.17 respectively.

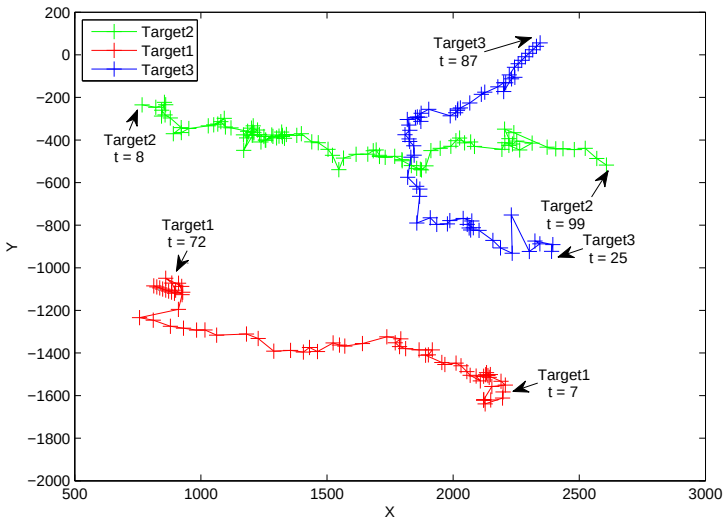


Figure 2.9: Estimated target trajectories using the DP based algorithm.

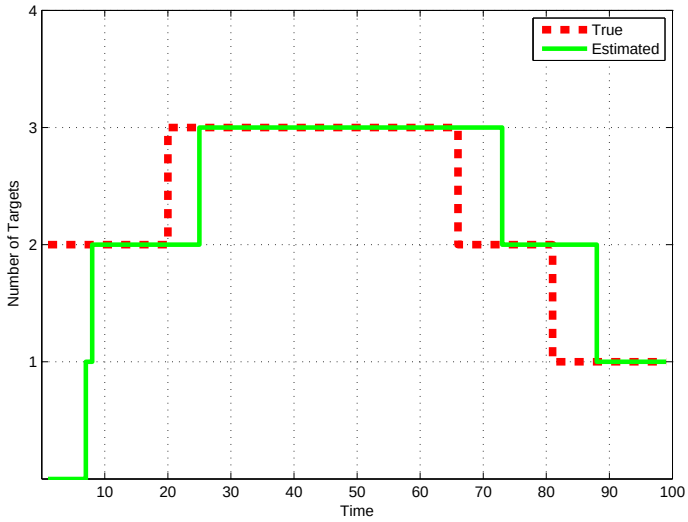


Figure 2.10: Estimated number of targets for DP based algorithm.

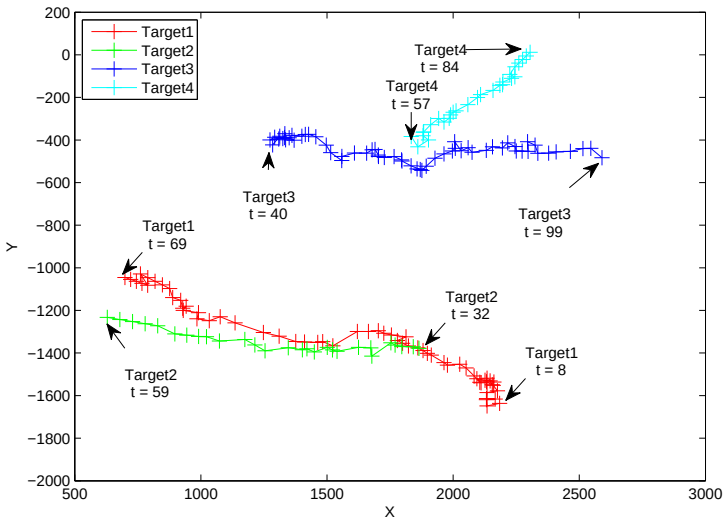


Figure 2.11: Estimated target trajectories using JPDA.

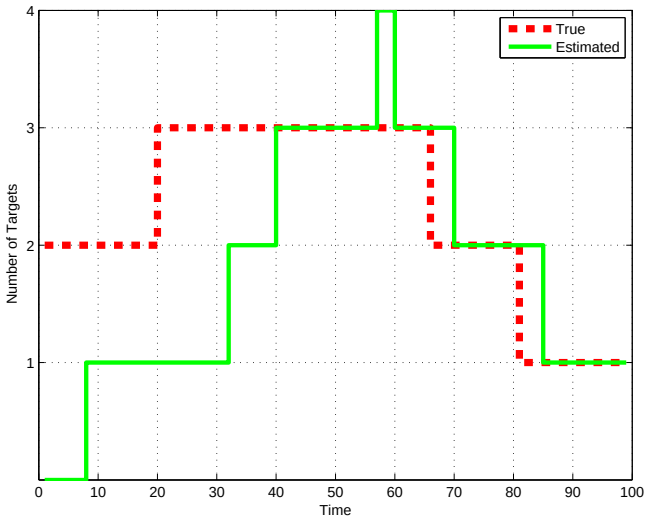


Figure 2.12: Estimated number of targets for JPDA

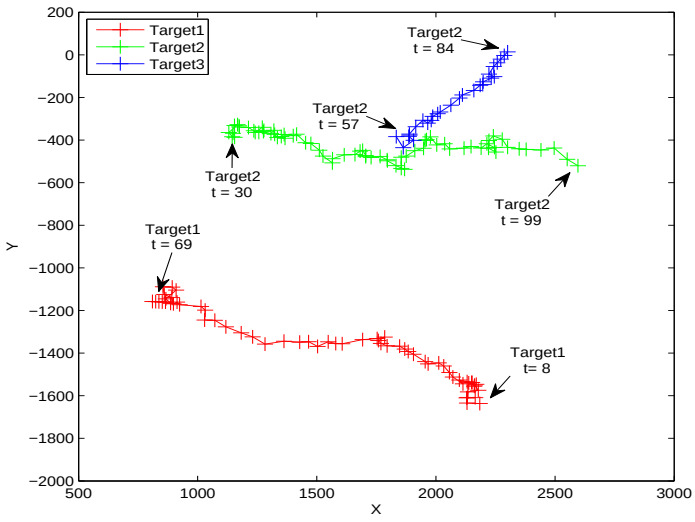


Figure 2.13: Estimated target trajectories using GNN.

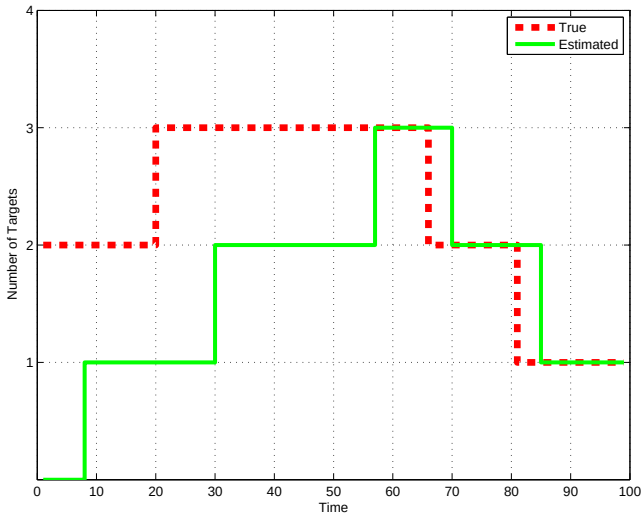


Figure 2.14: Estimated number of targets for GNN

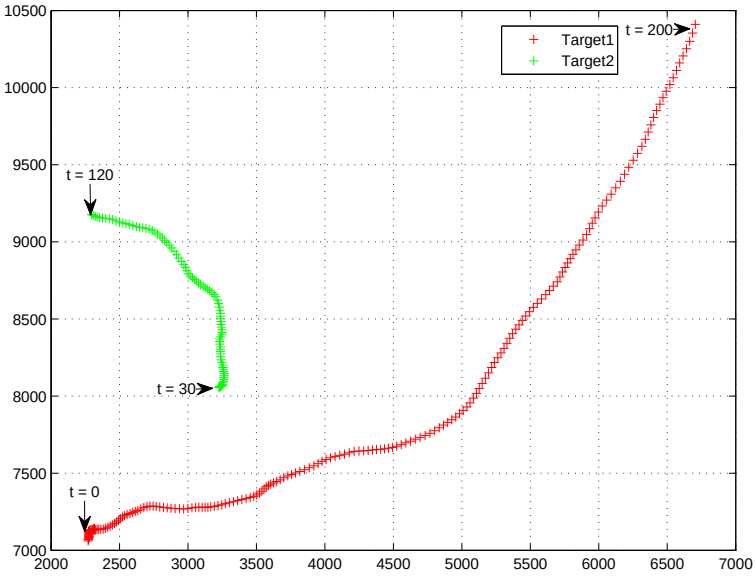


Figure 2.15: True Tracks for the bearing only tracking scenario.

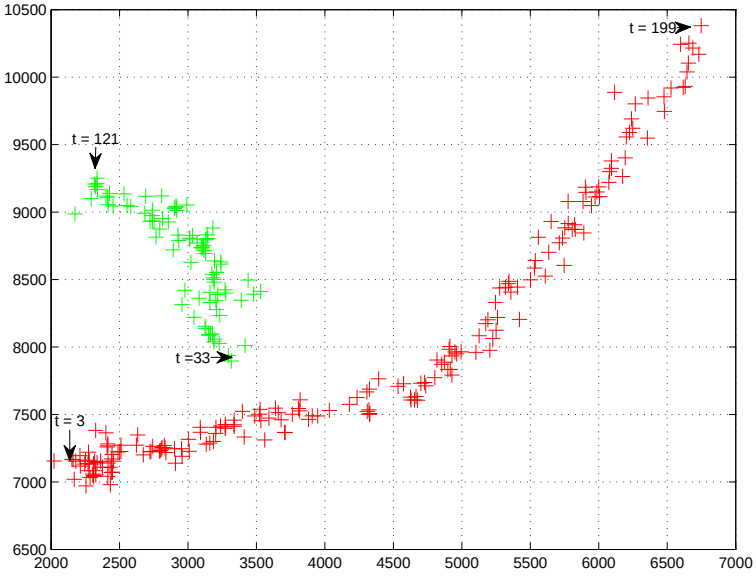


Figure 2.16: Estimated target trajetories using the DP based algorithm.

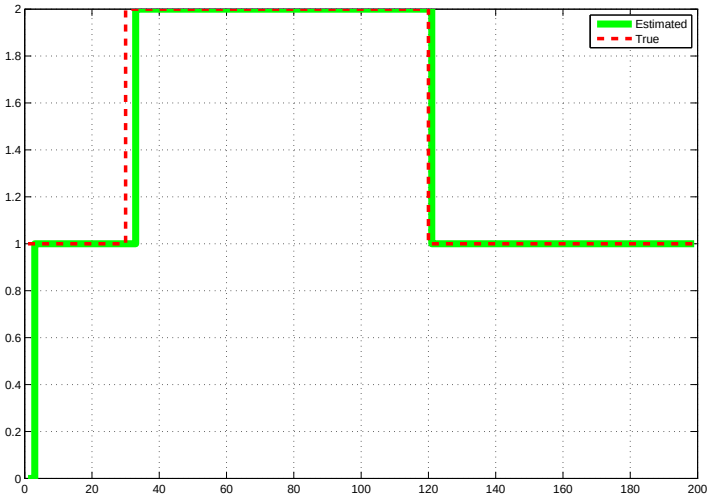


Figure 2.17: Estimated number of targets for DP based algorithm.

2.8 Conclusion

In this study we propose a new class of models for Bayesian multi-target tracking which is based on time varying extension of Dirichlet Process. The resulting algorithm is a complete tracking system which naturally handles the track initiation/deletion and target to measurement association tasks in a Bayesian framework by using time varying extension of Dirichlet processes as the priors. We also introduce a new output presentation logic for SMC based multi-target tracking algorithms which tracks the identity of the targets by combining the best hypothesis in consecutive time instants. The algorithm is tested on simulated data and its performance is compared with the conventional methods, namely JPDA and GNN algorithms which utilize standard (M/N) logic for track initiation/deletion tasks. In the simulations, the algorithm outperforms JPDA and GNN in heavy cluttered scenarios where JPDA and GNN algorithms fail to initiate/delete tracks properly. The proposed algorithm is a good combination of state of the art techniques and novel ideas such that it promises a high chance of showing good performance in different applications of signal processing.

CHAPTER 3

NONPARAMETRIC BAYESIAN FOR DYNAMIC SPEECH SPECTRUM REPRESENTATION AND TRACKING VARIABLE NUMBER OF VOCAL TRACT RESONANCE FREQUENCIES

3.1 Introduction

In this study, we extend our previous study on multi-target tracking to the formant tracking problem in speech signal processing literature. We propose a new approach for dynamic speech spectrum representation and tracking vocal tract resonance (VTR) frequencies. The method involves representing the spectral density of the speech signals as a mixture of Gaussians with unknown number of components. In the resulting representation, the number of formants is allowed to vary in time. Under the assumption of existence of varying number of formants in the spectrum, we propose the use of the DPM model based multi-target tracking algorithm for tracking unknown number of formants. The formant tracking problem is defined as a hierarchical Bayesian model and the inference is done using Rao-Blackwellized particle filter.

The air path of speech production mechanism (the vocal tract) is composed of various articulatory cavities (oral, pharyngeal, nasal, sinus etc.). Each cavity has particular natural frequencies at which the contained air naturally tends to vibrate. If the air inside the vocal tract is vibrated at natural frequencies, the vibrations are reinforced and the vocal tract resonates. That is, the vocal tract from glottis to lips acts as an acoustic resonator during speech production [15] [16]. The resonance frequencies, also known as formants, can be observed as the peaks of the magnitude spectrum of the speech signal.

Since formants are a rich source of information about uttered speech and the speaker, reliable formant estimation is critical for a wide range of applications, such as speech synthesis [15, 16, 39, 32, 31, 3], speech recognition [46, 18], voice conversion [40], vocal tract normalization [8, 41], measuring vocal tract length [45], accent classification [47] and speech enhancement [48].

The aim of our study is two-fold. First one is the dynamic representation of the speech spectrum as a Gaussian mixture with variable number of components. The second one is to track variable number of formant frequencies using the given representation. The representation of the speech spectrum with Gaussian mixture can also be seen in [50]. In [50], the parameters of the Gaussian mixture model are first estimated by Expectation Maximization (EM) algorithm then a reduction in the number of component stage is necessary while the order selection still remains as a problem. In our approach the number of mixture components is determined by the Dirichlet process naturally. We assume that the number of mixture components in the spectral density varies in time as the vocal tract resonance frequencies appear/disappear due to either unexcited sections of the vocal tract or abrupt changes in the spectrum during nasalization. In order to build a basis for our claim of varying number of formants, we investigate the effects of abrupt changes of the vocal tract in the state space framework. The analysis shows that the number of formants appearing in the spectrum may vary in time [49]. Therefore, we propose the use of a Dirichlet process based method which is flexible enough to allow varying number of mixture components in the estimated spectral density.

The following are the main contributions of this study.

- Dirichlet process based multi-target tracking algorithm is used for tracking variable number of formants where the inference is done via Rao-Blackwellized particle filters.

- The speech spectrogram is represented as a Gaussian mixture and the number of mixture components is determined dynamically.

The rest of this chapter is organized as follows. In Section 3.2, the definition of the formant tracking problem is introduced. Section 3.3 presents the details of the proposed method. The experimental results are given in Section 3.4 and finally we conclude the chapter with the Discussions and Conclusion sections.

3.2 Problem Definition

The problem of formant tracking involves the detection of the formants in a given speech spectrum. The formant frequencies are expected to appear in the regions of the spectrum where the energy is relatively high. Generally, the peaks of the spectrum are chosen to be the formant frequency candidates which are later eliminated according to their consistencies in time. However, the shape of the spectrum changes significantly during the open phase or the close phase of the glottis due to resonance anti-resonance cancelation. Therefore, a reliable estimation can not be done by simply detecting the peaks of the spectrum. A smoothing stage is necessary to minimize the variations in the peaks of the spectrum from frame to frame.

In our approach, the spectrum smoothing and the formant tracking tasks are done jointly. Dirichlet Process Mixture Model is used to represent the spectrum by an unknown number of Gaussians and the resulting mixture components are considered as the candidates to be the formant frequencies. We assume that the vocal tract resonance frequencies constituting the mixture components of the spectrum evolve according to a known state dynamics equation. Also the measurements originating from a specific formant are assumed to be produced according to a known measurement equation. Both of the equations are specified by the general state space representation of the VTR's which is given below.

$$f_{j,t} = h(f_{j,t-1}, v_t) \tag{3.1}$$

$$y_{j,t} = g(f_{j,t}, w_t) \tag{3.2}$$

Here $f_{j,t}$ denotes the state vector of the j^{th} formant at time t. $h(.)$ and $g(.)$ are possibly non-linear functions of the state. v_t and w_t are the process noise and the measurement noise sequences. A number of different models can be chosen to define the dynamics of the formant state and its relation with the measurements. These include simple linear Gaussian models as well as complex non-linear models. In the linear Gaussian model, the state dynamics evolve linearly and the noise sequences are assumed to be uncorrelated Gaussian white noise with known covariance matrices. Under these assumptions, Kalman filter produces the optimal state estimate (in the mean square sense). A standard choice of the model can be the constant velocity model given below.

$$\mathbf{f}_{j,t+1} = A\mathbf{f}_{j,t} + H\mathbf{v}_{j,t} \tag{3.3}$$

$$\mathbf{y}_{k,t} = C\mathbf{f}_{j,t} + \mathbf{w}_{k,t} \tag{3.4}$$

where $\mathbf{f}_{j,t} \triangleq \begin{pmatrix} f_{j,t} \\ \dot{f}_{j,t} \end{pmatrix}$ with $f_{j,t}$ and $\dot{f}_{j,t}$ being respectively the position and first derivatives of the formant j at time t.
$A = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}$, $H = \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix}$, $C = \begin{pmatrix} 1 & 0 \end{pmatrix}$, $\mathbf{v}_{j,t} \sim \mathcal{N}(0, \sigma_v^2)$, $\mathbf{w}_{k,t} \sim \mathcal{N}(0, \sigma_w^2)$ and T is the sampling time.

At each time step t, we have a set of measurements $y_{k,t}$, $k = 1 \ldots m_t$, which are acquired from the spectrum. Several alternatives are possible for measurement selection (see: Section 3.5). Assume that the measurements are distributed with the magnitude of the spectrum, i.e., they are the samples generated from the magnitude of the spectrum, see: Figure 3.1. In this case, the magnitude of the spectrum becomes the target density to be approximated as a mixture. In the model, the measurements are assumed to be originating from an unknown number of VTRs. The DPM model classifies the measurements into an unknown number of formants.

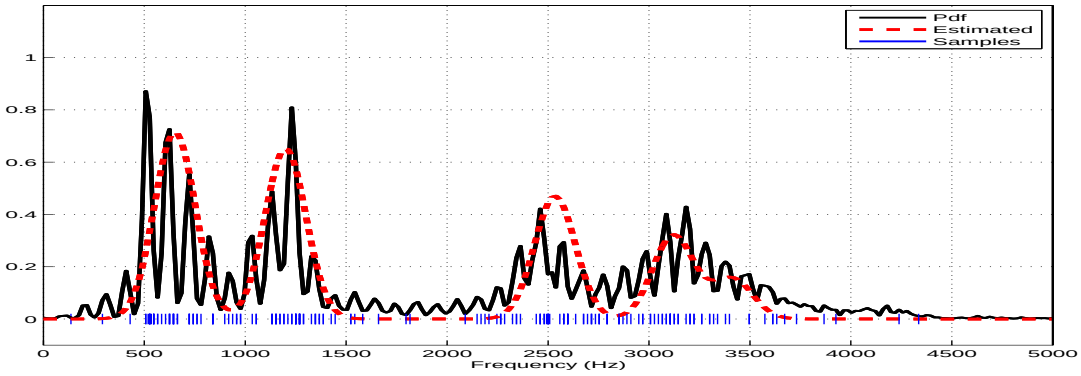


Figure 3.1: Measurement Extraction Procedure: 1-The DFT magnitude is normalized to be a probability density function (pdf)(solid line) . 2- Samples acquired from this pdf are considered as the measurements (shown on x-axis). 3- The DPM model classifies the measurements into an unknown number of formants the resulting representation is shown by the dashed line.

From the target tracking perspective, three joint problems have to be solved in order to be able to track the formant frequencies accurately. These sub-problems are:

- estimation of the number of VTRs,

- association of the measurements with the VTRs,
- estimation of the state vector of the VTRs.

Each sub-problem has to be solved as accurately as possible since the output of each step affects the performance of another.

3.2.1 Known Number of VTRs

Under the assumption of a prior knowledge of the number of VTRs, the tracking problem reduces to correctly associating each measurement y_k with one of the existing formants and estimating the state vector of each formant given the associated measurements. The structure will be formulated as a Hierarchical Bayesian Model and the Bayesian inference will be done by using the prior distributions defined in previous chapter. For ease of understanding, here we make the analysis for a single frame of the spectrum, and drop the time index t until the end of Section 3.2.

3.2.1.1 Model

Suppose there are K formants in the spectrum and our aim is to specify the origin of m measurements which are taken from a single frame. For this purpose we define the *label* c_k, $k = 1, ..., m$ for each measurement y_k, indicating the index of the associated formant. Labels take values from 1 to K and they are assumed to have a prior distribution which is multinomial with parameters $(m, \overline{\pi})$ (where $\overline{\pi} \triangleq [\pi_1, \ldots, \pi_K]$) such that:

$$p(c_k = j|\overline{\pi}) = \pi_j, \quad j = 1 \ldots K, \quad k = 1 \ldots m \tag{3.5}$$

π_j's are called the mixing coefficients or the weights which sum up to 1.

In order to identify the origin of each measurement among the K formants, the relation between the measurements and the formant states should be clearly defined. The distribution of the measurements originating from a specific formant is assumed to be known.

$$y_k|\theta, c_k \sim g(.|\theta_{c_k}) \tag{3.6}$$

where $\theta \triangleq [\theta_1, \ldots, \theta_K]$ is the stacked vector of the formant states and θ_{c_k} stands for the state vector of the c_k^{th} formant. It is assumed that the initial state vectors of the formants are distributed according to the base distribution denoted by $G_0(.)$.

$$\theta_{c_k} \sim G_0(.). \tag{3.7}$$

As an example, $G_0(.)$ can be a Gaussian distribution with a mean equal to the center of the spectrum and a covariance which is sufficiently large to cover the whole spectrum. Another choice can be a mixture of Gaussians, in which the mixture components are centered around the nominal values of the formants. Given the definitions above, the resulting Hierarchical Model becomes

$$\begin{aligned} \pi|\alpha, K &\sim Dirichlet(\frac{\alpha}{K}), && (3.8)\\ c_k|\pi &\sim Multinomial(\pi), && (3.9)\\ \theta_j|G_0 &\sim G_0(.), && (3.10)\\ y_k|\theta, c_k &\sim g(.|\theta_{c_k}). && (3.11) \end{aligned}$$

The conditional probability $p(c_m = j|c_1, \ldots, c_{m-1})$, i.e. the prediction probability of the allocation variable for the future observations is (see: (2.21))

$$p(c_m = j|c_{-m}, \alpha, K) = \frac{n_j + \frac{\alpha}{K}}{m - 1 + \alpha} \tag{3.12}$$

where n_j indicates the number of measurements previously assigned to the formant j.

3.2.2 Extension to infinite dimensional priors

A standard approach for deriving the infinite dimensional priors for the model is letting the number of formants K go to infinity [34]. It is important to notice that, although we try to define a prior on infinite number of components, only a finite number of these can be observed given finite number of measurements. The resulting model makes the number of formants K now variable which will be updated according to the incoming measurements. This flexibility enables the model to adapt unknown number of components. A simple calculation shows that, in the limiting case, the resulting prediction density for the future observation is equal to (see: (2.22),(2.23))

$$\begin{aligned} p(c_m = j|c_{-m}, \alpha) &= \frac{n_j}{m - 1 + \alpha} && (3.13)\\ p(c_m = c_{new}|c_{-m}, \alpha) &= \frac{\alpha}{m - 1 + \alpha} && (3.14) \end{aligned}$$

That is, each measurement is either a member of one of the existing formants with probability proportional to the number of measurements previously assigned to that formant, or it is one of the unobserved infinite number of formants (i.e., new formant) with probability proportional to α. The equations above also define the probabilities to sample from the famous Chinese Restaurant Process and Polya Urn scheme which are introduced in the previous chapter.

Returning back to our hierarchical model, for the infinite dimensional case, the model can be specified by the following equations.

$$G|G_0, \alpha \sim DP(G_0, \alpha) \tag{3.15}$$

$$\theta_k|G \sim G(.) \tag{3.16}$$

$$y_k|\theta_k \sim g(.|\theta_k). \tag{3.17}$$

Conditional on the previous values of the association variables, the predictive distribution of a new association variable can be written as,

$$(c_{n+1}|c_1, \ldots, c_n, G_0, \alpha) \sim \frac{\alpha}{\alpha + n}\delta_{m+1}(.) + \frac{1}{\alpha + n}\sum_{i=1}^{n}\delta_i(.) \tag{3.18}$$

where, the association variables take positive integer values in increasing order, and m is the number of distinct values taken by the previous association variables. These probabilities define the prior distribution of associating a new observation to the existing tracks or a new track. The posterior distribution will later be calculated by taking the measurement likelihood given the association into account.

3.3 Method

For the given model the inference is done via sequential Monte Carlo, aka. particle filter, which approximates the joint density of the formant state vectors and the association variables in a Rao-Blackwellised fashion. The density to be approximated can be decomposed into linear and non-linear states.

$$p(f_{1:t}, c_{1:t}|y_{1:t}) = p(f_{1:t}|y_{1:t}, c_{1:t})\, p(c_{1:t}|y_{1:t}) \tag{3.19}$$

The formant states evolve linearly according to the state dynamic equation (3.1), and the association variables depend on the past values in accordance with Dirichlet process. Rao-Blackwell idea make use of the fact that, conditional on the association variables, the sufficient

statistics of the formant states can be computed by using Kalman filtering technique. On the other hand, the marginal density of the association variables can be approximated by point masses, i.e., particles with their weights.

$$\widehat{p}(c_{1:t}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{c_{1:t}^{(i)}}(c_{1:t}) \tag{3.20}$$

where w_t^i and $c_{1:t}^i$ are functions of $y_{1:t}$. From the target tracking point of view, each particle defined as above corresponds to a hypothesis on the associations made between the formants and the measurements. Different hypotheses may conclude existence of different number of formants in the spectrum. Each particle, keeps the formant state vectors with sufficient statistics and the past values of the association variables. In the standard approach, the particle filter update is done as follows.

Suppose we have N particles with their states and weights at time/frame $t-1$.

$$\{c_{1:t-1}^{(i)}, \quad \hat{f}_{t-1|t-1,1}^{(i)} \cdots \hat{f}_{t-1|t-1,M_i}^{(i)}, \quad \Sigma_{t-1|t-1,1}^{(i)} \cdots \Sigma_{t-1|t-1,M_i}^{(i)}\}_{i=1}^{N}, \qquad \{w_{t-1}^{(i)}\}_{i=1}^{N}$$

Here M_i represents the number of formants of the i^{th} particle. f and Σ denote the state vector and the covariance matrix of the formants respectively. At time t we have a set of m_t measurements which are actually sampled from the magnitude of the spectrum. Prior to the measurement update, the sufficient statistics of each particle must be extrapolated to time t before processing the measurements of the frame t.

Step1: Prediction update of the particles.

- for i=1:N
  - compute $\{\hat{f}_{t|t-1,1}^{(i)} \cdots \hat{f}_{t|t-1,M_i}^{(i)}, \quad \Sigma_{t|t-1,1}^{(i)} \cdots \Sigma_{t|t-1,M_i}^{(i)}\}$

Once the prediction update for the sufficient statistics is complete, we proceed with the measurement update. In a generic Rao-Blackwellised Particle Filter (RBPF) the next steps are the sampling for the non-linear states (association variables), weight update and the resampling stages respectively. These steps are to be repeated for all the measurements of the frame t.

Step2: Measurement update.

- for k=1:m_t

- for i=1:N
  - sample from $q(c_{k,t}|c^{(i)}_{-k,t}, y_{k,t}, y_{-k,t})$.
  - update the weights as $\tilde{w}^{(i)}_{k,t} \propto w^{(i)}_{k-1,t} \frac{p\left(y_{k,t}|y_{-k,t}, c^{(i)}_{k,t}, c^{(i)}_{-k,t}\right) p\left(c^{(i)}_{k,t}|c^{(i)}_{-k,t}\right)}{q(c_{k,t}|c^{(i)}_{-k,t}, y_{k,t}, y_{-k,t})}$.
- Normalize weights $w^{(i)}_{k,t} = \frac{\tilde{w}^{(i)}_{k,t}}{\sum_{j=1}^{N} \tilde{w}^{(j)}_{k,t}}$.
- Calculate effective sample size, $N_{eff} = \frac{1}{\sum_{j=1}^{N} (w^{(j)}_{k,t})^2}$.
- Resample if $N_{eff} < N_{Threshold}$

where,

$$w_{0,t} \triangleq w_{m_{t-1},t-1}, \quad y_{-k,t} \triangleq \bigcup_{i=1}^{t-1} \{y_{1:m_i,i}\} \bigcup \{y_{1:k-1,t}\}, \quad c_{-k,t} \triangleq \bigcup_{i=1}^{t-1} \{c_{1:m_i,i}\} \bigcup \{c_{1:k-1,t}\},$$

and $q(c_{k,t}|c^{(i)}_{-k,t}, y_{k,t}, y_{-k,t})$ is the importance distribution for the association variables.

It is possible to sample from the optimal importance distribution $q(c_{k,t}|c_{-k,t}, y_{k,t}, y_{-k,t}) = p(c_{k,t}|c_{-k,t}, y_{-k,t}, y_{k,t})$. By applying Bayes rule, one can factorize the optimal distribution into

$$p(c_{k,t}|c_{-k,t}, y_{-k,t}, y_{k,t}) \propto p(y_{k,t}|c_{k,t}, c_{-k,t}, y_{-k,t}) \times p(c_{k,t}|c_{-k,t}). \tag{3.21}$$

Prior is calculated according to Polya Urn (see equations (3.13) and (3.14)), and the likelihood is:

$$p(y_{k,t}|c_{k,t}, c_{-k,t}, y_{-k,t}) = \mathcal{N}(g(\mu^{(i)}_{c_{k,t},t}), S^{(i)}_{c_{k,t},t}) \tag{3.22}$$

where $\mu^{(i)}_{c_{k,t},t}, S^{(i)}_{c_{k,t},t}$ are the mean and the innovation covariance calculated by Kalman Filter (KF) using the previous measurements assigned to that formant. If $c_{k,t}$ corresponds to a new formant, the likelihood is calculated by initiating a KF with mean x_0 and covariance P_0. More specifically one can write,

$$\begin{aligned} \text{if} \quad & c_{k,t} = c_{new}, \\ & p(y_{k,t}|c_{k,t}, c_{-k,t}, y_{-k,t}) = \mathcal{N}(CAx_0, [C(AP_0A^T + Q)C^T + R]) \end{aligned} \tag{3.23}$$

After $c_{k,t}$ is sampled, the KF of the corresponding formant is immediately updated by using the measurement $y_{k,t}$.

It is important to notice that at the sampling stage, each measurement is either associated with one of the existing formants or it is considered to be originating from a new formant. Therefore, the corresponding association variable can only take a finite number of values which

implies that it is possible to generate all possible hypotheses. The sampling stage can be replaced by a full exploration of the support of $p(c_{k,t}|c_{-k,t})$ and N-best particles having the highest weights can be used to represent the best hypotheses. In that case, the measurement update step of the algorithm is modified as follows.

Step2: Measurement update for the modified algorithm.

- for k=1:m_t
  - for i=1:N
    - for each possible value of $c_{k,t} = j$
      - Calculate the weights $w_{k,t}^{(i)} = w_{k-1,t}^{(i)} p(y_{k,t}|c_{k,t}, c_{-k,t}, y_{-k,t}) p(c_{k,t}|c_{-k,t})$.
  - Keep N-best particles with the highest weights.

3.3.1 r-order Markov model

It is possible to model the association priors as an r-order Markov Chain in which the association priors are conditioned on the previous associations made within the last r frames [42]. Markov model implies,

$$p(c_{k,t}|c_{-k,t}) = p(c_{k,t}|\tilde{c}_{-k,t}) \tag{3.24}$$

$$\text{where,} \quad c_{-k,t} \triangleq \bigcup_{i=1}^{t-1} \{c_{1:m_i,i}\} \bigcup \{c_{1:k-1,t}\}, \quad \tilde{c}_{-k,t} \triangleq \bigcup_{i=t-r}^{t-1} \{c_{1:m_i,i}\} \bigcup \{c_{1:k-1,t}\}.$$

In this scheme, the association variables before time $t-r$ are forgotten. Therefore, if a formant is not associated with any of the measurements between time $t-r$ and t, then it is deleted. Consequently, the formants being tracked are allowed to disappear enabling the algorithm to track time-varying number of formants.

3.3.2 Output Presentation

Posterior to the measurement update, formant states density can be approximated by using the particles, their weights and the sufficient statistics as follows.

$$\widehat{p}\left(\bar{f}_t \middle| y_{1:t}\right) = \sum_{i=1}^{N} \sum_{j=1}^{M_i} w_t^{(i)} \mathcal{N}(\hat{f}_{t|t,j}^{(i)}, \Sigma_{t|t,j}^{(i)}) \tag{3.25}$$

where M_i represents the number of formants of the i^{th} particle and $\bar{f}_t$ is the vector of all formants. Considering the similarities of the algorithm with Multiple Hypothesis Tracker (MHT), it is also possible to use the particle with the highest weight (best hypothesis) and its estimate of the formant states for output presentation. As mentioned earlier, among all the clusters that represent the speech spectrum, only the time-consistent ones are declared to be the formants. Therefore a consistency check has to be applied to the formant candidates. For this purpose, an association should be made between the formants of the particles having the highest weight in consecutive frames. In order to associate the existing formants with the new ones properly, the distance between the new and old formants are calculated and the association is done by using the Auction algorithm (see: Section 2.6). The formants are included in the output presentation only if their track life is long enough and sufficiently large number of measurements are associated with them. In our experiments only the formant candidates which last over at least 11 frames and which are associated to at least 10% of the incoming measurements on average are declared as the formants and shown at the output.

3.4 Experimental Results

In this section, we illustrate the performance of the algorithm on examples chosen from TIMIT database. These examples are representative of our extensive tests performed on many sentences in the database. The examples have been selected to clearly illustrate the variation of number of formants during speech utterance. In order to provide a general understanding of the performance of the algorithm in different speech utterances of different individuals, the algorithm parameters are kept fixed (except for the larger measurement noise variance used for female utterances to increase the formant bandwidth) for all the utterances. Moreover, the output of the algorithm is depicted together with the WaveSurfer output and hand labeled formants [9]. Additionally, we compare the output of our algorithm with the Wavesurfer, by calculating the distance with the hand-labeled data in the vowel-like regions of the utterances, as those regions are considered to be the only regions where a reasonable comparison can be made between a variable number of formants tracker and a standard one. Unfortunately all the examples can not be included within this chapter but they are available online. All the results, including the larger versions of the figures given in this section are available at "http://www.eee.metu.edu.tr/~ozkan/DBFTFigures.pdf".

For all the given utterances, standard linear Gaussian constant velocity model (see: Section 3.2) is used as the formant dynamics. The sampling frequency is chosen to be 10KHz. A pre-emphasis filter is used to reduce the spectral tilt between the formant frequencies. After the pre-emphasis stage, speech signal is divided into frames using Hamming window. The frame length and frame rate are 40 msecs and 10 msecs respectively. Magnitude spectrum of each frame is found by using 512 points DFT. At most 100 measurements per frame is fed to the algorithm. Number of measurements per frame is kept proportional to the energy of the frame. Low energy regions of the spectrum are truncated to zero. The concentration variable α is chosen to be 0.2, and the order of the Markov model r is chosen to be 5. Base distribution G_0 is chosen to be a Gaussian distribution centered at 2.0 KHz with a covariance sufficiently large to cover the whole spectrum. 100-best particles are kept at each update and output presentation is done by using the particles having the highest weight. Only the formant candidates which last over at least 11 frames and which are associated to at least 10% of the incoming measurements on the average are declared as the formants and shown as the output. The LPC order of the WaveSurfer algorithm is chosen as 12 and 14 while tracking four formants and five formants respectively.

In calculating the distance with the hand-labeled data in vowel-like regions, average absolute error is computed by using the following formula.

$$E_j = \frac{1}{N_c} \sum_{i=1}^{N_c} |\hat{F_i}^j - F_i^j| \tag{3.26}$$

where E_j is the average absolute error of the j^{th} formant, $\hat{F_i}^j$ and F_i^j are the estimated and hand labeled formant trajectories of the j^{th} formant at frame i respectively and N_c is the number of frames in which the error is calculated. As mentioned earlier, the average absolute error is calculated only for the vowel-like regions of the utterances for the first three formants. In order to include effects of missed formants we define the coverage ratio of each formant by simply computing the ratio of the length of the formant trajectory produced by Dirichlet Based Formant Tracker (DBFT) and the whole length of the section in which the average error is calculated. In the given figures, following vectors are defined in order to provide information

about the tracking performance of the algorithms in the vowel-like regions as follows.

$$E_{DBFT} \triangleq [E_1 E_2 E_3] \tag{3.27}$$

where E_i's, $i = 1, .., 3$, are calculated by using the DBFT formant trajectories.

$$E_{WS} \triangleq [E_1 E_2 E_3] \tag{3.28}$$

where E_i's, $i = 1, .., 3$, are calculated by using the WaveSurfer formant trajectories.

E_{DBFT} and E_{WS} are the average absolute error vectors for the estimated formants of DBFT and the Wavesurfer respectively in vowel-like regions of the whole utterance. C_{DBFT} is the stacked vector holding the coverage ratio of the formants of DBFT in vowel-like regions of the whole utterance.

Firstly, an example utterance is given in which the change in the number of formants can be clearly observed. In Figure 3.2, the formant trajectories of DBFT ('dotted line'), WaveSurfer ('solid line') and the hand labeled database ('dashed line') are superimposed on the spectrum of the utterance 'his head flopped back' taken from the TIMIT database. At the beginning of the sentence, both the proposed Dirichlet Based Formant Tracker (DBFT) and WaveSurfer successfully detect and track the four formants until the start of the fricative sound —z—. Between 0.25s and 0.35s, DBFT drops two low frequency formants and initiates a new formant located around 4.5KHz whereas WaveSurfer misses the high frequency formant as it tends to preserve the continuity of the previously initiated formants. After the sound —z—, in DBFT output, the disappearing formants reappear and the high frequency formant disappears in the spectrum. During the interval [0.5s-0.65s], DBFT drops the low frequency formant and tracks the three high frequency formants. Between 0.64s and 0.77s DBFT tracks three formants and misses the formant located around 2.5KHz which is not as clear. During the closure part of the plosive sound —b— (time [0.77s-0.91s]), no formant exists in the spectrum due to complete constriction of the vocal tract. In this region DBFT tracks no formants, whereas WaveSurfer produces distorted formant trajectories as it tries to track non-existing four formants. During [0.91s-1.1s], five formants are tracked for the vowel —ae— by DBFT, on the other hand WaveSurfer misses one of the high frequency formants as it tracks fixed number of four formants. At time 1.1s, two of the formants tend to merge in the spectrum, which is known as the 'velar pitch' phenomenon that occurs before the velar consonant —k—. Both WaveSurfer and DBFT manage to track the changes successfully for these formants. Posterior to the closure part of the consonant —k—, DBFT initiates a new formant around 2KHz which

is the right location coinciding with the formant loci. In vowel-like regions of the utterance, formant trajectories found by both algorithms coincide with the hand labeled trajectories.

For this speech utterance, the coverage ratio of DBFT is 95%, 99% and 80% for the first three formants respectively. The coverage ratio of the third formant is lower than the first two due to the missed track between 0.64s and 0.77s mentioned above. The average absolute error of DBFT, is 103Hz, 68Hz and 130Hz for the first three formants respectively. For the same utterance average absolute error of WaveSurfer is 99Hz, 54Hz and 122Hz. Considering the fact that WaveSurfer is known to be a good formant tracker in vowel-like regions (see: [9], [35]), the performance of the DBFT algorithm seems satisfactory. Average absolute error vectors of DBFT and Wavesurfer are given with the coverage ratio of DBFT below each figure in order to provide an information about the tracking performance of the algorithm for a given speech utterance.

Figure 3.3 is the speech utterance 'Where were you while we were away' which is composed of all voiced sounds. This example differs from the previous one as it includes long formant trajectories. The proposed method manages to track the formant trajectories continuously in most of the regions. WaveSurfer tracks four formants accurately but misses the fifth one as it tries to follow only four formants.

Figure 3.4 is the output of the algorithm for the speech utterance 'Books are for schnooks' from the TIMIT database. The spectrogram representation of this utterance is given in Figure 3.10.

3.4.1 Order Selection Problem

One of the main difficulties encountered by a fixed number of formants tracker is to make a decision on the number of formants that are expected to exist in the spectrum in advance. This problem can be examined clearly in Figures 3.5 and 3.6. In Figure 3.5, the number of formants to be tracked by the WaveSurfer is fixed to 4. At the beginning of the utterance there exist 5 formants which causes the Wavesurfer algorithm to switch the formant trajectories among each other. More specifically, one can observe that at 0.5s the estimated formant trajectories around 2.5KHz and 3.0KHz make a jump to the formants at higher frequencies hence the third formant is missed during 0.5 and 0.7s. In Figure 3.6 the number of formants

to be tracked by the WaveSurfer is fixed to 5. In this case the algorithm produces a redundant track (second formant) at the beginning of the utterance which causes errors (abrupt jumps and track changes) during the whole utterance. The output of DBFT is also depicted on the same figures 'dotted line'. The algorithm successfully tracks the formants in both cases as it is capable of determining the number of formants automatically and dynamically, which is claimed to be the main advantage of our approach in tracking formants.

3.4.2 Nasal Sounds

In nasal sounds, the air path of speech production mechanism includes the nasal cavity therefore the formants structure is changed. The resonance frequencies of the nasal cavity are different from the oral cavity and this fact causes difficulties when the formant tracks are considered to be fixed and continuous. The zoomed sections of Figure 3.7 that are given in Figure 3.8 are the examples of the mentioned cases. As can be seen from these figures, Wavesurfer looses the trajectory of the second formant because of the extra formant produced during the nasal sound. On the other hand the DBFT manages to follow the trajectories given by the hand labeled data correctly, and it initiates and deletes the extra formant successfully.

3.4.3 Spectrogram Representation

Figure 3.9 and 3.10 show a comparison of the original magnitude spectrogram and the estimated spectrogram. In a single frame, proposed method represents the noisy DFT as a mixture of Gaussians as shown in Figure 3.11. The number of Gaussians in the mixture is not fixed and it changes dynamically in time. As can be seen from the figures, the resulting estimation represents the spectrum with a good performance. It is also worth mentioning that the spectrogram is represented as an analytical expression which can later be used as an approximation of the actual one in further processing of the signal, like synthesis, recognition, coding, etc.

[1]Definitions for E_{DBFT}, E_{WS} and C_{DBFT} are given in equations (3.27), (3.28) and the paragraph below.

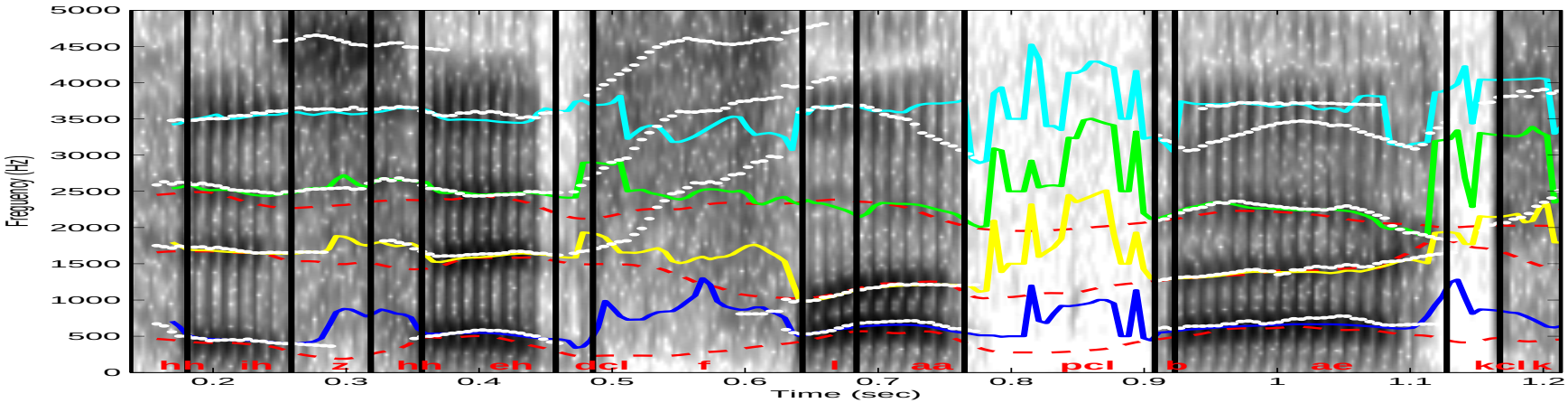


Figure 3.2: DBFT's (white dotted line), hand-labeled (red dashed line) and WaveSurfer's (solid line) formant trajectories superimposed on the spectrogram of the utterance "His head flopped back"

(TIMIT\Train\dr6\mabc0\SI1620.WAV) from TIMIT database. E_{DBFT}=[103 68 130], E_{WS}=[99 54 122] and C_{DBFT}=[95 99 80][1]

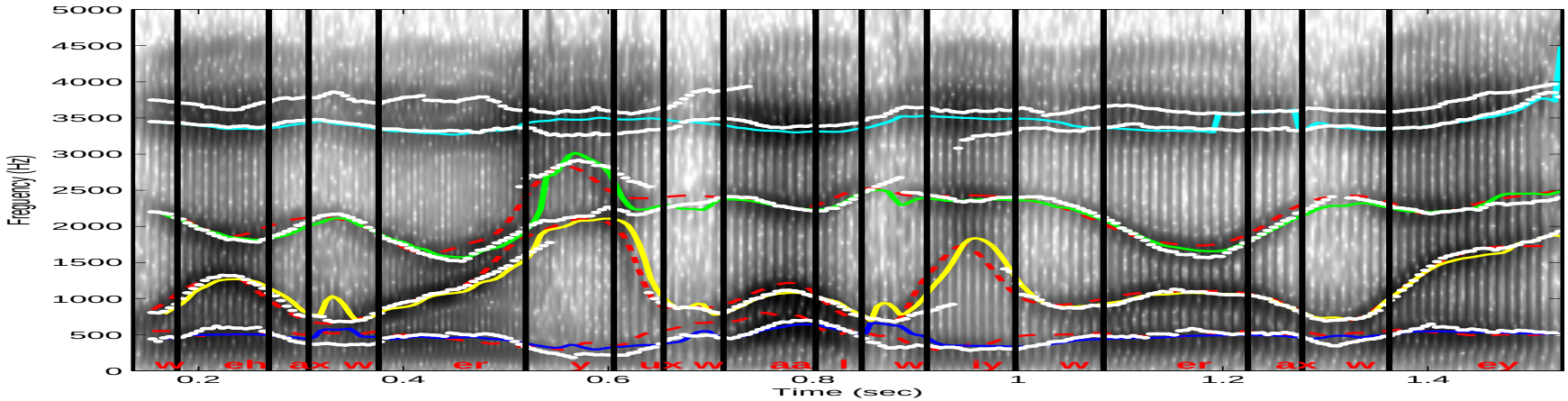


Figure 3.3: DBFT's (white dotted line), hand-labeled (red dashed line) and WaveSurfer's (solid line) formant trajectories superimposed on the spectrogram of the utterance "Where were you while we were away?"
(TIMIT\Test\dr8\mjln0\SX9.WAV) from TIMIT database. E_{DBFT}=[89 73 110], E_{WS}=[62 65 73] and C_{DBFT}=[100 90 99][1]

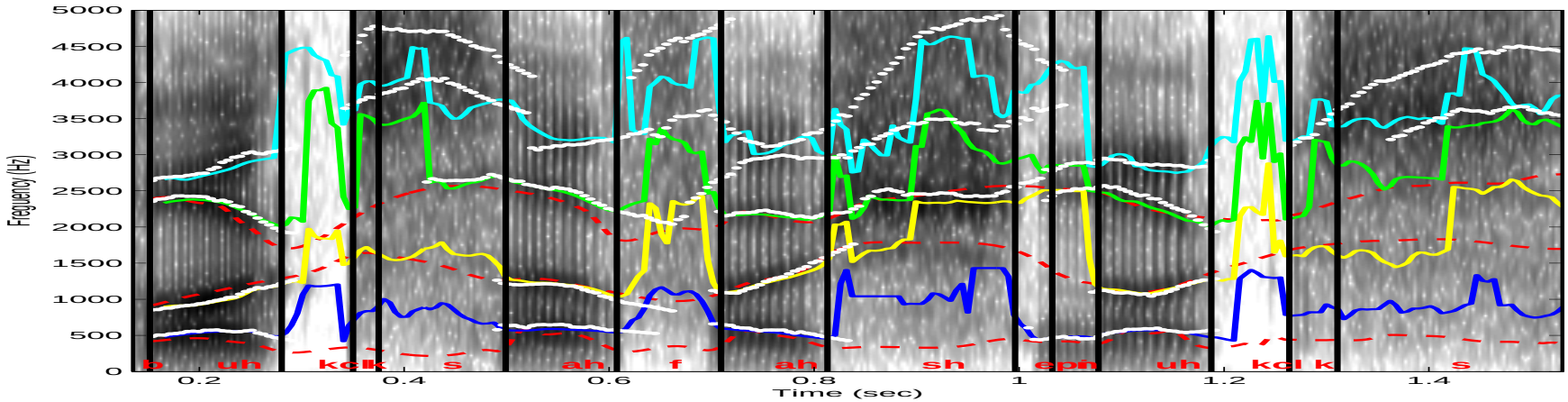


Figure 3.4: DBFT's (white dotted line), hand-labeled (red dashed line) and WaveSurfer's (solid line) formant trajectories superimposed on the spectrogram of the utterance "Books are for schnooks"

(TIMIT\Test\dr1\mwbt0\SI2183.WAV) from TIMIT database. E_{DBFT}=[109 63 120], E_{WS}=[85 46 77] and C_{DBFT}=[100 100 100][1]

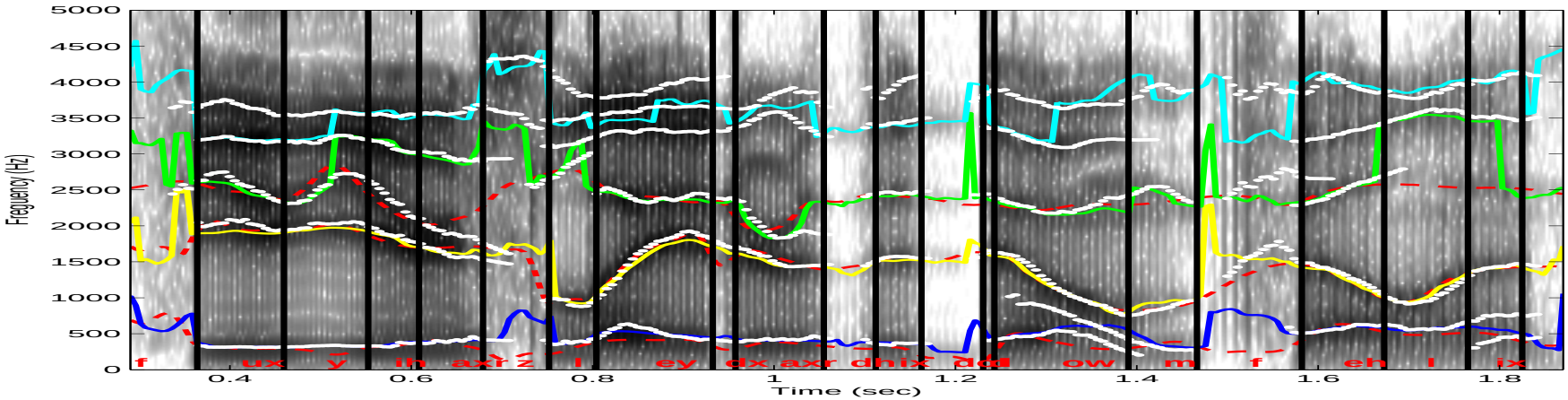


Figure 3.5: DBFT's (white dotted line), hand-labeled (red dashed line) and WaveSurfer's (solid line) formant trajectories superimposed on the spectrogram of the utterance "A few years later the dome fell in" (TIMIT\Test\dr2\mwew0\SI731.WAV) from TIMIT database. E_{DBFT}=[80 55 114], E_{WS}=[78 43 299] and C_{DBFT}=[97 97 85][1]

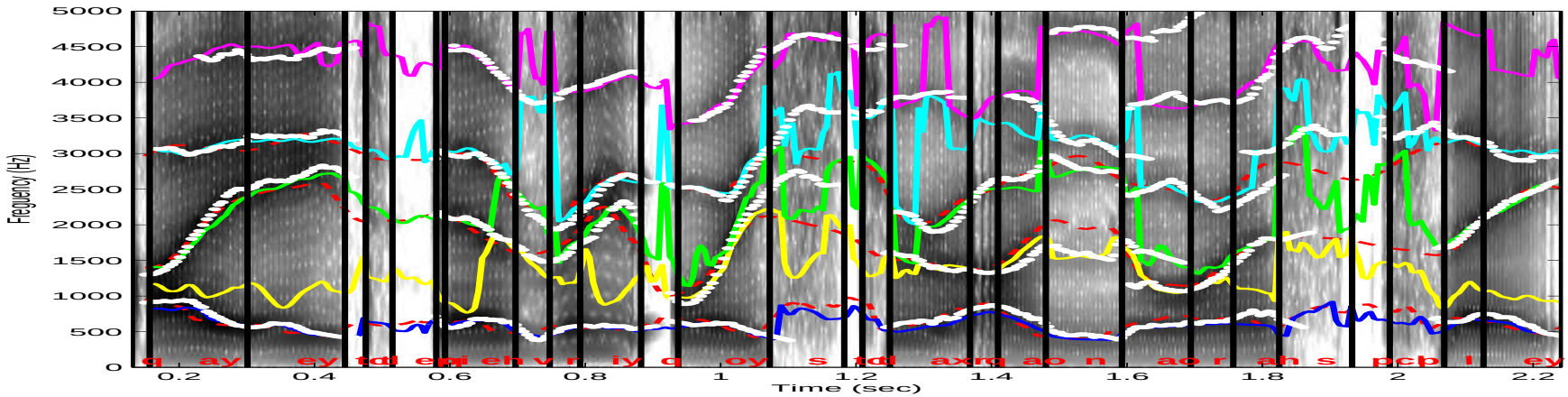


Figure 3.6: DBFT's (white dotted line), hand-labeled (red dashed line) and WaveSurfer's (solid line) formant trajectories superimposed on the spectrogram of the utterance "I ate every oyster on Nora's plate." (TIMITTrain\dr7\fmah1\SX249.WAV) from TIMIT database. E_{DBFT}=[74 122 95], E_{WS}=[52 70 63] and C_{DBFT}=[100 100 99][1]. For this utterance only, the predetermined number of formants to be tracked in WaveSurfer algorithm is fixed to five.

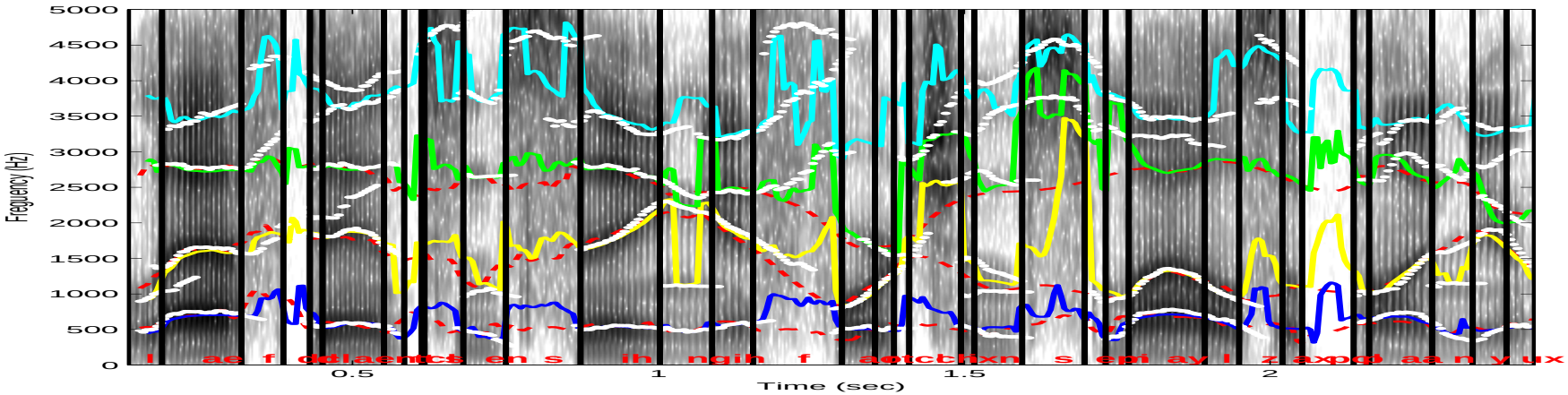


Figure 3.7: DBFT's (white dotted line), hand-labeled (red dashed line) and WaveSurfer's (solid line) formant trajectories superimposed on the spectrogram of the utterance "Laugh, dance, and sing if fortune smiles upon you" (TIMIT\Test\dr5\mbpm0\SX407.WAV) from TIMIT database. E_{DBFT}=[57 73 136], E_{WS}=[45 53 79] and C_{DBFT}=[96 94 87][1]

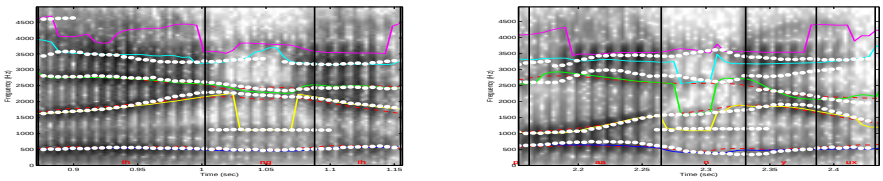


Figure 3.8: Zoomed sections ([0.85s-1.15s] and [2.15s-2.45s]) of the spectrogram given in Figure 3.7. The output of DBFT and WaveSurfer are depicted together for the nasal sound —ng— and —n—.

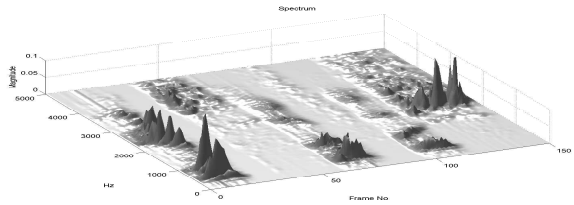


Figure 3.9: DFT spectrogram of the utterance “Books are for schnooks” from TIMIT database plotted in 3-D.

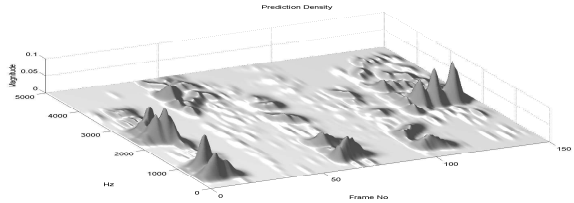


Figure 3.10: Estimated spectrogram of the utterance “Books are for schnooks” from TIMIT database plotted in 3-D.

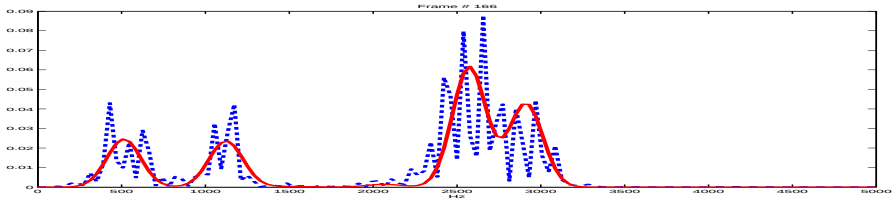


Figure 3.11: Estimated magnitude spectrum (solid line) superimposed on DFT spectrum (dashed line) of the frame no. 166 of the utterance “Books are for schnooks”.

We would like to mention some mistakes done by the algorithm as well. As it is described in more details in Section 3.5.1, the algorithm may produce redundant tracks in some cases. The formants centered around 1KHz at 0.15s, 2.4KHz at 0.5s of the Figure 3.7, formant centered around 0.75KHz at 1.4s of Figure 3.5 can be shown as the examples of the redundant tracks. A track miss might also occur and the examples are already mentioned in the results.

3.5 Discussion

3.5.1 Measurement Selection

Depending on the state space model that defines the formant dynamics and the relation between the states and the measurements, different types of measurements can be extracted from the spectrum and can be fed to the algorithm to be classified into unknown number of formants. Considering the joint task of spectrum representation and formant tracking, our primary choice on the measurements is the samples generated from the magnitude of the spectrum. Although such a choice of measurements let the algorithm perform very well in spectrum representation, it might cause undesirable effects in formant tracking. The algorithm tends to create clusters, which are actually the formant candidates, at frequencies where the spectrum is flat but nonzero. As stated earlier, the formants are related to the peaks of the spectrum, therefore the convex parts of the spectrum might not be the right place to look for a formant candidate. On the other hand, another choice of measurements can be the peaks of the envelope of the spectrum, which might lead the algorithm find good formant candidates but might cause the resulting representation of the spectrum to be oversimplified. It is also possible to assume a non-linear state space model for the formants. The formant states can be directly related to LPC coefficients via nonlinear equations. In that case, the sufficient statistics for the formant states can not be calculated analytically, therefore the use of Rao-Blackwell idea becomes impossible. The use of particle filters is still possible but may not be desired as an increase in the performance is not guaranteed and the computational power required for the algorithm will increase significantly.

The choice of measurements mentioned above are not the only possibilities and are actually a small portion of all possible alternatives. The algorithm presented here stands as a generic one and the choice on the measurements should be specified in accordance with the requirements

of the application.

3.5.2 α Parameter

α parameter determines the probability of creating new formant candidates in the spectrum. An increase in α will favor the particles creating new clusters from the measurements against the particles which associates the measurements with the existing formants. The output of the algorithm is the consistent formant trajectories of the particles having the highest weight (i.e. consistent formant trajectories of the most likely association hypothesis). Even in case of extra formant candidates are created, the resulting candidates must be time consistent (must last for multiple frames and must be supported by sufficiently large number of measurements) to appear at the output. Depending on the nature of the measurements, this might cause the algorithm to produce redundant formant trajectories in the spectrum. Or in contrast, if α parameter is chosen to be very small, the particles which are reluctant to create new clusters and having tendency to combine the closely spaced formants into one will be favored. In our experiments, where the measurements are chosen to be the samples from DFT, the value of α does not alter the average performance significantly unless it is increased/decreased in the order of tens or hundreds. This makes α a sentence independent parameter and it is kept constant in all the utterances (including females) in our experiments.

3.6 Conclusion

In this study, we successfully apply the Dirichlet Process based multi-target tracking algorithm proposed in the first chapter to the problem of tracking VTR frequencies in speech signals. Our approach allows the disappearance/reappearence of VTRs during the speech utterance. Consequently, the number of VTRs that are being tracked is allowed to change in time. DPM model enables us to represent the spectrum as a Gaussian mixture with varying number of components. VTRs are later estimated from the spectrum representation considering the time consistency of the mixture components (formant candidates). The algorithm is tested on real data which are the sentences from TIMIT database. The experimental results show that the algorithm performs very well in tracking varying number of VTRs in real speech signals making the proposed method a practical novel tool in the speech processing

literature.

CHAPTER 4

REGULARIZED PARTICLE METHODS FOR FILTER DERIVATIVE APPROXIMATION

4.1 Introduction

Particle filters (PF) have efficiently been used in the state estimation problem of general state space models [11]. The applicability of the PF to the complex non-linear systems has drew much attention and their use became widespread in many different areas. Although so much is done for the use of PF in the state estimation problem, only a few studies focused on their use in the estimation of the model parameters. The existing particle filtering based parameter estimation methods in the literature can be classified into four subgroups as follows [28].

- Bayesian or Maximum Likelihood (ML)
- Off-line(Batch) or On-line

In Bayesian approach, suitable prior distributions are defined for the unknown parameters and inference is done from the posterior distribution. In ML based methods the likelihood of the measurements given the parameters is maximized with respect to the unknown parameters. The batch algorithms process the whole set of measurements to produce their estimates of the unknown parameters. The on-line methods update their estimate of the unknown parameters sequentially with every available measurement. Here we will consider the filter gradient based particle methods for parameter estimation which is an on-line ML method. In this group of algorithms, the filter derivative is approximated via particle filters by approximating either the path density or the marginal density. The parameter estimation algorithms which tend to approximate the path density [1] [14] suffer from the so-called *degeneracy problem* which re-

sults in an accumulation of error in time. Here we suggest the use of regularization techniques to compensate for the error accumulation of the path based algorithms. In the context of the path density based filter derivative approximation, two sources of error are identified in the previous works. First type of error is the aforementioned *degeneracy problem* which is caused by approximating a density of growing dimension by a finite number of particles. The second type of error will be referred as the *mixing problem* which is caused by the inefficiency of the path based algorithm by closely spacing the particles having weights with opposite signs in filter derivative representation. The mixing problem causes an inefficient approximation of the filter derivative as many particles whose weight might sum up to zero might be closely spaced in a region of the state space that have low total mass. More importantly, this type of error tends to build up in time hence the approximation error is claimed to be less accurate as the data length increases [36]. The method described in this chapter aims to compensate for the error caused by the *mixing problem*. A regularization technique is utilized in order to prevent the *mixing* of the particles with weights having opposite signs and the corresponding error which would tend to accumulate in time. The regularization algorithm we use differs from the standard regularization in that it preserves the sum of particle weights before and after the regularization. Before introducing the details, we need to clarify the basic differences between two particle filtering based parameter estimation methods, the path based and the marginal PF algorithms.

4.2 Parameter Estimation using Particle Filters

The methodology addressed here is a stochastic approximation algorithm which tries to maximize the measurement likelihood with respect to the unknown parameters, which leads to the maximum likelihood estimate of the unknown parameters. The unknown parameter estimates are found iteratively. At each iteration one tries to maximize the likelihood by updating the latest estimates towards the local gradient direction.

$$\theta_{n+1} = \theta_n + \gamma_n D_n \nabla l(\theta_n) \tag{4.1}$$

where θ_n is an estimate of the vector of unknown parameters θ at time n, $\nabla l(\theta_n)$ is the noisy estimate of the gradient of the likelihood function $l(\theta_n)$ with respect to θ evaluated at θ_n, γ_n is the step size and D_n is a positive definite weighting matrix. Choosing D_n to be equal to identity matrix will lead to a stochastic approximation of the steepest ascent algorithm, whereas choos-

ing D_n as the inverse of Hessian matrix will lead to an approximation of Newton-Rhapson method. Below we introduce the likelihood function that is intended to be maximized for parameter estimation in the general state space model.

Consider the state space model defined by the given state dynamic equation and the measurement equation.

$$x_n|x_{n-1} \sim f_\theta(.|x_{n-1}) \tag{4.2}$$

$$y_n|x_n \sim g_\theta(.|x_n) \tag{4.3}$$

The first equation determines the evolution of the state by a Markov transition density $f_\theta(.|x)$. The second equation is the measurement equation which defines the relation between the state vector x_n and the measurements y_n by the conditional density $g_\theta(y|x)$. The measurements y_n are assumed to be conditionally independent given x_n. Both $f(.)$ and $g(.)$ functions may depend on unknown parameters θ and our aim is to estimate θ based on the observation sequence y_n by maximizing a series of log-likelihood functions $\{\log p_\theta(y_{0:n})\}$. Notice that the log-likelihood of the measurements $y_{0:n}$ can be written as a sum of the conditional log-likelihoods as follows.

$$\log p_\theta(y_{0:n}) = \sum_{k=0}^{n} \log p_\theta(y_k|y_{0:k-1}) \tag{4.4}$$

$p_\theta(y_n|y_{0:n-1})$ is known as the predictive likelihood and can be written as,

$$p_\theta(y_n|y_{0:n-1}) = \int\int g_\theta(y_n|x_n) f_\theta(x_n|x_{n-1}) p_\theta(x_{n-1}|y_{0:n-1}) dx_{n-1} dx_n. \tag{4.5}$$

Defining $l(\theta)$ as

$$l(\theta) = \lim_{k\to\infty} \frac{1}{k+1} \sum_{n=0}^{k} \log p_\theta(y_n|y_{0:n-1}). \tag{4.6}$$

Our aim is to maximize $l(\theta)$ by utilizing the aforementioned stochastic approximation algorithm as follows.

$$\theta_{n+1} = \theta_n + \gamma_n \nabla \log p_{\theta_{0:n-1}}(y_n|y_{0:n-1}) \tag{4.7}$$

where θ_{n-1} is the parameter estimate at time $n-1$ and $\nabla \log p_{\theta_{0:n-1}}(y_n|y_{0:n-1})$ denotes the gradient of $\log p_{\theta_{0:n-1}}(y_n|y_{0:n-1})$. Provided that the step size γ_n is a positive non-increasing sequence, such that $\sum \gamma_n = \infty$ and $\sum \gamma_n^2 < \infty$ it can be shown that the iterations will converge to the set of (global or local) maxima of the function $l(\theta)$ [4], [33]. In this approach the numerical approximation of $\nabla \log p_{\theta_{0:n-1}}(y_n|y_{0:n-1})$ is calculated using particle filters. Two different approaches are given in the following sections.

4.3 Path Based Method

In order to calculate the gradient of the predictive likelihood given by equation (4.5) within the particle filtering context, we need the numerical approximations of $p(x_{0:n}|y_{0:n})$ and $\nabla p(x_{0:n}|y_{0:n})$, such that the approximated distribution and the derivatives can be represented by a set of particles and their weights [36].

$$p(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} a_n^{(i)} \delta(x_{0:n}^{(i)}) \tag{4.8}$$

$$\nabla p(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} a_n^{(i)} \beta_n^{(i)} \delta(x_{0:n}^{(i)}) \tag{4.9}$$

By applying the Bayes rule one can write the recursive expression for $p(x_{0:n}|y_{0:n})$ as follows.

$$\begin{aligned} p(x_{0:n}|y_{0:n}) &= p(x_{0:n}|y_n, y_{0:n-1}) \\ &= \frac{p(y_n|x_{0:n}, y_{0:n-1}) p(x_{0:n}|y_{0:n-1})}{p(y_n|y_{0:n-1})} \\ &= \frac{g(y_n|x_n) f(x_n|x_{n-1})}{p(y_n|y_{0:n-1})} p(x_{0:n-1}|y_{0:n-1}) \end{aligned} \tag{4.10}$$

One can also express $p(x_{0:n}|y_{0:n})$ as

$$p(x_{0:n}|y_{0:n}) = \frac{\xi(x_{0:n}|y_{0:n})}{\int \xi(x_{0:n}|y_{0:n}) dx_{0:n}} \tag{4.11}$$

where

$$\xi(x_{0:n}|y_{0:n}) \triangleq g(y_n|x_n) f(x_n|x_{n-1}) p(x_{0:n-1}|y_{0:n-1}) \tag{4.12}$$

is defined as the unnormalized density. The gradient of $p(x_{0:n}|y_{0:n})$ is equal to

$$\nabla p(x_{0:n}|y_{0:n}) = \frac{\nabla \xi(x_{0:n}|y_{0:n})}{\int \xi(x_{0:n}|y_{0:n}) dx_{0:n}} - p(x_{0:n}|y_{0:n}) \frac{\int \nabla \xi(x_{0:n}|y_{0:n}) dx_{0:n}}{\int \xi(x_{0:n}|y_{0:n}) dx_{0:n}} \tag{4.13}$$

where

$$\begin{aligned} & \nabla \xi(x_{0:n}|y_{0:n}) \\ = \; & \nabla([g(y_n|x_n) f(x_n|x_{n-1})] p(x_{0:n-1}|y_{0:n-1})) \\ = \; & \nabla[g(y_n|x_n) f(x_n|x_{n-1})] p(x_{0:n-1}|y_{0:n-1}) + [g(y_n|x_n) f(x_n|x_{n-1})] \nabla p(x_{0:n-1}|y_{0:n-1}) \\ = \; & [g(y_n|x_n) f(x_n|x_{n-1})][\nabla \log g(y_n|x_n) + \nabla \log f(x_n|x_{n-1})] p(x_{0:n-1}|y_{0:n-1}) \\ & + [g(y_n|x_n) f(x_n|x_{n-1})] \nabla p(x_{0:n-1}|y_{0:n-1}) \end{aligned} \tag{4.14}$$

Substituting the particle approximations of $p(x_{0:n-1}|y_{0:n-1})$ and $\nabla p(x_{0:n-1}|y_{0:n-1})$ into (4.12) and (4.14) results:

$$\xi(x_{0:n}|y_{0:n}) = \sum_{i=1}^{N} a_{n-1}^{(i)} [g(y_n|x_n) f(x_n|x_{n-1})] \delta(x_{0:n-1}^{(i)}) \tag{4.15}$$

$$\nabla\xi(x_{0:n}|y_{0:n}) = \sum_{i=1}^{N} a_{n-1}^{(i)} [g(y_n|x_n) f(x_n|x_{n-1})] \times$$
$$[\nabla \log g(y_n|x_n) + \nabla \log f(x_n|x_{n-1}) + \beta_{n-1}^{(i)}] \delta(x_{0:n-1}^{(i)}) \tag{4.16}$$

These functions are to be evaluated on the path space $x_{0:n}$, for which x_n's will be generated from an importance distribution. Then the resulting approximations will be:

$$\xi(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} \tilde{a}_n^{(i)} \delta(x_{0:n}^{(i)}) \tag{4.17}$$

$$\nabla\xi(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} \rho_n^{(i)} \delta(x_{0:n}^{(i)}) \tag{4.18}$$

where

$$\tilde{a}_n^{(i)} = a_{n-1}^{(i)} \frac{g(y_n|x_n^{(i)}) f(x_n^{(i)}|x_{n-1}^{(i)})}{q(x_n^{(i)}|x_{n-1}^{(i)}, y_n)} \tag{4.19}$$

$$\rho_n^{(i)} = \tilde{a}_n^{(i)} [\nabla \log g(y_n|x_n^{(i)}) + \nabla \log f(x_n^{(i)}|x_{n-1}^{(i)}) + \beta_{n-1}^{(i)}] \tag{4.20}$$

Weight update equation for (4.8) and (4.9) will be as follows (see equations (4.11) (4.13)).

$$a_n^{(i)} = \frac{\tilde{a}_n^{(i)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}} \tag{4.21}$$

$$a_n^{(i)} \beta_n^{(i)} = \frac{\rho_n^{(i)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}} - a_n^{(i)} \frac{\sum_{j=1}^{N} \rho_n^{(j)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}} \tag{4.22}$$

The gradient of the log-likelihood, also known as the *score*, can be approximated through the particle approximation of the unnormalized density and its derivative as:

$$\begin{aligned} \nabla \log p(y_n|y_{0:n-1}) &= \frac{\nabla p(y_n|y_{0:n-1})}{p(y_n|y_{0:n-1})} \\ &= \frac{\int \nabla\xi(x_{0:n}|y_{0:n}) dx_{0:n}}{\int \xi(x_{0:n}|y_{0:n}) dx_{0:n}} \\ &\simeq \frac{\sum_{j=1}^{N} \rho_n^{(j)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}} \end{aligned} \tag{4.23}$$

4.4 Marginal Particle filter

In standard sequential Monte Carlo (SMC), the joint posterior density of the state is approximated by using sequential importance sampling. In this approach, the dimension of the target density (joint posterior density) grows with each time step. This simple fact causes the algorithm to degenerate quickly and the use of resampling strategies becomes necessary in order to ensure a reasonable approximation of the target density. Although the standard SMC approximates the joint posterior density, only the filtering density is of interest in most of the applications. Marginal particle filter [30] approximates directly the marginal filtering density where the dimension is fixed and it produces estimates with smaller variance than the conventional particle filters. The second method that we will consider here is similar to the previous one except that the recursive expressions are derived for the marginal of the filtering density (i.e., $p(x_n|y_{0:n})$ instead of $p(x_{0:n}|y_{0:n})$) and its derivative $\nabla p(x_n|y_{0:n})$ [36].

Assume at time $n-1$ we have the particle approximations for $p(x_{n-1}|y_{0:n-1})$ and $\nabla p(x_{n-1}|y_{0:n-1})$.

$$p(x_{n-1}|y_{0:n-1}) \simeq \sum_{i=1}^{N} a_n^{(i)} \delta(x_{n-1}^{(i)}) \tag{4.24}$$

$$\nabla p(x_{n-1}|y_{0:n-1}) \simeq \sum_{i=1}^{N} a_n^{(i)} \beta_n^{(i)} \delta(x_{n-1}^{(i)}) \tag{4.25}$$

We write $p(x_n|y_{0:n})$ as

$$p(x_n|y_{0:n}) = \frac{\xi(x_n|y_{0:n})}{\int \xi(x_n|y_{0:n}) dx_n} \tag{4.26}$$

where

$$\xi(x_n|y_{0:n}) \triangleq g(y_n|x_n) \int f(x_n|x_{n-1}) p(x_{n-1}|y_{0:n-1}) dx_{n-1} \tag{4.27}$$

is defined as the unnormalized density. The gradient of $p(x_n|y_{0:n})$ is equal to

$$\nabla p(x_n|y_{0:n}) = \frac{\nabla \xi(x_n|y_{0:n})}{\int \xi(x_n|y_{0:n}) dx_n} - p(x_n|y_{0:n}) \frac{\int \nabla \xi(x_n|y_{0:n}) dx_n}{\int \xi(x_n|y_{0:n}) dx_n} \tag{4.28}$$

where

$$\begin{aligned} & \nabla \xi(x_n|y_{0:n}) \\ = \; & g(y_n|x_n) \int f(x_n|x_{n-1}) [\nabla \log g(y_n|x_n) + \nabla \log f(x_n|x_{n-1})] p(x_{n-1}|y_{0:n-1}) dx_{n-1} \\ + \; & g(y_n|x_n) \int f(x_n|x_{n-1}) \nabla p(x_{n-1}|y_{0:n-1}) dx_{n-1} \end{aligned} \tag{4.29}$$

Substituting the particle approximations of $p(x_{n-1}|y_{0:n-1})$ and $\nabla p(x_{n-1}|y_{0:n-1})$ into (4.27) and (4.29) results:

$$\begin{aligned}\xi(x_n|y_{0:n}) &= \sum_{k=1}^{N} a_{n-1}^{(k)}[g(y_n|x_n)f(x_n|x_{n-1}^{(k)})] \\ \nabla\xi(x_n|y_{0:n}) &= \sum_{k=1}^{N} a_{n-1}^{(k)}[g(y_n|x_n)f(x_n|x_{n-1}^{(k)})][\nabla \log g(y_n|x_n) + \nabla \log f(x_n|x_{n-1}^{(k)}) + \beta_{n-1}^{(k)}]\end{aligned}$$

These functions are to be evaluated on the path space x_n, for which x_n's will be generated from an importance distribution. Then the resulting approximations will be:

$$\xi(x_n|y_{0:n}) \simeq \sum_{i=1}^{N} \tilde{a}_n^{(i)}\delta(x_n^{(i)}) \tag{4.30}$$

$$\nabla\xi(x_n|y_{0:n}) \simeq \sum_{i=1}^{N} \rho_n^{(i)}\delta(x_n^{(i)}) \tag{4.31}$$

where

$$\tilde{a}_n^{(i)} = \frac{g(y_n|x_n^{(i)})\sum_{k=1}^{N} a_{n-1}^{(k)} f(x_n^{(i)}|x_{n-1}^{(k)})}{\sum_{k=1}^{N} q(x_n^{(i)}|x_{n-1}^{(k)}, y_n)} \tag{4.32}$$

$$\rho_n^{(i)} = \frac{g(y_n|x_n^{(i)})\sum_{k=1}^{N} a_{n-1}^{(k)} f(x_n^{(i)}|x_{n-1}^{(k)})[\nabla \log g(y_n|x_n) + \nabla \log f(x_n^{(i)}|x_{n-1}^{(k)}) + \beta_{n-1}^{(k)}]}{\sum_{k=1}^{N} q(x_n^{(i)}|x_{n-1}^{(k)}, y_n)} \tag{4.33}$$

Weight update equation for (4.24) and (4.25) will be as follows (see equations (4.26) (4.28)).

$$a_n^{(i)} = \frac{\tilde{a}_n^{(i)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}} \tag{4.34}$$

$$a_n^{(i)}\beta_n^{(i)} = \frac{\rho_n^{(i)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}} - a_n^{(i)}\frac{\sum_{j=1}^{N} \rho_n^{(j)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}} \tag{4.35}$$

The gradient of the log-likelihood can be approximated through the particle approximation of the unnormalized density and its derivative as:

$$\begin{aligned}\nabla \log p(y_n|y_{0:n-1}) &= \frac{\nabla p(y_n|y_{0:n-1})}{p(y_n|y_{0:n-1})} \\ &= \frac{\int \nabla\xi(x_n|y_{0:n})dx_n}{\int \xi(x_n|y_{0:n})dx_n} \\ &\simeq \frac{\sum_{j=1}^{N} \rho_n^{(j)}}{\sum_{j=1}^{N} \tilde{a}_n^{(j)}}\end{aligned} \tag{4.36}$$

4.5 Path-based Approach vs Marginal Approach

The main difference between the path based and the marginal algorithm arises from the divert approaches in approximating the filter derivative to compute the log-likelihood gradient.

In [36] and [37] the authors claim that the algorithms which attempt to approximate the filter gradient through the sequence of path densities will produce error which accumulates in time. The rationale behind this idea stems from the fact that the approximated path density $p(x_{0:n}|y_{0:n})$ is growing in dimension hence the approximation with a finite number of particles will fail to represent the joint distribution. On the other hand, in standard SMC the approximation to the density $p(x_n|y_{0:n})$ which is obtained through the path density $p(x_{0:n}|y_{0:n})$ by discarding the first $n-1$ states is considered to be a *valid* approximation of the filtering density since a finite number of particles are used to represent a density with a fixed dimension. More specifically, it can be proved that the particle approximation of $p(x_{0:n}|y_{0:n})$ satisfies the following bound [2]. For any $n \geq 1$ and any test function $f_n : X^n \rightarrow \mathbb{R}$ there exist some constant $c_{n,\theta}(f_n)$ such that for any $N \geq 1$

$$E[(\int_{X^n} f_n(x_{0:n})[\widehat{p}_\theta^N(x_{0:n}|y_{0:n}) - p_\theta(x_{0:n}|y_{0:n})])^2] \leq \frac{c_{n,\theta}(f_n)}{N} \tag{4.37}$$

where N is the number of particles. The problem with the above bound is that the constant $c_{n,\theta}(f_n)$ typically grows exponentially with n. So the error bound increases in time, therefore a proper approximation of the joint density $\{p(x_{0:n}|y_{0:n})\}$ can not be obtained using fixed and finite number of particles. A similar bound can be derived for fixed-lag density estimates (including the marginal density) where the bound is not a function of n but dependant on the fixed-lag.

$$E[(\int_{X^n} f_n(x_{n-L:n})[\widehat{p}_\theta^N(x_{n-L:n}|y_{0:n}) - p_\theta(x_{n-L:n}|y_{0:n})])^2] \leq \frac{d_{L,\theta}(f_n)}{N} \tag{4.38}$$

where L is a positive integer representing the fixed-lag, $f_n : X^n \rightarrow \mathbb{R}$ is any test function and $d_{L,\theta}(f_n)$ is the error bound which is fixed for all n. Particle filters are able to represent the densities $p(x_{n-L}|y_{0:n})$ with fixed dimension L, with finite error which goes to zero in the limiting case as $N \rightarrow \infty$. Therefore the standard path based particle filtering method can properly approximate the filtering density $p(x_n|y_{0:n})$ by neglecting the first $n-1$ states to obtain the marginal density.

Another main difference between the two methods is the resulting representations of the filter gradient. Path based method *mixes* (or closely spaces) the positively and negatively signed particles in the approximation of the filter gradient. That results an inefficient use of particles and the approximation error tends to increase in time [38]. On the other hand, the marginal algorithm keeps the positively and negatively signed particles in separate regions of the state space. This difference is illustrated in Figures 4.1 and 4.2.

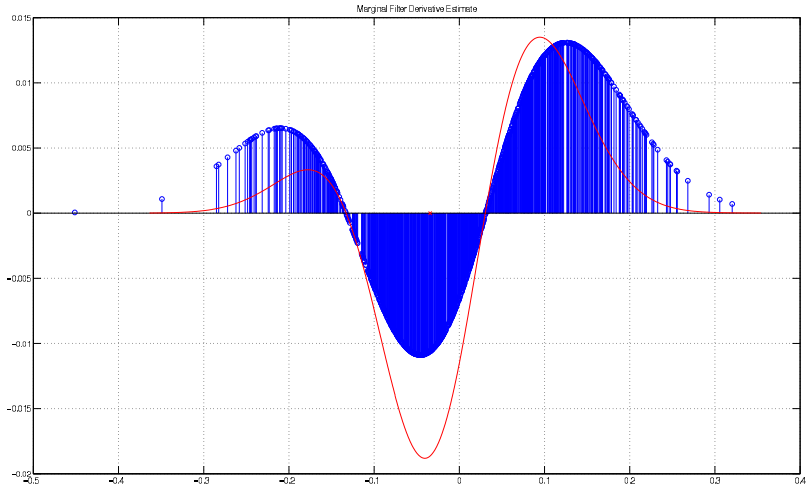


Figure 4.1: Filter derivative estimate for the marginal filter for a linear Gaussian model where the true derivative is plotted as the red-line.

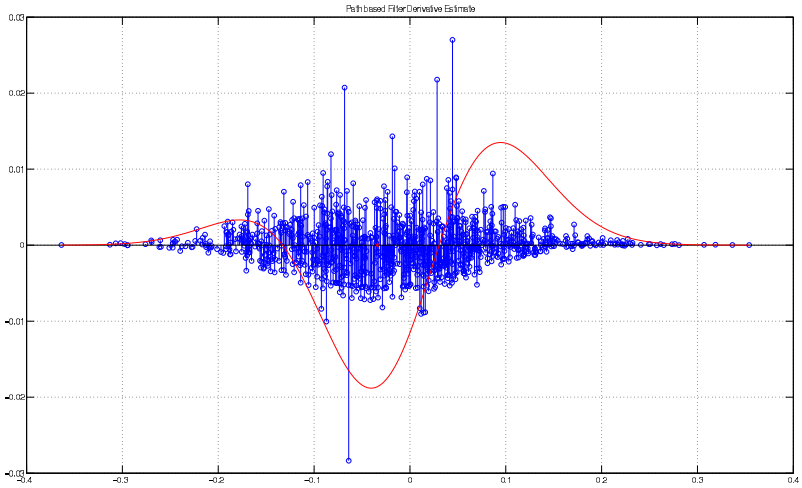


Figure 4.2: Filter derivative estimate for the path-based algorithm for a linear Gaussian model where the true derivative is plotted as the red-line.

The experimental results show that including regularization steps in filter derivative computations can prevent the error caused by the mixing problem which is expected to build up in time and it is possible to keep the log-likelihood gradient approximation of the path density within a neighborhood of the true value.

4.6 Proposed Method

4.6.1 Regularization

Particle filters approximate the posterior distributions by a set of discrete points and their weights. However, it is also possible to construct a continuous approximation of the target density by using kernel smoothing techniques. Adding a kernel smoothing step to the PF leads to regularized particle filters [9]. In the smoothing stage, Dirac delta functions in the density approximation are replaced by kernel functions.

$$p(x_n|y_{0:n}) \simeq \sum_{i=1}^{N} \tilde{a}_n^{(i)} \delta(x_n - x_n^{(i)}) \tag{4.39}$$

$$p_c(x_n|y_{0:n}) \simeq \sum_{i=1}^{N} \tilde{a}_n^{(i)} K_h(x_n - x_n^{(i)}) \tag{4.40}$$

where $K_h(x) = h^{-N_x} K(\frac{x}{h})$, N_x is the dimension of the state, K being a symmetric, unimodal and smooth probability density function such that $\int_{\infty}^{\infty} K(x)dx = 1$ and $h > 0$ being the bandwidth or the smoothing parameter of the kernel. The limit as h tends to zero results sum of Dirac delta functions where as a large value for h would lead to an approximation where the details are obscured. The resulting continuous approximation will inherit all the continuity and differentiability properties of the kernel. If a Gaussian kernel is used the resulting approximation will be a smooth curve having derivatives of all orders. Another famous choice of kernel is Epanechnikov kernel.

$$K(x) = \begin{cases} \frac{N_x+2}{2V_{N_x}}(1 - ||x||^2) & \text{if } ||x|| < 1 \\ 0 & \text{otherwise} \end{cases}$$

Here we suggest the use of regularization techniques to improve the approximation of the filter derivative for the path-based density. We aim to improve the filter derivative approximation of the path based algorithm by using regularization techniques. In the modification scheme we propose, we apply kernel smoothing to the unnormalized density $\xi(x_{0:n}|y_{0:n})$ and its derivative $\nabla\xi(x_{0:n}|y_{0:n})$ where the mixing problem occurs. In the smoothing stage, we interchange each particle with a kernel and distribute its weight among all the particles such that the total weight sum of the representations for both the unnormalized density $\xi(x_{0:n}|y_{0:n})$ and its derivative $\nabla\xi(x_{0:n}|y_{0:n})$ are preserved. The kernel smoothing method we use differs

from the standard method in that if, in the standard regularization, new weights of the (old) particles are calculated, the sum of these might not be equal to the sum of the weights before regularization. The modified smoothing stage is described below.

At time n suppose we have the representation of the unnormalized density $\xi(x_{0:n}|y_{0:n})$ and its derivative $\nabla\xi(x_{0:n}|y_{0:n})$ such that

$$\xi(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} a_n^{(i)} \delta(x_{0:n} - x_{0:n}^{(i)}) \tag{4.41}$$

$$\nabla\xi(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} \rho_n^{(i)} \delta(x_{0:n} - x_{0:n}^{(i)}) \tag{4.42}$$

$$\tag{4.43}$$

After the smoothing stage we use the same set of particles but with different weights in the representation of the functions.

$$\xi(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} \bar{a}_n^{(i)} \delta(x_{0:n} - x_{0:n}^{(i)}) \tag{4.44}$$

$$\nabla\xi(x_{0:n}|y_{0:n}) \simeq \sum_{i=1}^{N} \bar{\rho}_n^{(i)} \delta(x_{0:n} - x_{0:n}^{(i)}) \tag{4.45}$$

$$\tag{4.46}$$

where $\bar{a}_n^{(i)}$ and $\bar{\rho}_n^{(i)}$ are calculated according to

$$\bar{a}_n^{(i)} = \sum_{k=1}^{N} \frac{a_n^{(k)} K(x_n^{(i)} - x_n^{(k)})}{\sum_{j=1}^{N} K(x_n^{(k)} - x_n^{(j)})} \tag{4.47}$$

$$\bar{\rho}_n^{(i)} = \sum_{k=1}^{N} \frac{\rho_n^{(k)} K(x_n^{(i)} - x_n^{(k)})}{\sum_{j=1}^{N} K(x_n^{(k)} - x_n^{(j)})} \tag{4.48}$$

It is easy to check that $\sum_{i=1}^{N} \bar{a}_n^{(i)} = \sum_{i=1}^{N} a_n^{(i)}$ and $\sum_{i=1}^{N} \bar{\rho}_n^{(i)} = \sum_{i=1}^{N} \rho_n^{(i)}$.

$$\sum_{i=1}^{N} \tilde{a}_n^{(i)} = \sum_{i=1}^{N} \sum_{k=1}^{N} \frac{a_n^{(k)} K(x_n^{(i)} - x_n^{(k)})}{\sum_{j=1}^{N} K(x_n^{(k)} - x_n^{(j)})} \tag{4.49}$$

$$= \sum_{k=1}^{N} a_n^{(k)} \frac{\sum_{i=1}^{N} K(x_n^{(i)} - x_n^{(k)})}{\sum_{j=1}^{N} K(x_n^{(k)} - x_n^{(j)})}$$

$$= \sum_{k=1}^{N} a_n^{(k)} \tag{4.50}$$

This step aims to decrease the inefficiency of the path based methods in representing the filter derivative by mixing the oppositely signed particles. Consider the example given in the figures below.

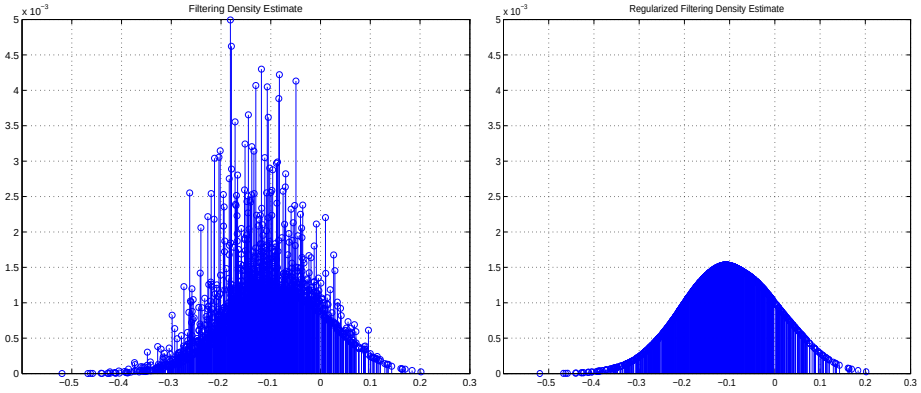


Figure 4.3: Filtering density estimate and regularized filtering density estimate for the path-based algorithm.

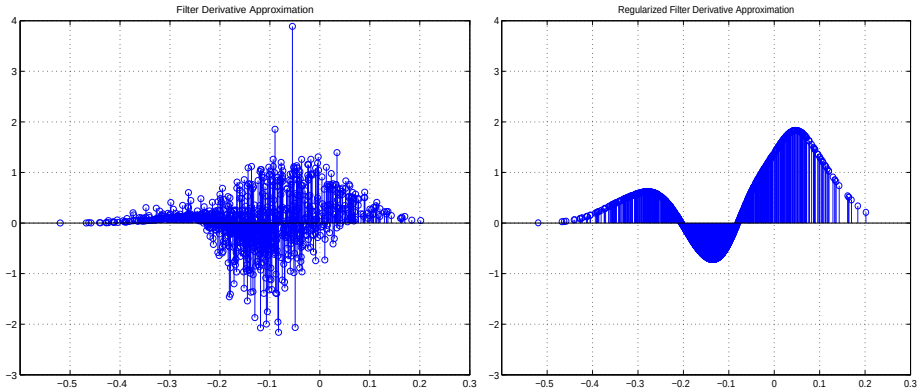


Figure 4.4: Filter derivative estimate and regularized filter derivative estimate for the path-based algorithm.

The resulting approximation is the representation of a derivative of a probability measure where the particles having opposite signs are placed in separate regions of the state space. The derivative of a probability measure is a signed measure and can be expressed as a difference of two probability measures $\nu = c(\pi_1 - \pi_2)$. This approach is known as weak derivative decomposition and it is possible to decompose a given signed measure by using arbitrarily many different probability measures. From weak derivative point of view, this representation corresponds to Hahn-Jordan decomposition of the filter derivative such that the probability measures of the decomposition are concentrated in disjoint regions [38] for an appropriate kernel bandwith. Consequently, the algorithm does not suffer from the *mixing problem*.

The *mixing problem* of the path based algorithm is caused by the fact that in this method

each particle and its weight are updated separately. Depending on the MC realization at the sampling stage, two particles which are closely spaced might have opposite signs. By adding the regularization step, the weights are computed by considering all the particles hence the effects of MC realization is removed in weight calculations. This results in more consistent estimates of the filter gradient. Consider the example given below. Here we compare log-likelihood gradient estimate $\log p_\theta(y_{0:n}) = \sum_{k=0}^{n} \log p_\theta(y_k|y_{0:k-1})$ of 30 runs of the path based and the regularized path based algorithm. The linear Gaussian model given in 4.7 is used in our example. It is evident that the regularized path based method produces more consistent estimates having smaller variance.

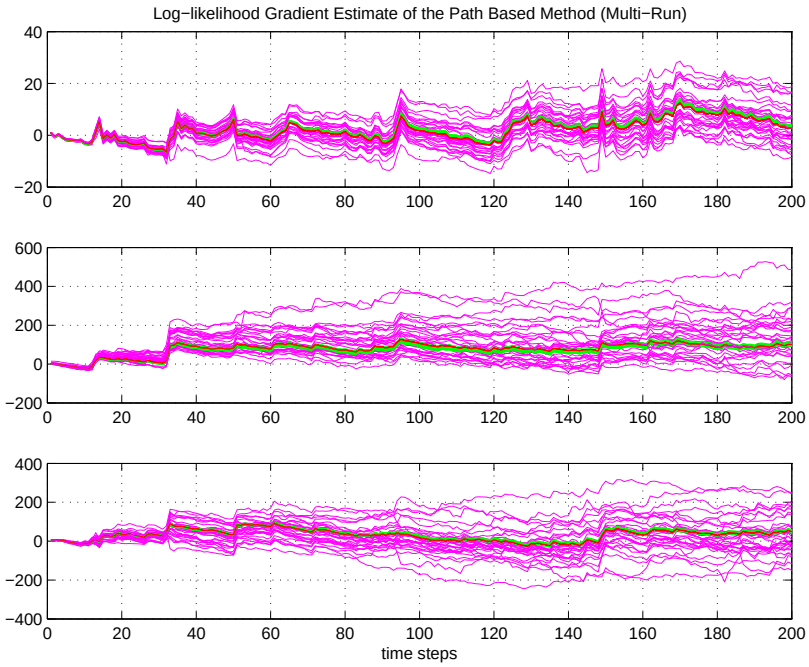


Figure 4.5: Log-likelihood gradient estimate of the path based method on multi-runs. Log-likelihood gradient w.r.t $\theta = [\phi \quad \sigma_v \quad \sigma_w]$ are depicted respectively from top to bottom. 'Pink' line indicates the approximated log-likelihood gradient by the path based algorithm. 'Green' line indicates the true log-likelihood gradient computed by Kalman filter.

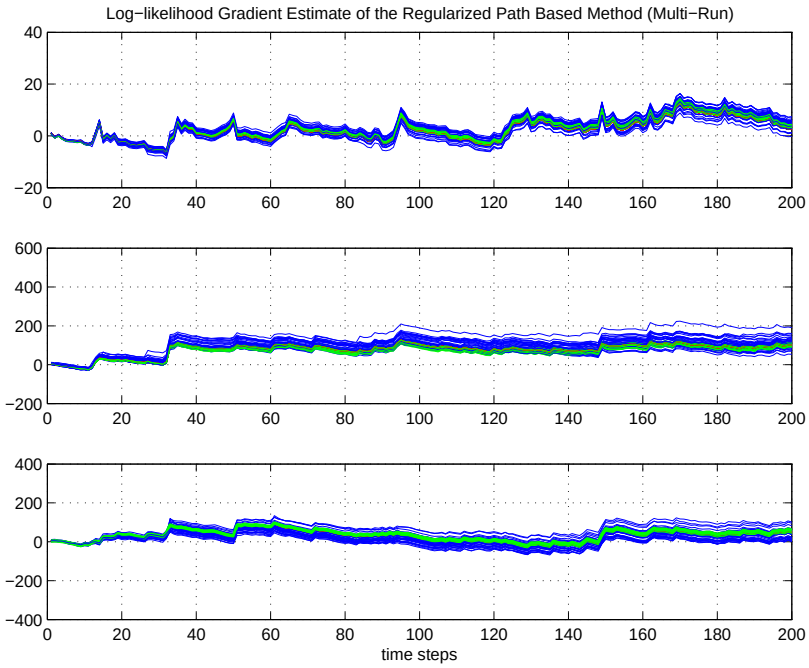


Figure 4.6: Log-likelihood gradient estimate of the regularized path based method on multi-runs. Log-likelihood gradient w.r.t $\theta = [\phi \quad \sigma_v \quad \sigma_w]$ are depicted respectively from top to bottom. 'Blue' line indicates the approximated log-likelihood gradient by the path based algorithm. 'Green' line indicates the true log-likelihood gradient computed by Kalman filter.

4.6.2 Bandwidth selection

Problems with the regularization methods are that they introduce bias in the log-likelihood derivative and one should determine the appropriate kernel bandwidth to approximate the continuous distribution. In our method, a small bandwidth would lead to the same representation of the path based method. On the other hand, choosing a large bandwidth would cause the resulting approximation to loose the important details of the filter gradient leaving an oversimplified representation behind. Adding the regularization step with a poor choice on the bandwidth would result in a poor approximation of the derivative and causes a bias in the log-likelihood gradient estimates.

4.7 Simulation Results

4.7.1 Linear Gaussian Model

Here we illustrate the effects of regularization on the gradient estimate of a linear Gaussian state space model. Consider the model given below.

$$x_{n+1} = \phi x_n + v_n \tag{4.51}$$

$$y_n = x_n + w_n \tag{4.52}$$

where $v_n \sim \mathcal{N}(0, \sigma_v)$ and $w_n \sim \mathcal{N}(0, \sigma_w)$. The unknown parameters are $\theta = [\phi, \sigma_v, \sigma_w]$. For this linear Gaussian model, it is possible to compute the derivative of the log-likelihood analytically using Kalman filter and its derivative. We compare the log-likelihood gradient estimate (score function) of the standard path based method and the regularized path based method depicted with the true value of the score in Figure 4.14. The results show that adding the regularization step to the path based algorithm will produce more consistent approximation of the log-likelihood gradient.

In our second experiment we assumed that the parameters of the model $\theta^* = [\phi, \sigma_v, \sigma_w]$ are unknown and needed to be estimated online. The values of the true parameters are set to $\theta = [0.8, 0.1, 0.1]$. Both the regularized algorithm and the path based algorithm are run with 500 particles. The variable step size is chosen as $\gamma_n = \gamma_0 n^{-\frac{5}{6}}$. A total number of 50 MC runs are made for each algorithm and we compare the RMS error between the estimated values of the parameters and the true values in Figures 4.8 and 4.9. In the MC runs, the regularized path based method produces more consistent estimates of the parameters with a shorter convergence time and reduced RMS error when compared to the standard path based method. Typical outputs of the algorithms on a single run are also given in Figures 4.10 and 4.11. The noisy behavior of the unmodified path based method is evident in its single run output.

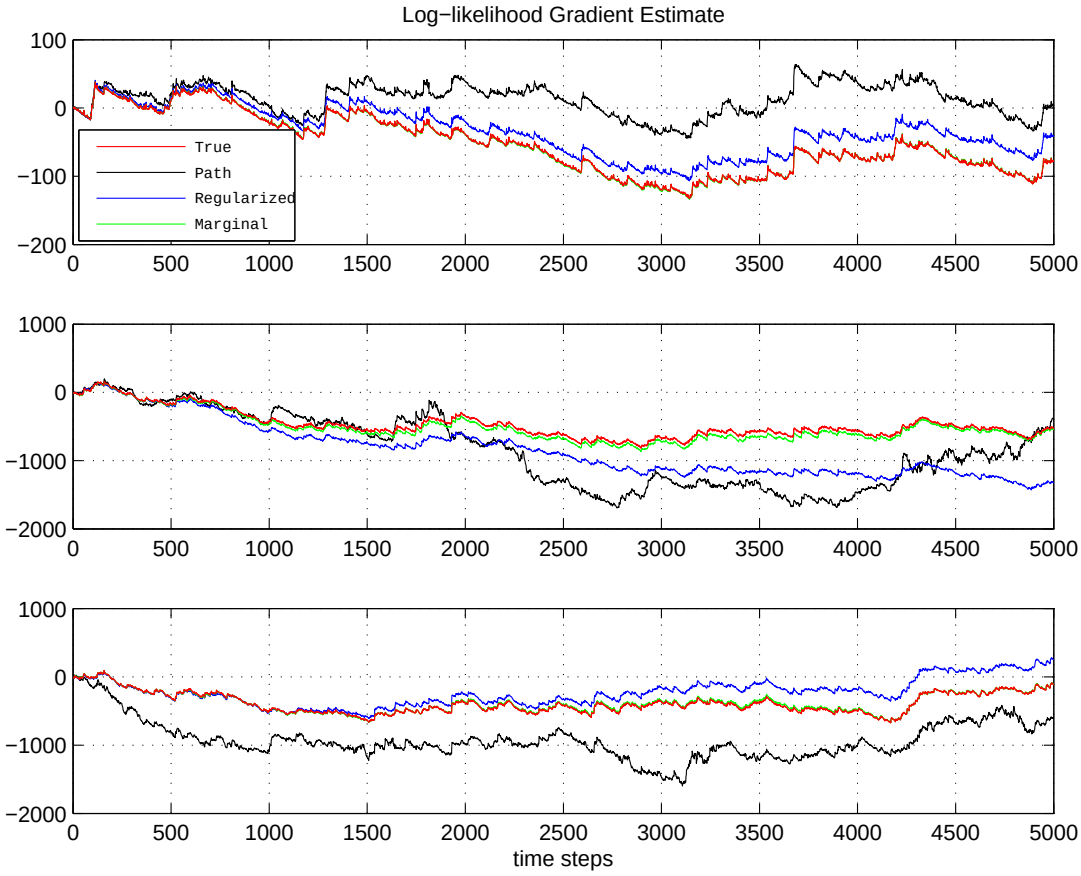


Figure 4.7: Comparison of the algorithms: Three figures, corresponds to the log-likelihood derivative w.r.t. ϕ, σ_v and σ_w. The red line is the true log-likelihood gradient. Green line is the approximation found by using the marginal density. The blue line, which represents the regularized path based method, remains within the neighborhood of the true log-likelihood gradient, whereas the black line, which represents the standard path based method, degenerates in time.

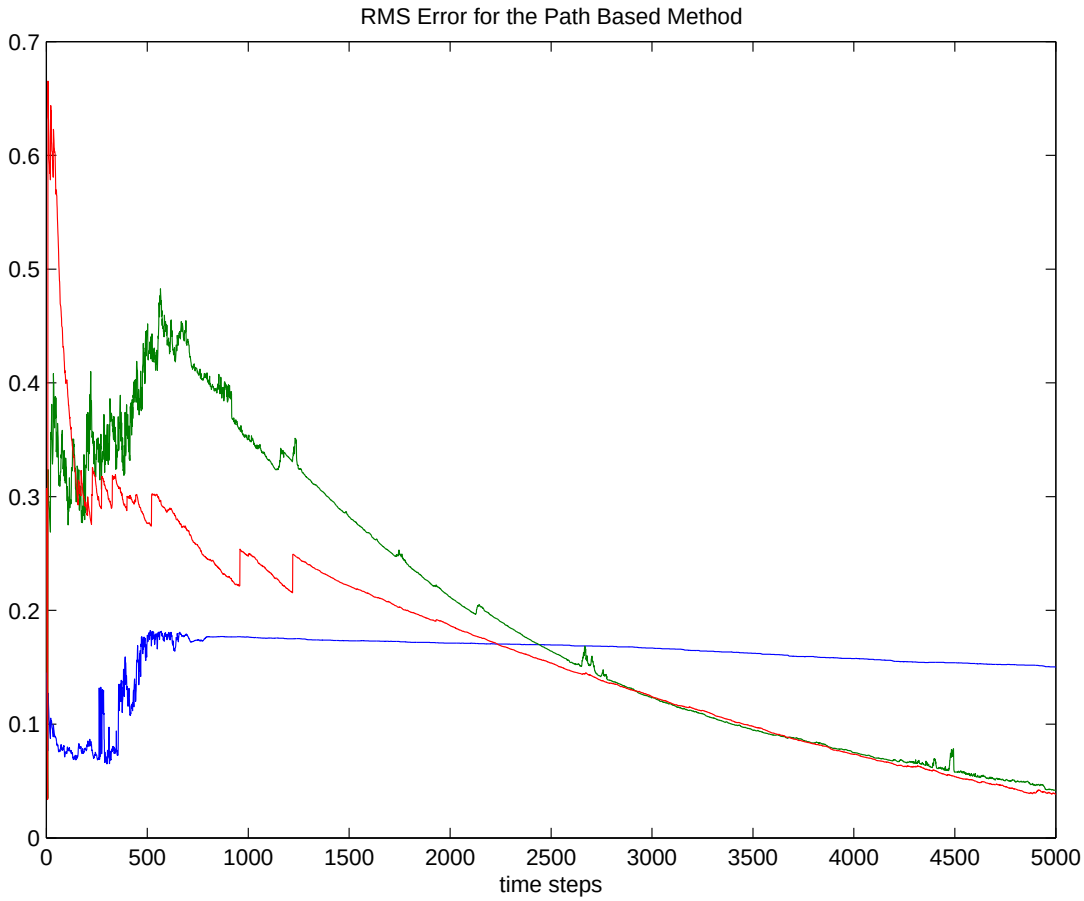


Figure 4.8: Comparison of the algorithms: RMS error of the path based algorithm for the estimation of the unknown parameters ϕ, σ_v and σ_w are depicted in 'blue', 'green' and 'red' lines respectively.

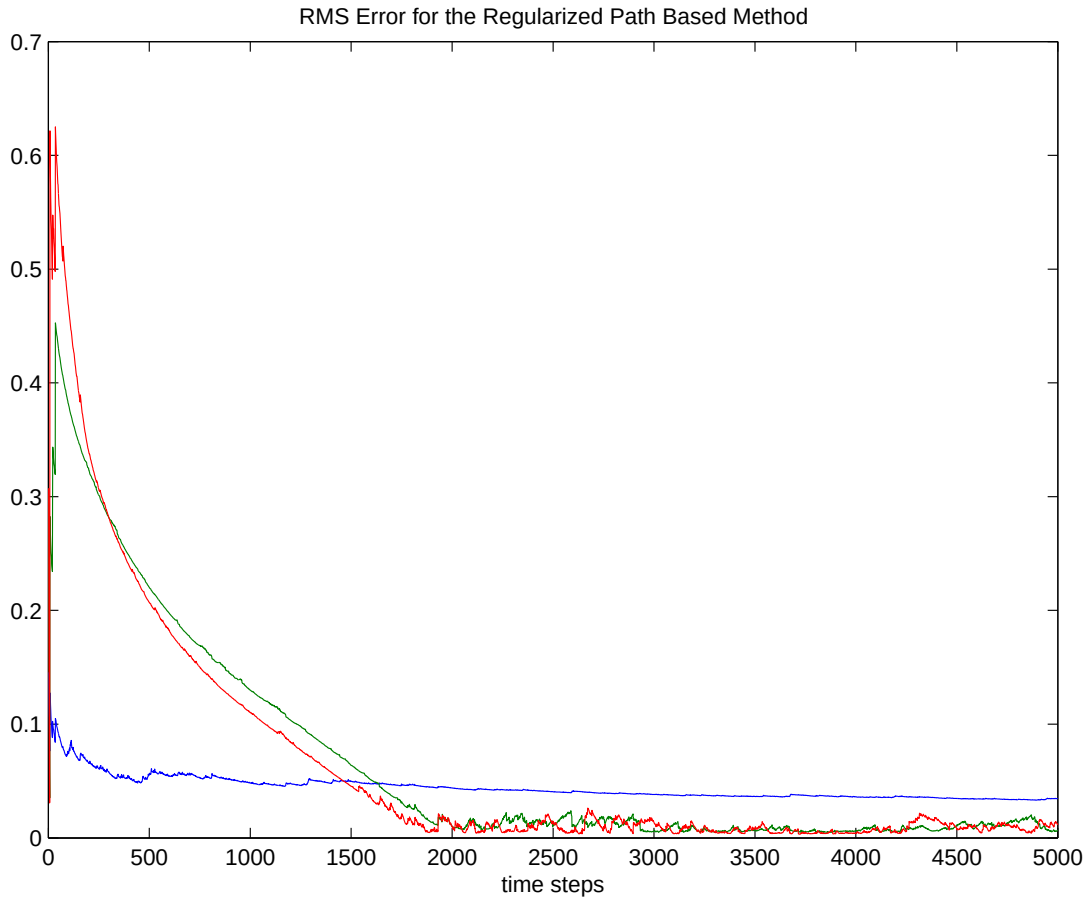


Figure 4.9: Comparison of the algorithms: RMS error of the regularized path based algorithm for the estimation of the unknown parameters ϕ, σ_v and σ_w are depicted in 'blue', 'green' and 'red' lines respectively.

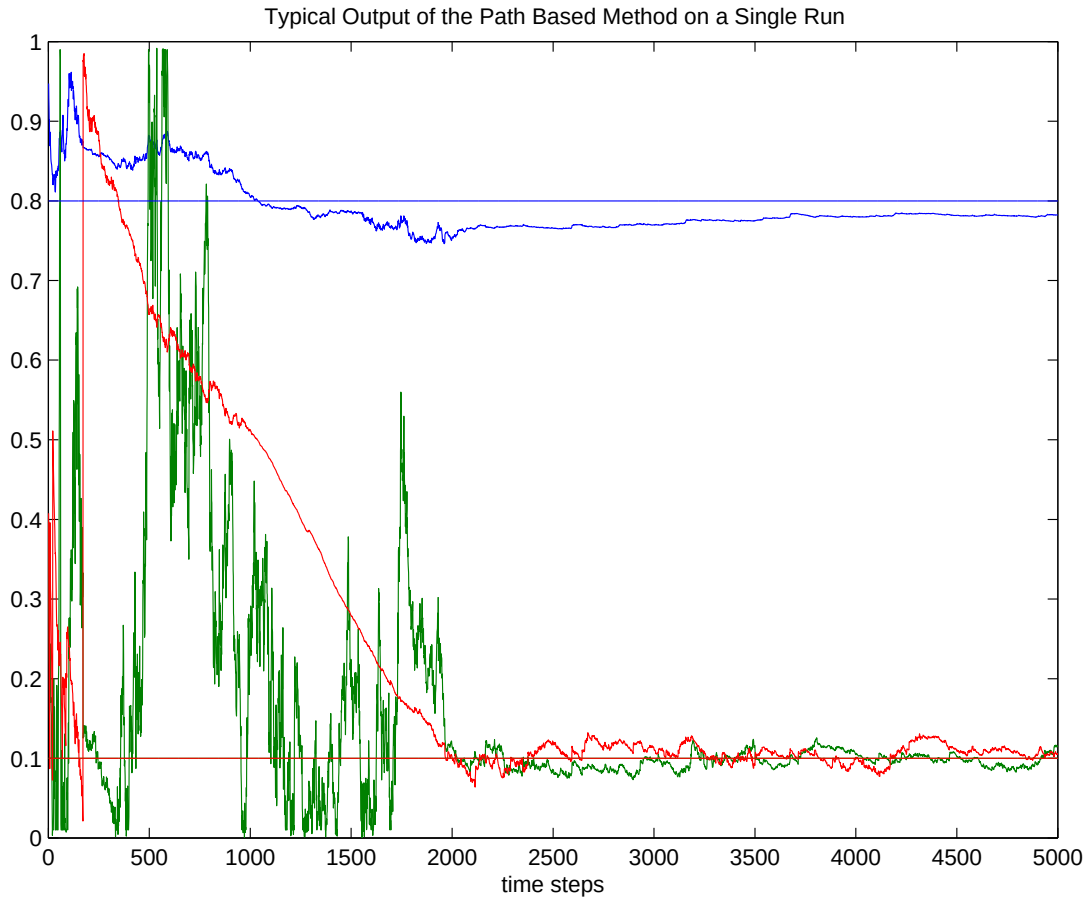


Figure 4.10: A typical single run of the path based algorithm for the estimation of the unknown parameters ϕ, σ_v and σ_w. The estimated and true values are depicted in 'blue', 'green' and 'red' lines respectively.

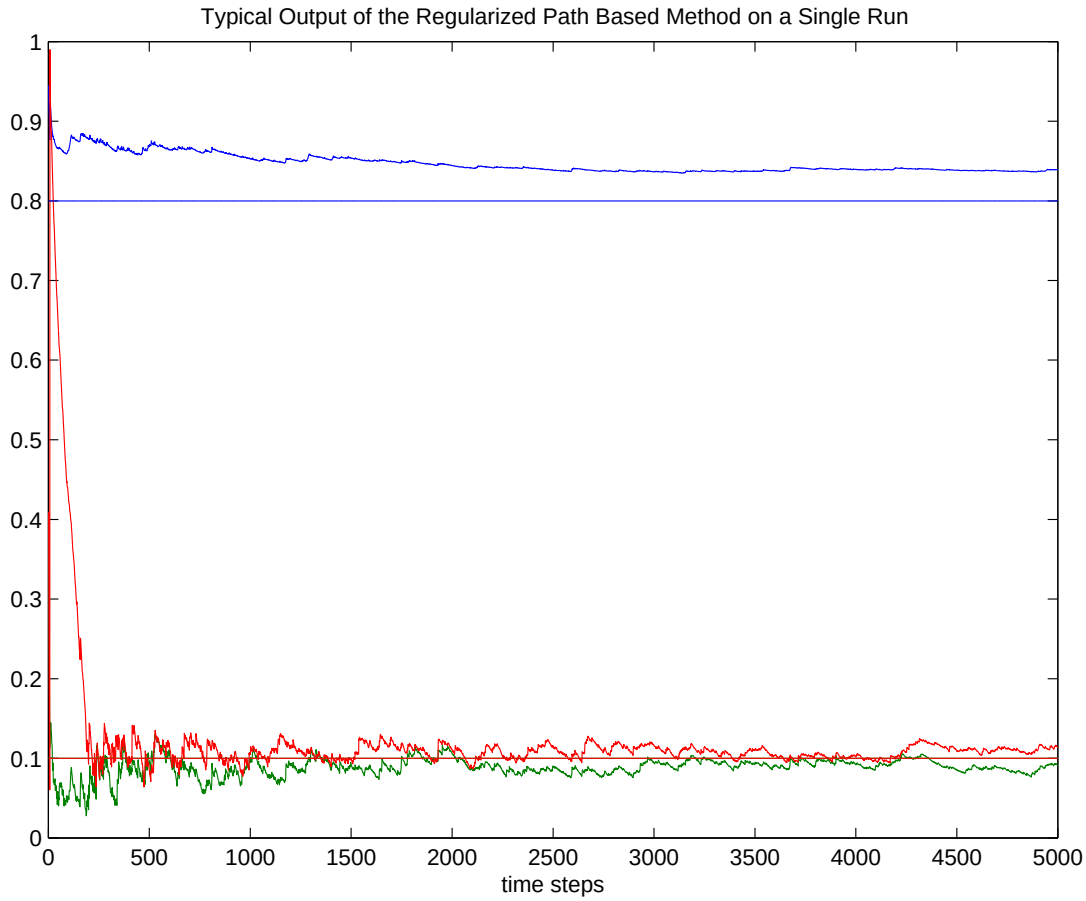


Figure 4.11: A typical single run of the regularized path based algorithm for the estimation of the unknown parameters ϕ, σ_v and σ_w. The estimated and true values are depicted in 'blue', 'green' and 'red' lines respectively.

4.7.2 Jump Markov Linear Systems

4.7.2.1 Switching Noise Model

Here we illustrate the performance improvement gained by modifying the path based method on a switching noise model. Consider the jump Markov system given below.

$$x_{n+1} = \phi x_n + \sigma_{z_n} v_n \tag{4.53}$$

$$y_n = x_n + w_n \tag{4.54}$$

where $v_n \sim \mathcal{N}(0, 1)$ and $w_n \sim \mathcal{N}(0, 10)$. The process noise variance is switching according to underlying Markov chain. The transition probability matrix of the Markov chain is

$$p(z_n|z_{n-1}) = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}. \tag{4.55}$$

where $z_n \quad \epsilon \quad \{1, 2\}$. The unknown parameters are $\theta = [\phi \quad \sigma_1 \quad \sigma_2]$. True values of the unknown parameters are set to $\theta^* = [0.9 \quad 1.3 \quad 0.5]$. The effect of regularization can be clearly observed by comparing the log-likelihood derivative approximation of the path based algorithm and the regularized algorithm. The standard path-based method diverges in time whereas the approximation of the regularized path based method remains within the neighborhood of the true log-likelihood derivative which in this example is approximated by running the marginal algorithm with too many particles (blue line). Output of the algorithm that uses GPB approximation in filter derivative computations is also depicted on the figures.

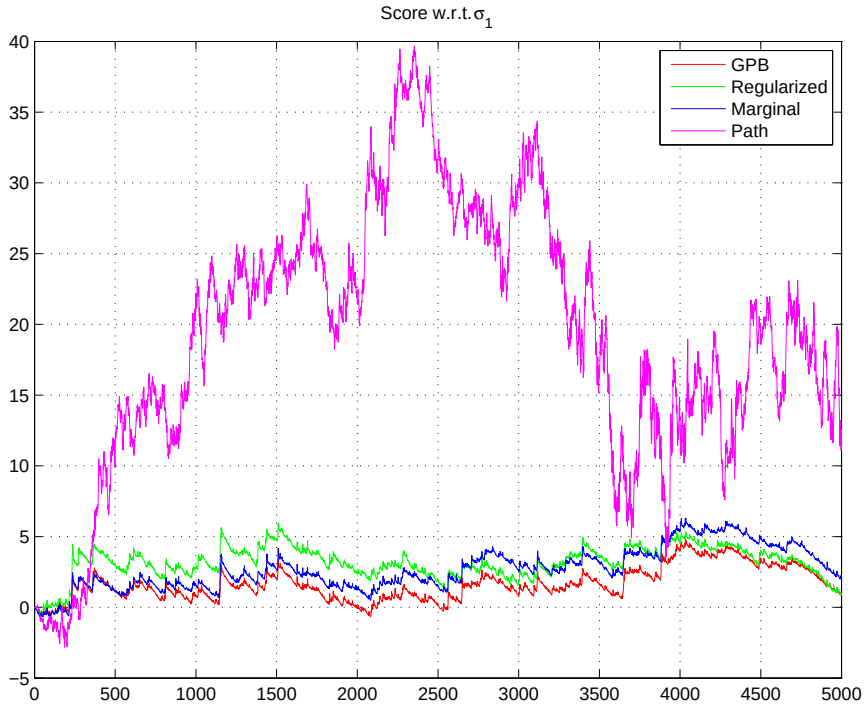


Figure 4.12: Comparison of the algorithms: The log-likelihood derivative w.r.t. σ_1. The pink line, which represents the regularized path based method, remains within the neighborhood of the true log-likelihood gradient (blue line), whereas the green line, which represents the standard path based method, degenerates in time.

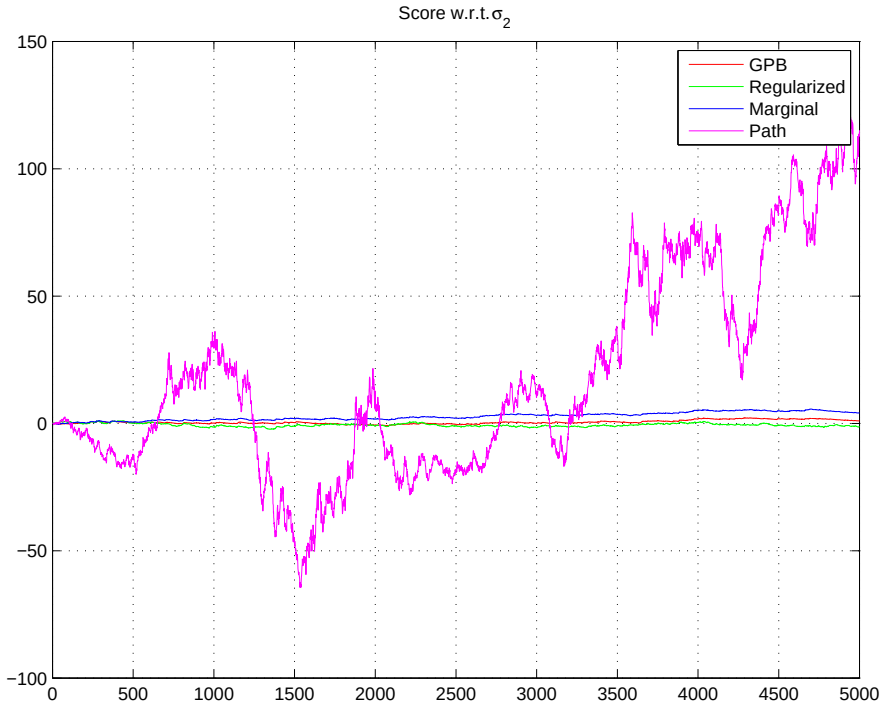


Figure 4.13: Comparison of the algorithms: The log-likelihood derivative w.r.t. σ_2. The pink line, which represents the regularized path based method, remains within the neighborhood of the true log-likelihood gradient (blue line), whereas the green line, which represents the standard path based method, degenerates in time.

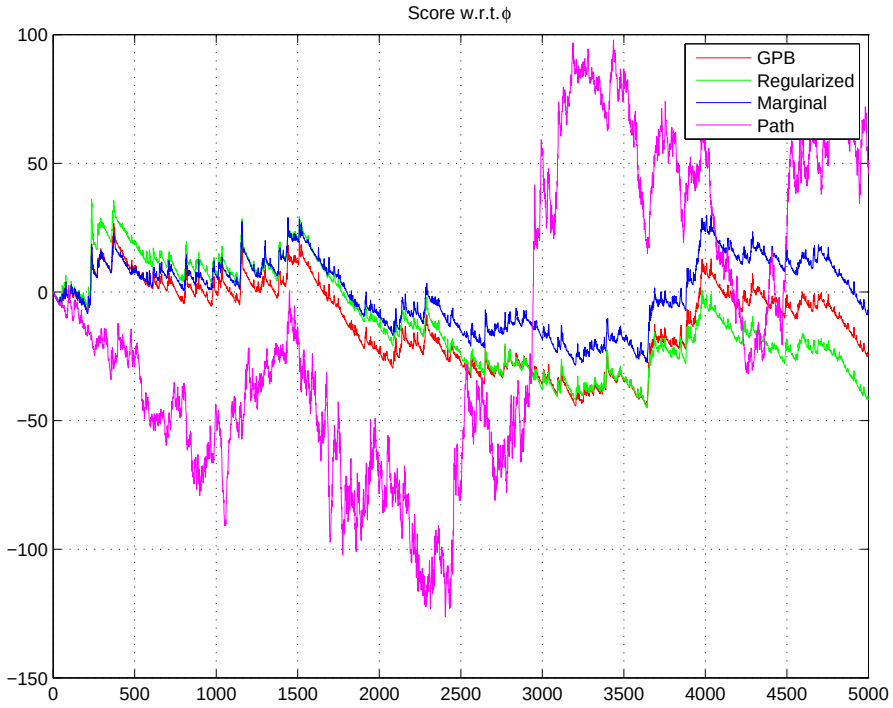


Figure 4.14: Comparison of the algorithms: The log-likelihood derivative w.r.t. ϕ. The pink line, which represents the regularized path based method, remains within the neighborhood of the true log-likelihood gradient (blue line), whereas the green line, which represents the standard path based method, degenerates in time.

4.8 Conclusion

A new method is proposed for the static parameter estimation of general state space systems via particle filters. The method proposed here approximates the path-density and its derivative by a set of particles and utilize kernel smoothing techniques to prevent the degeneracy of the algorithm which would cause error accumulation and leads the algorithm to diverge in time. The regularization technique we propose is special in that it keeps the weight sum constant before and after the regularization step unlike the standard regularization methods. Our experiments show that the proposed algorithm is capable of approximating the filter gradient with a good performance. Moreover we show that including the proposed regularization step in the algorithm results more consistent approximation of the log-likelihood gradient. The algorithm has been shown via a standard example to reduce the RMS parameter estimation error compared to the unmodified path based method. It is also important to emphasize that our approach illustrates that the path based methods can be utilized in parameter estimation with much better accuracy than the ones in the literature.

CHAPTER 5

CONCLUSION

In this research we study a number of problems which involve complex non-linear models which necessitate utilization of particle filtering techniques in the solution. We propose novel ideas in the solution of these problems and contribute to the existing methods in the literature. The main contributions of the thesis work can be summarized as follows.

- A new probabilistic model for full Bayesian multi-target tracking is proposed. The resulting algorithm is a complete multi-target tracking system which uses time varying Dirichlet process based models. The proposed algorithm is quite novel in many ways as it combines state of the art techniques with novel ideas. In our experiments we show that the algorithm performs better than joint probabilistic data association (JPDA) and global nearest neighborhood (GNN) algorithms which use standard (M/N) ad-hoc logic for track initiation and deletion procedures. In addition to its capability of constructing a mathematical model for track deletion/initiation tasks, the proposed method can keep multiple hypotheses for track to measurement/clutter association and it is able to outperform both JPDA and GNN algorithms under heavy clutter.

- Dirichlet process based multi-target tracking algorithm is successfully adapted for tracking variable number of vocal tract resonance frequencies in speech signal spectrum. The proposed method is an original approach to the formant tracking problem as it represents the spectrum as a Gaussian mixture with varying number of components. The capability of the algorithm in formant tracking is shown on an extensive set of real data and the resulting paper is accepted to be published in IEEE Transactions on Audio,

Speech and Language Processing.

- A new method is proposed for the static parameter estimation of general state space systems via particle filters. The method proposed here approximates the path-density and its derivative by a set of particles and utilize kernel smoothing techniques to prevent the degeneracy of the algorithm which would cause error accumulation and leads the algorithm to diverge in time. The experimental results show that including the proposed regularization method in the path based parameter estimation algorithm will produce more consistent estimates and reduce the RMS parameter estimation error compared to the unmodified path based algorithm.

REFERENCES

[1] Doucet A. and Tadic V.B. Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics*, 55(2):409–422, 2003.

[2] Christophe Andrieu. On-line parameter estimation in general state-space models. In *In Proceedings of the 44th Conference on Decision and Control*, pages 332–337, 2005.

[3] J.R. Bellegarda. A global, boundary-centric framework for unit selection text-to-speech synthesis. *IEEE Trans. Speech Audio Processing*, 14(4):990–997, 2006.

[4] Albert Benveniste, Pierre Priouret, and Michel Métivier. *Adaptive algorithms and stochastic approximations*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[5] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House Radar Library, 1999.

[6] D. Blackwell and J.B. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

[7] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings* -, 146(1):2–7, Feb 1999.

[8] T. Claes, I. Dologlou, L. ten Bosch, and D. van Compernolle. A novel feature transformation for vocal tract length normalization in automatic speech recognition. *IEEE Trans. Speech Audio Processing*, 6:549–557, 1998.

[9] Li Deng, Xiaodong Cui, R. Pruvenok, Yanyi Chen, S. Momen, and A. Alwan. A database of vocal tract resonance trajectories for research in speech processing. In *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages 369–372, 2006.

[10] R. Van der Merwe, A. Doucet, N. De Freitas, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems*, 2001.

[11] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in practice*. Springer-Verlag, 2001.

[12] A. Doucet, B.N. Vo, C. Andrieu, and M. Davy. Particle filtering for multi-target tracking and sensor management. In *International conference on information fusion*, 2002.

[13] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

[14] Cerou F., LeGland F., and Newton N.J. Stochastic particle methods for linear tangent filtering equations, 2001.

[15] G Fant, editor. *Acoustic Theory of Speech Production*. Mouton, 1960.

[16] James L. Flanagan, editor. *Speech analysis synthesis and perception*. Springer Verlang, 1965.

[17] E.B. Fox, D.S. Choi, and A.S. Willsky. Nonparametric Bayesian methods for large scale multi-target tracking. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, Canada*, 2006.

[18] P. Garner and W. Holmes. On the robust incorporation of formant features into hidden Markov models for automatic speech recognition. In *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

[19] H. Gauvrit, J.-P. Le Cadre, and C. Jauffret. A formulation of multitarget tracking as an incomplete data problem. *IEEE Transactions on Aerospace and Electronic Systems*, 33:1242–1257, 1997.

[20] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr 1993.

[21] C. Hue, J. P. Le Cadre, and P. Perez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50:309–325, 2002.

[22] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[23] A.H. Jaswinski. *Stochastic processes and filtering theory*. Academic Press, NewYork, 1970.

[24] M. I. Jordan. Dirichlet processes Chinese restaurant processes and all that. In *Tutorial presentation at the NIPS Conference*, 2005.

[25] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transaction on Automatic Control*, 45:477–482, 2000.

[26] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. pages 182–193, 1997.

[27] Keiji Kanazawa, Daphne Koller, and Stuart Russell. Stochastic simulation algorithms for dynamic probabilistic networks, 1995.

[28] N. Kantas, A Doucet, S.S. Singh, and J.M Maciejowski. An overview of sequential monte carlo methods for parameter estimation in general state-space models. In *15th IFAC Symposium on System Identification*, 2009.

[29] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian non-linear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[30] Mike Klaas, Nando de Freitas, and Arnaud Doucet. Toward practical n2; monte carlo: the marginal particle filter. In *Uncertainty in Artificial Intelligence*. AUAI Press, 2005.

[31] E. Klabbers and R. Veldhuis. Reducing audible spectral discontinuities. *IEEE Trans. Speech Audio Processing*, 9:39–51, 2001.

[32] D. Klatt. Software for a cascade/parallel formant synthesizer. *Journal of the Acoustic Society of America*, 67(3):971–995, 1980.

[33] F. LeGland and L. Mevel. Recursive estimation in hidden markov models. *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, 4:3468–3473 vol.4, Dec 1997.

[34] R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

[35] İ. Yücel Özbek and Mübeccel Demirekler. Vocal tract resonances tracking based on voiced and unvoiced speech classification using dynamic programming and fixed interval Kalman smoother. In *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[36] G. Poyiadjis. *Particle Methods for Parameter Estimation in General State Space models*. PhD thesis, University of Cambridge, 2006.

[37] G. Poyiadjis, A. Doucet, and S.S. Singh. Particle methods for optimal filter derivative: application to parameter estimation. *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, 5:v/925–v/928 Vol. 5, March 2005.

[38] George Poyiadjis, Arnaud Doucet, and Sumeetpal S. Singh. Maximum likelihood parameter estimation in general state-space models using particle methods. In *Proc of the American Stat. Assoc*, 2005.

[39] L. R. Rabiner. Digital-formant synthesizer for speech-synthesis studies. *Journal of the Acoustic Society of America*, 43(4):822–828, 1968.

[40] Dimitrios Rentzos, Saeed Vaseghi, Qin Yan, and Ching-Hsiang Ho. Parametric formant modeling and transformation in voice conversion. *International Journal of Speech Technology Springer*, 8:227–245, 2005.

[41] Rohit Sinha S. Umesh. A study of filter bank smoothing in MFCC features for recognition of children's speech. *IEEE Trans. Speech Audio Processing*, 15(8):2418–2430, 2007.

[42] S. Sarkka, A. Vehtari, and J. Lampinen. Rao-Blackwellized particle filter for multiple target tracking. *Information Fusion*, 8:2–15, 2007.

[43] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[44] R. L. Streit and T. E. Luginbuhl. Maximum likelihood method for probabilistic multihypothesis tracking. In *Proceedings of Signal and Data Processing of Small Targets*, 1994.

[45] A. Watanabe and T. Sakata. Reliable methods for estimating relative vocal tract lengths from formant trajectories of common words. *IEEE Trans. Speech Audio Processing*, 14(4):1193– 1204, 2006.

[46] L. Welling and H. Ney. Formant estimation for speech recognition. *IEEE Trans. Speech Audio Processing*, 6:36–48, 1998.

[47] Q. Yan, S. Vaseghi, D. Rentzos, and C-H. Ho. Analysis and synthesis of formant spaces of british, australian, and american accents. *IEEE Trans. Speech Audio Processing*, 15:676–689, 2007.

[48] Qin Yan, Saeed Vaseghi, Esfandiar Zavarehei, Ben Milner, Jonathan Darch, Paul White, and Ioannis Andrianakis. Formant tracking linear prediction model using HMMs and Kalman filters for noisy speech processing. *Comput. Speech Lang.*, 21(3):543–561, 2007.

[49] Özkan E, Özbek i. Y., and Demirekler M. Dynamic speech spectrum representation and tracking variable number of vocal tract resonance frequencies with time varying dirichlet process mixture models. *IEEE Transactions on Audio, Speech, and Language Processing*, Accepted for publication, 2009.

[50] Parham Zolfaghari, Hiroko Kato, Yasuhiro Minami, Atsushi Nakamura, and Shigeru Katagiri. Dynamic assignment of Gaussian components in modeling speech spectra. *The Journal of VLSI Signal Processing*, 45(1-2):7–19, 2006.

APPENDIX A

DETAILS OF THE ALGORITHM IN APPLICATION TO NONLINEAR MODELS

Algorithm:

- Step 1 Initialization
- For $i = 1, .., N$
  - Do $w_0^{(i)} \leftarrow \frac{1}{N}$
- Step 2 Iterations
- For $t = 1, 2, \ldots$ do
  - For each particle $i = 1, .., N$ do
    - For $k = 1, \ldots, n$, sample $\widetilde{c}_{k,t}^{(i)} \sim q(c_{k,t}|y_{k,t}, \widetilde{c}_{1:k-1,t}^{(i)}, c_{t-r:t-1}^{(i)}, x_{t-1}^{(i)})$
    - For $j \in c_{t-r:t-1}^{(i)}$, sample $\widetilde{x}_{j,t}^{(i)} \sim q(x_{j,t}|x_{j,t-1}^{(i)}, y_t, \widetilde{c}_t^{(i)})$
    - For $j \in \overline{c_{t-r:t-1}^{(i)}} \cap \widetilde{c}_t^{(i)}$, sample $\widetilde{x}_{j,t}^{(i)} \sim q(x_{j,t}|y_t, \widetilde{c}_t^{(i)})$
  - For $i = 1, .., N$, update the weights as follows

$$\widetilde{w}_t^{(i)} \propto w_{t-1}^{(i)} \frac{\prod_{k=1}^n p(\mathbf{y}_{k,t}|\widetilde{c}_{k,t}^{(i)}, \widetilde{\mathbf{x}}_t^{(i)}) \prod_{k=1}^n \Pr(\widetilde{\mathbf{c}}_{k,t}^{(i)}|\widetilde{\mathbf{c}}_{1:k-1,t}^{(i)}, \mathbf{c}_{t-r:t-1}^{(i)})}{\prod_{k=1}^n q(\widetilde{c}_{k,t}^{(i)}|\mathbf{y}_{k,t}, \widetilde{\mathbf{c}}_{k-1,t}^{(i)}, \mathbf{c}_{t-r:t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)})} \frac{\prod_{j \in \mathbf{c}_{t-r:t-1}^{(i)}} p(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{x}_{j,t-1}^{(i)}) \prod_{j \in \overline{\mathbf{c}_{t-r:t-1}^{(i)}} \cap \widetilde{\mathbf{c}}_t^{(i)}} p_0(\widetilde{\mathbf{x}}_{j,t}^{(i)})}{\prod_{j \in \mathbf{c}_{t-r:t-1}^{(i)}} q(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{x}_{j,t-1}^{(i)}, \mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)}) \prod_{j \in \overline{\mathbf{c}_{t-r:t-1}^{(i)}} \cap \widetilde{\mathbf{c}}_t^{(i)}} q(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)})}$$

    with $\sum_{i=1}^N \widetilde{w}_t^{(i)} = 1$.
  - Compute N_{eff}. If $N_{\text{eff}} \leq \eta$, the duplicate the particles with large weight and remove the particles with small weights, resulting in a new set of particles denotes $\cdot_t^{(i)}$ (without a $\widetilde{\cdot}$) with weights $w_t^{(i)} = 1/N$. Otherwise, rename the particles by removing the $\widetilde{\cdot}$.

At time step t, assume that we have N particles, $\{x_{t-1}^{(i)}\}_{i=1}^{N}$ and their weights, $\{w_{t-1}^{(i)}\}_{i=1}^{N}$ to represent the posterior density. Let us define the state vector of each particle as:

$$x_t^{(i)} = \left[\bar{x}_{1,t}^{(i)} \cdots \bar{x}_{l,t}^{(i)}\right]^T$$

where l is the number of clusters/targets. Firstly the time update of the EKF's are done for each cluster of each target. Let $\mu_{j,t}^{i}$ and $\Sigma_{j,t}^{i}$ represent the mean vector and the covariance matrix of the EKF's. In the prediction step the values are set to:

$$\mu_{j,t}^{i} = F_t x_{j,t-1}^{(i)}, \quad \Sigma_{j,t}^{i} = Q_t$$

The measurement prediction density is approximated by a single Gaussian, $N(g(\mu_{j,t}^{i}), (H\Sigma_{j,t}^{i}H' + R))$, where $g(.)$ is the non-linear function of the state and H is the Jacobian evaluated at $\mu_{j,t}^{(i)}$. After the prediction step is completed, the allocation variable $c_{k,t}$ is sampled from the optimal importance density.

Sampling from optimal importance density: $p(\widetilde{c}_{k,t}|y_{k,t}, \widetilde{c}_{1:k-1,t}^{(i)}, c_{t-r:t-1}^{(i)}, x_{t-1}^{(i)})$

One can factor the optimal importance density as:

$$p(\widetilde{c}_{k,t}|y_{k,t}, \widetilde{c}_{1:k-1,t}^{(i)}, c_{t-r:t-1}^{(i)}, x_{t-1}^{(i)}) \propto p(y_{k,t}|\widetilde{c}_{k,t}, \widetilde{c}_{1:k-1,t}^{(i)}, c_{t-r:t-1}^{(i)}, x_{t-1}^{(i)}) \times p(c_{k,t}|c_{1:k-1,t}, c_{t-r:t-1})$$

Prior is trivial, and the likelihood is:

$$p(y_{k,t}|\widetilde{c}_{k,t}, \widetilde{c}_{1:k-1,t}^{(i)}, c_{t-r:t-1}^{(i)}, x_{t-1}^{(i)}) = N(g(\mu_{j,t}^{(i)}), S_{j,t})$$

where $\mu_{j,t}^{(i)}, S_{j,t}$ are the mean and the innovation covariance calculated by EKF using the previous measurements assigned to that cluster. If c_k is a new cluster, the likelihood is calculated by initiating an EKF with mean x_0 and and covariance P_0. More specifically one can write,

$$\begin{aligned} if \qquad & c_{k,t} \in \overline{(c_{t-r:t-1} \cup c_{1:k-1,t})}, \\ & p(y_{k,t}|\widetilde{c}_{k,t}, \widetilde{c}_{1:k-1,t}^{(i)}, c_{t-r:t-1}^{(i)}, x_{t-1}^{(i)}) = N(g(F_t x_0), [H(F_t P_0 F_t' + Q)H' + R]) \end{aligned}$$

After $c_{k,t}$ is sampled, the EKF of the corresponding cluster is immediately updated by using the measurement $y_{k,t}$.

Sampling from $\hat{p}(x_{j,t}^{(i)}|y_t, x_{j,t-1}^{(i)}, \widetilde{c}_t^{(i)})$

Each sample is generated from the density approximated by the EKF.

$$\hat{p}(x_{j,t}^{(i)}|y_t, x_{j,t-1}^{(i)}, \widetilde{c}_t^{(i)}) = N(\mu_{j,t}^{(i)}, \Sigma_{j,t}^{(i)})$$

Calculating the Measurement Likelihood $\hat{p}(y_{k,t}|c_{k,t}, x_{j,t}^{(i)})$

$$\hat{p}(y_{k,t}|c_{k,t}, x_{j,t}^{(i)}) = N(g(x_{k,t}^{(i)}), R)$$

Calculating the prior $\hat{p}(x_{j,t}^{(i)}|x_{j,t-1}^{(i)})$

$$\hat{p}(x_{j,t}^{(i)}|x_{j,t-1}^{(i)}) = N(g(F_t x_{j,t-1}^{(i)}), Q)$$

After the sampling stage is completed, the weight update is done by using the given equation.

$$\widetilde{w}_t^{(i)} \propto \quad w_{t-1}^{(i)} \frac{\prod_{k=1}^{n} p(\mathbf{y}_{k,t}|\widetilde{c}_{k,t}^{(i)}, \widetilde{\mathbf{x}}_t^{(i)}) \prod_{k=1}^{n} \Pr(\widetilde{\mathbf{c}}_{k,t}^{(i)}|\widetilde{\mathbf{c}}_{1:k-1,t}^{(i)}, \mathbf{c}_{t-r:t-1}^{(i)})}{\prod_{k=1}^{n} q(\widetilde{c}_{k,t}^{(i)}|\mathbf{y}_{k,t}, \widetilde{\mathbf{c}}_{k-1,t}^{(i)}, \mathbf{c}_{t-r:t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)})}$$
$$\frac{\prod_{j \in \mathbf{c}_{t-r:t-1}^{(i)}} p(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{x}_{j,t-1}^{(i)}) \prod_{j \in \overline{\mathbf{c}_{t-r:t-1}^{(i)}} \cap \widetilde{\mathbf{c}}_t^{(i)}} p_0(\widetilde{\mathbf{x}}_{j,t}^{(i)})}{\prod_{j \in \mathbf{c}_{t-r:t-1}^{(i)}} q(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{x}_{j,t-1}^{(i)}, \mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)}) \prod_{j \in \overline{\mathbf{c}_{t-r:t-1}^{(i)}} \cap \widetilde{\mathbf{c}}_t^{(i)}} q(\widetilde{\mathbf{x}}_{j,t}^{(i)}|\mathbf{y}_t, \widetilde{\mathbf{c}}_t^{(i)})}$$

In the last step, effective sample size is calculated and resampling is done if necessary.

VITA

PERSONAL INFORMATION

Surname, Name: Özkan, Emre

Nationality: Turkish (TC)

Date and Place of Birth: 9 May 1980 , Ankara

Marital Status: Single

Phone: +90 533 648 79 25

email: fenerforever@gmail.com

EDUCATION

Degree	Institution; Year of Graduation
BS	METU, Electrical and Electronics Engineering; 2002
High School	Ankara Atatürk Anatolian High School; 1998

PROFFESSIONAL EXPERIENCE

Year	Institution; Status
2002–2007	Department of Electrical and Electronics Engineering, METU; Research and Teaching Assistant
2007–2008	University of British Columbia; Visiting Researcher