

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

**SEQUENTIAL MONTE CARLO SAMPLERS FOR NONPARAMETRIC
BAYESIAN MIXTURE MODELS**

Ph.D. Thesis by

Yener ÜLKER

Department of Electronics and Communications Engineering

Telecommunications Engineering Programme

MARCH 2012

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

**SEQUENTIAL MONTE CARLO SAMPLERS FOR NONPARAMETRIC
BAYESIAN MIXTURE MODELS**

Ph.D. Thesis by

Yener ÜLKER

(504032309)

Department of Electronics and Communications Engineering

Telecommunications Engineering Programme

Thesis Advisor: Prof. Dr. Bilge GÜNSEL

MARCH 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**PARAMETRİK OLMAYAN BAYESÇİ KARIŞIM MODELLERİ İÇİN
ARDIŞIK MONTE CARLO ÖRNEKLEYİCİLER**

DOKTORA TEZİ

Yener ÜLKER

(504032309)

Elektronik ve Haberleşme Mühendisliği Anabilimdalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Bilge GÜNSEL

MART 2012

Yener ÜLKER, a Ph.D. student of ITU Institute of Graduate School of Science Engineering and Technology student ID 504032309, successfully defended the dissertation entitled "SEQUENTIAL MONTE CARLO SAMPLERS FOR NONPARAMETRIC BAYESIAN MIXTURE MODELS", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Bilge GUNSEL**
Istanbul Technical University

Jury Members : **Prof. Dr. Ethem ALPAYDIN**
Boğaziçi University

Prof. Dr. Ertuğrul ÇELEBİ
Istanbul Technical University

Prof. Dr. Hakan Ali ÇIRPAN
Istanbul Technical University

Yrd. Doç. Dr. Ali Taylan CEMGİL
Boğaziçi University

Date of Submission : **13 January 2012**

Date of Defense : **20 March 2012**

FOREWORD

I thank my advisor Prof. Dr. Bilge Günsel for her guidance. Further, I wish to thank the members of my thesis evaluation committee, Prof. Dr. Ethem Alpaydın, Prof. Dr. Ertuğrul Çelebi and Prof. Dr. Hakan Ali Çırpan for their suggestions, helpful discussions and evaluation.

I also thank Assist. Prof. Dr. A. Taylan Cemgil for his invaluable support and guidance throughout my research. I had greatly benefited from his classes and his reading group meetings on statistics. I deeply appreciate his time spent reading, editing and discussing my publications.

Finally, I thank my parents for their caring, understanding, and unconditional support at all times. I dedicate this thesis to them.

February 2012

Yener ÜLKER

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	vii
TABLE OF CONTENTS	x
ABBREVIATIONS	xi
LIST OF TABLES	xiii
LIST OF FIGURES	xvi
SUMMARY	xvii
ÖZET	xix
1. INTRODUCTION	1
1.1 Problem Statement	2
1.1.1 Maneuvering target tracking	2
1.1.2 Sequential inference for Nonparametric Bayesian mixture models	4
1.2 Thesis Organization	6
2. SEQUENTIAL MODELS AND SEQUENTIAL BAYESIAN INFERENCE	9
2.1 Sequential Bayesian Estimation	9
2.2 Bayesian Optimal Filters: Kalman Filtering	10
2.2.1 Extended Kalman filter	12
2.3 Monte Carlo Integration	13
2.4 Importance Sampling	15
2.5 Sequential Importance Sampling (SIS)	16
2.5.1 Degeneracy problem	18
2.5.2 Importance density	18
2.5.3 Resampling	19
2.5.4 Regularization	20
2.6 Sequential Monte Carlo (SMC) Samplers	22
2.6.1 Backward kernels	25
2.6.2 Forward kernels	26
2.7 Dirichlet Process Mixture Models	27
2.8 Models and Algorithms for Maneuvering Target Tracking	29
2.8.1 Target dynamic motion models	29
2.8.1.1 Non maneuvering motion model	30
2.8.1.2 Constant acceleration model (CA)	31
2.8.1.3 Constant turn model (CT)	32
2.8.1.4 Curvilinear motion models (CL)	33
2.8.2 Interacting multiple models (IMM)	34
2.8.3 Multiple model particle filter (MMPF)	36
2.8.4 Jump Markov system particle filter (JMS-PF)	37
2.8.5 Variable rate particle filters	38

2.8.5.1	Variable rate models	39
2.8.5.2	Variable rate state estimation using particle filters	41
2.8.5.3	Variable rate particle filtering algorithm	43
3.	MULTIPLE MODEL TARGET TRACKING WITH VARIABLE RATE PARTICLE FILTERS	47
3.1	Variable Rate Tracking	50
3.2	Multiple Model Variable Rate Model	51
3.3	Multiple Model Variable Rate State Estimation	54
3.4	Regularization	57
3.4.1	Regularized variable rate particle filters	58
3.4.2	Regularized multiple model variable rate particle filter	60
3.5	Experimental Results	61
4.	SEQUENTIAL MONTE CARLO SAMPLERS FOR DIRICHLET PROCESS MIXTURE MODELS	75
4.1	Revisiting the DPM Model and SCM samplers	76
4.1.1	A generic SMC sampler	77
4.2	A SMC sampler for the Dirichlet Process Mixtures	78
4.2.1	MCMC kernels	80
4.2.2	Sequential approximation	81
4.2.3	Algorithmic details	84
4.3	Experimental Results	85
5.	ANNEALED SMC SAMPLERS FOR NONPARAMETRIC BAYESIAN MIXTURE MODELS	89
5.1	SMC Samplers for DPM Models	90
5.2	Annealed SMC Samplers for DPM Models	91
5.2.1	The annealing parameter	92
5.2.2	Multivariate conjugate prior selection for the DPM model	93
5.3	Test Results	95
5.3.1	Univariate density estimation problem	95
5.3.2	Multivariate density estimation problem	98
6.	DIRICHLET PROCESS MIXTURES FOR TIME SERIES CLUSTERING	101
6.1	Model Construction	102
6.2	Sequential Monte Carlo Sampler for the Proposed Model	104
6.2.1	Kernel selection	105
6.2.2	Weight update function	106
6.3	An application: Clustering Mixture of Markov Chains	107
6.4	Experimental Results	108
6.4.1	Clustering audio signal	114
7.	CONCLUSION	117
7.1	Summary and Contributions	117
7.2	Future Work	119
7.2.1	Infinite dimensional hidden semi-Markov models	119
7.2.2	Dirichlet process mixtures for non-linear dynamic systems	119
	REFERENCES	121
	CIRRICULUM VITAE	127

LIST OF ABBREVIATIONS

APF	:	Auxiliary Particle Filter
CA	:	Constant Acceleration Model
CT	:	Constant Turn Model
CL	:	Curvilinear Model
CV	:	Constant Velocity Model
DPM	:	Dirichlet Process Mixtures
EKF	:	Extended Kalman Filter
GMM	:	Gaussian Mixture Model
HMM	:	Hidden Markov Model
IMM	:	Interacting Multiple Model
IMM-EKF	:	Interacting Multiple Model Extended Kalman Filter
IS	:	Importance Sampling
JMS-PF	:	Jump Markov System Particle Filter
ESS	:	Effective Sample Size
MCMC	:	Markov Chain Monte Carlo
MM	:	Multiple Model
MM-PF	:	Multiple Model Particle Filter
MM-VRPF	:	Multiple Model Variable Rate Particle Filter
RJ-MCMC	:	Reversible Jump Monte Carlo Markov Chain
RM	:	Resample and Move
SMC	:	Sequential Monte Carlo
R-PF	:	Regularized Particle Filter
VRPF	:	Variable Rate Particle Filter
SIS	:	Sequential Importance Sampling
SIR	:	Sequential Importance Resampling
MM-VRPF	:	Multiple Model Variable Rate Particle Filter
RMM-VRPF	:	Regularized Multiple Model Variable Rate Particle Filter

LIST OF TABLES

	<u>Page</u>
Table 3.1: MM-VRPF and VRPF parameters for the bearing-only scenarios.	65
Table 3.2: RMSE for varying particle size obtained by using true initials for the Scenario-1.	68
Table 3.3: RMSE for varying particle size obtained by using true initials for the Scenario-2.	71
Table 3.4: RMSE for varying particle size obtained by using erroneous initials for the Scenario-1.	73
Table 3.5: RMSE for varying particle size obtained by using erroneous initials for the Scenario-2.	73
Table 4.1: Mixture model parameters.	86
Table 4.2: Estimated average Log-marginal likelihoods, mean values and respective estimation variances (in parenthesis) for SMC-1, SMC-2, SMC-3, PF and GS.	87
Table 4.3: Estimated average Log-marginal likelihoods, mean values and respective estimation variances (in parenthesis) for SMC-1, SMC-2, SMC-3, PF and GS.	88
Table 5.1: Mixture model parameters.	95
Table 5.2: Estimated average Log-marginal likelihoods, mean values and respective Monte Carlo standard errors (in parenthesis).	97
Table 5.3: Estimated average Log-marginal likelihoods, mean values and respective estimation variances (in parenthesis).	99
Table 5.4: Estimated average Log-marginal likelihoods, mean values and estimation variance (in parenthesis).	100

LIST OF FIGURES

	<u>Page</u>
Figure 1.1: Localization of states for fixed rate (left side) and the variable rate models.	3
Figure 1.2: Observations (red dots) on 2-D space, generated from a Gaussian mixture density with three components. 50% confidence intervals of the Estimated mixture densities by the (a) Proposed SMC algorithm, (b) Conventional particle filtering algorithms.	6
Figure 3.1: (a) Target and observer trajectories for the scenario-1. Trajectories and states of a particle generated by (b) the MM-VRPF, and (c) the VRPF.	62
Figure 3.2: Target and observer trajectories for the scenario-2.	63
Figure 3.3: RMSE _t versus time <i>t</i> for true initials ($N_p = 2000$).	66
Figure 3.4: RMSE _t versus time <i>t</i> for true initials ($N_p = 8000$).	67
Figure 3.5: Posterior distribution of number of states $p(\mathcal{N}_t^+ \mathbf{y}_{0:t})$ where $t = 40$	67
Figure 3.6: RMSE _t versus time <i>t</i> for true initials ($N_p = 2000$).	69
Figure 3.7: RMSE _t versus time <i>t</i> for true initials ($N_p = 8000$).	69
Figure 3.8: RMSE _t versus time <i>t</i> for erroneous initial conditions ($N_p = 2000$).	70
Figure 3.9: RMSE _t versus time <i>t</i> for erroneous initial conditions ($N_p = 8000$).	70
Figure 3.10: RMSE _t versus time <i>t</i> for erroneous initial conditions ($N_p = 2000$).	72
Figure 3.11: RMSE _t versus time <i>t</i> for erroneous initial conditions ($N_p = 8000$).	72
Figure 4.1: Estimated mixture densities by the (a) SMC-1 and (b) SMC-3 algorithm for 50 Monte Carlo runs.	86
Figure 5.1: Estimated mixture densities by the (a) PF, (b) SMC-G, (c) SMC-A (d) SMC-M algorithm for 50 Monte Carlo runs. SMC-A and SMC-M represent all tree components of the mixture density in all runs.	96
Figure 5.2: Observations (red dots) on 2-D space and 50% confidence intervals of the Estimated mixture densities by the (a) SMC-G, (b) SMC-MA algorithms for a single Monte Carlo run. SMC-MA represent all tree components of the mixture density.	99
Figure 6.1: (a) Discrete time series generated from a mixture of Markov Chains (b) Switching labels that select the active Markov chain.	101
Figure 6.2: (a) Synthetically generated data and (b) associated cluster labels.	109
Figure 6.3: (a) True cluster labels versus observation index. (b) Estimated cluster labels versus observation index.	110
Figure 6.4: Normalized expected latency versus changepoint location estimation error.	111
Figure 6.5: (a) Synthetically generated data and (b) associated cluster labels.	112

Figure 6.6:	(a) True cluster labels versus observation index. (b) Estimated cluster labels versus observation index.	113
Figure 6.7:	Normalized expected latency versus changepoint location estimation error.	114
Figure 6.8:	(a) 44100 kHz audio signal. (b) Mel-frequency cepstral coefficients (MFCCs). (c) MFCC coefficients digitized by using the k-means algorithm (Input data).	115
Figure 6.9:	(a) Estimated clustering labels versus observation index. (b) Clustered digitized MFCC coefficients (c) Clustered audio signal.	115

SEQUENTIAL MONTE CARLO SAMPLERS FOR NONPARAMETRIC BAYESIAN MIXTURE MODELS

SUMMARY

This thesis deals with the Bayesian model construction and the inference problem by using Sequential Monte Carlo (SMC) methods. SMC based methods have been the most promising approach among the recursive numerical Bayesian techniques in the latest decade. The key idea of the SMC sampler is to estimate the desired posterior distribution by a set of random samples and associated weights that compute estimates based on these weights and samples. As the number of samples approaches infinity, equivalent representation to the usual functional description of the posterior distribution converges to the optimal Bayesian filter. However due to the computational complexity it is crucial to design efficient samplers that are able to represent the true posterior distribution with a reasonable computational load. In this research our aim is to develop efficient SMC methods for posterior inference and design new probabilistic models that characterize the engineering problems such as target tracking.

First part of the thesis focuses on recently introduced variable rate particle filter (VRPF) that achieves to track the maneuvering objects with a small number of states by imposing a probability distribution on the state arrival times. The variable rate models represent the target dynamics with a single motion model that hinders the capability of estimating maneuver parameters as well as the state arrival times precisely. To overcome this weakness we have incorporated multiple model approach with the variable rate model structure. The introduced model, referred as multiple model variable rate particle filter (MM-VRPF), utilizes a parsimonious representation for smooth regions of trajectory while it adaptively locates frequent state points at high maneuvering regions, resulting in a much more accurate tracking compared to conventional methods.

Next, we deal with the sequential inference problem in Dirichlet process mixtures model (DPM) which is one of the well known nonparametric Bayesian approach to the model selection problem. We developed a novel online algorithm based on the sequential Monte Carlo samplers framework for posterior inference in DPM models. Our method generalizes many sequential importance sampling approaches based on particle filtering. The proposed method enables us to design sophisticated clustering update schemes, such as updating past trajectories of the particles in light of recent observations, and still ensures convergence to the true DPM posterior distribution asymptotically. It provides a computationally efficient improvement to particle filtering that is less prone to getting trapped in isolated modes of the target posterior distribution. Performance improvement over conventional models has been illustrated for Bayesian infinite Gaussian mixture density estimation problem in terms of estimation variance, average log-marginal likelihood and classification accuracy.

In the final section of the thesis, we proposed a novel model for time series clustering problem based on the DPM model structure by using the semi-Markov model formalism. The proposed model is able to estimate the number of clusters, parameters and the sojourn times representing the time series data under a Bayesian framework. We devised a sampling algorithm for sequential inference in the proposed model that also enable us to handle large datasets efficiently. We applied the proposed model to the Markov chain clustering problem and the experimental results showed that the algorithm is able to successfully cluster both synthetic and the real audio data even for large dataset sizes.

PARAMETRİK OLMAYAN BAYESÇİ KARIŞIM MODELLERİ İÇİN ARDIŞIK MONTE CARLO ÖRNEKLEYİCİLER

ÖZET

Bu tezde, son dönemde en ilgi çeken numerik Bayesçi tekniklerden biri olan ardışık Monte Carlo (SMC) metodları ile parametre kestirimi ve model seçimi problemlerinin çözümü üzerine çalışılmıştır. Ardışık Monte Carlo örnekleyicilerin ana fikri, hedef sonsal olasılık dağılımını rastgele örnekler ve ilişkilendirilmiş ağırlıkları ile temsil etmek ve kestirimleri bu örnek ve ağırlıkları kullanarak hesaplamaktır. Hedef sonsal dağılımını ifade eden örnek sayısı arttırıldıkça kestirilen dağılımın fonksiyonel ifadesi optimum Bayesçi filtreye yaklaşacaktır. Ancak limitli hesap gücü nedeniyle, kabul edilebilir bir işlemsel karmaşıklıkta gerçek sonsal dağılımı başarılı kestirebilen etkin örnekleyicilerin tasarlanması çok önemlidir. Bu çalışmanın temel amacı etkin SMC tabanlı algoritmalar tasarlamak ve gerçek hayat mühendislik problemlerini ifade edebilen olasılıksal modeller üzerinde bu metodları uygulamaktır. Tezde, hedef takibi problemlerinin ve Dirichlet süreci karışım modellerinin önerilen yenilikçi model ve algoritmalar ile çözümü üzerine çalışılmıştır.

Hedef takibi probleminde amaç, hedeften alınan gürültülü ölçümlerden hedef kinematiklerini (konum, hız ve bnz.) kestirmek ve hedefi anlık olarak takip etmektir. Kullanılan en yaygın çözüm tekniklerinin başında hedef durum ve gözlemlerini birer rastgele değişken dizisi ile modelleyen ve değişkenler arası tanımlanan ilişkidenden faydalanarak hedef durumlarını kestirmeyi amaçlayan olasılıksal modeller gelmektedir. Bu modellerin çözümünde, hedef hareketlerinin anlık izlenebilmesi için, hedef durum değişkenlerini her yeni gözlem alındığı zaman kestirebilen algoritmalara ihtiyaç vardır. Kalman süzgeçleri, genişletilmiş Kalman süzgeçleri ve parçacık filtreleri hedef takibi probleminin çevrimiçi çözümü için en sık kullanılan kestirim metodlarının başında gelmektedir. Tüm bu algoritmalar durum uzay denklemleriyle tanımlanmış, sabit oranlı bir Markov modelinde hedef kinematiklerini kestirmeyi amaçlamaktadır. Ancak özellikle manevra yapan hedeflerin takibi probleminde hedef hareket parametrelerinin takip öncesi bilinmemesi yüksek manevra kabiliyetine sahip hedeflerin takibinde kestirim başarımını düşürmektedir. Bu nedenle hedefe ilişkin farklı hareket rejimlerini tek bir modelde birleştirerek takip edilen hedefin hareket karakteristiğine en uygun hareket rejimini adaptif olarak seçen çoklu model yaklaşımları literatürde kullanılmıştır.

Bu tezde, son yıllarda manevralı hedeflerin takibinde ortaya atılan en yenilikçi modellerden biri olan değişken oranlı parçacık süzgeçleri (VRPF) ele alınmıştır. Hedef takibi literatürdeki son gelişmeler, yarı-Markov modellerin sabit oranlı Markov model yapısına dayalı klasik süzgeçlere iyi bir alternatif olabileceğini göstermektedir. Markov model yapısındaki durum uzay gösterimlerinde durumlar ve gözlemlerin

aynı anda olduğu yani durum varış zamanlarının sabit olduğu varsayılır. Yarı-Markov model yapısında ise durum varış zamanları gözlem zamanlarından bağımsız varsayılarak bir rastgele değişken olarak tanımlanır ve sabit oranlı durum uzay gösterilimine göre çok daha az sayıda durumla farklı manevra karakteristiklerinin modellenebilmesi amaçlanır. Daha az durum sayısı ile gezinenin modellenebilmesi ve durum varış zamanlarının bir dağılımla ifade edilmesi, işletilen çözüm algoritmasının etkinliğini arttırmakta, hedef hareketinin sürekli zamanda hedef hareket yapısının doğasına daha uygun, az sayıda parametre ile modellenebilmesini sağlamaktadır.

Bilinmektedir ki, özellikle insanlı hedefler sert ve kısa zamanlı manevralar ile yeni rotasına ulaştıktan sonra yeni hedef noktasına ulaşmak için uzun süreli düz hareketler yapmaktadır. Ancak, değişken oranlı modelin tek bir hedef hareket modeli kullanması, manevra parametreleri ve varış zamanları üzerine sadece tek bir önsel olasılık dağılımı tanımlanabilmesine olanak vermekte ve farklı hareket koşullarının göz önünde bulundurulması mümkün olmamaktadır. Biz çalışmamızda çoklu model yaklaşımlarını değişken oranlı modeller ile birleştirerek, varış zamanları ve manevra parametrelerini adaptif bir yapı ile kestirebilen çoklu model değişken oranlı parçacık süzgeçlerini önerdik. Önerilen model, adaptif olarak, manevra anlarını sık, düz gezinmeleri ise az sayıda durum ile ifade etmektedir. Önerilen model kerterizden hedef takibi senaryolarda test edilmiş ve literatürde önerilen modellerden daha yüksek takip başarımına sahip olduğu karşılaştırmalı olarak gösterilmiştir.

Sonrasında, tezin literatüre temel katkısı olarak da nitelendirilebilecek, parametrik olmayan Bayesçi model seçim tekniği Dirichlet süreci karışım modellerinin ardışık Monte Carlo metodları ile çözümü üzerinde çalışmalarımızı yoğunlaştırdık. Dirichlet süreci karışım modelleri (DPM), verinin parametrik bir aile ile kısıtlı olmadığı durumlarda, olasılık dağılım fonksiyonları ve sınıflandırma problemlerinin hiyerarşik modellenebilmesi için temel çözüm bloğu olmuştur. DPM modelin prensipte etkin olarak çözülebildiği varsayılırsa herhangi bir olasılık dağılımını istenen doğrulukta modelliyebilme yeteneğine sahiptir. Ancak, model sonsalının analitik çözümü bulunmaması nedeniyle model sonsal dağılımının çeşitli yaklaşık çözümlerle kestirilmesi önemli ve popüler bir araştırma alanı olmuştur. Bu amaçla literatürde, çıkarsamayı bütün veriseti üzerinde gerçekleştiren yığın algoritmaları önerilmekle beraber, gerçek hayatta birçok problemin anlık çözüme ihtiyaç duyması yada veri setlerinin çok büyük olması, yığın algoritmaların çözüme çok uzun sürede ulaşmalarına neden olmuş ve çevrim içi algoritmalara olan ilgiyi arttırmıştır. Ancak literatürde parçacık filtresi altyapısına dayalı olarak önerilen çevrimiçi algoritmalarının sadece küçük veri setlerinde yüksek başarımla gösterdiği, orta ve büyük veri setlerinde ise çok kısıtlı başarımla sahip olduğu gözlenmiştir. Bunun en temel sebebi olarak parçacık filtrelerinin yeni gözlemler geldikçe parçacık geçmişini güncelleyememeleri, dolayısıyla algoritmanın lokal minimumlara takılması ve zamanla toplamsal Monte Carlo Hatasının artması olarak gösterilebilir. Ayrıca parçacık filtrelerinde kullanılan tekrar örnekleme adımında düşük ağırlıklı parçacıkların göz ardı edilmesi ve bu parçacıkların tekrar değerlendirilememesi başarımla düşüren bir diğer faktördür. Parçacık filtresi tabanlı algoritmaların zayıflıklarını gidermek amacıyla, biz çalışmamızda DPM sonsal dağılımını ardışık Monte Carlo örnekleyicileri çatısı altında etkin olarak kestirebilen yenilikçi çevrimiçi çıkarsama algoritmaları tasarladık. Önerilen metod yenilikçi yapısının yanı sıra, DPM için kullanılan diğer ardışık örnekleyicileri genelleme özelliğine de sahiptir. Kullanılan

yöntem, yeni gelen gözlemlerin ışığında örneklere ilişkin geçmiş gezinmeleri güncelleyerek gerçek DPM sonsal dağılımına daha iyi yaklaşıklık sağlayan etkin örnekler elde etmekte ve kestirim başarımını arttırmaktadır. Öneri dağılımlarının tasarımında tavlama tekniği kullanılarak karışım özelliği yüksek olan öneri dağılımları elde edilmiş ve tasarlanan yapılar farklı sınıf etiketi güncelleme metodlarıyla beraber kullanılmıştır. Kullanılan tavlama tekniği özellikle yığın algoritmalarında kullanılan klasik tavlama tekniğinden farklı olarak her yeni gözlem adımında çevrimiçi kestirim sonucuna ulaşılabilmesini sağlamaktadır.

Önerilen algoritmalar parçacık süzgeçleri, Gibbs tabanlı parçacık filtreleri ve Gibbs algoritması ile tek ve çok boyutlu yoğunluk dağılımı kestirim problemlerinde karşılaştırılmıştır. Test amaçlı olarak sentetik veri setlerinin yanısıra ses verilerinden ön işleme adımlarıyla elde edilen gerçek hayat verileri de kullanılmıştır. Önerilen metodun, özellikle sonsal dağılımın izole modlara sahip olduğu durumlarda klasik metodlara göre çok daha düşük kestirim sapmasına sahip olduğu, lokal çözümlere daha düşük olasılıkla takıldığı ve dolayısıyla sonuca daha yüksek doğrulukta yakınsayabildiği görülmüştür.

Tezin son bölümünde yarı-Markov model yapısı ve Dirichlet karışım modelleri tek bir çatı altında toplanarak zaman serilerini sınıflandırma problemi için yenilikçi parametrik olmayan bir model önerilmiştir. Problemden, zaman serisini tanımlayan sınıf sayısı, sınıf parametreleri ve parametre değişim zamanları birer rastgele değişkenler dizisi olarak tanımlanmıştır. Amaç, zaman serisini tanımlayan sınıf sayısını, parametreleri ve sınıfların aktif olduğu zamanları Bayesçi bir çatı altında kestirebilmektedir. Tanımlanan yarı-Markov model yapısı, her bir gözlemin bir durum değişkeni ile ilişkilendirildiği sabit oranlı modellerden farklı olarak durum varış zamanlarının Markov sürecine uyduğunu varsayarak sınıf parametrelerinin değiştiği anları kestirmeyi hedeflemektedir. Durum varış zamanlarının Markov yapıda bir değişken dizisi olarak kabul edilmesi sürekli yada ayrık zamandaki verinin doğal yapısının daha iyi modellenmesini ve problemin sabit oranlı modellere göre çok daha az sayıda durum değişkeni ile düşük karmaşıklıkla ifade edilebilmesini sağlamaktadır. Tasarlanan modelde, ardışık durum varış zamanları arası sürenin herhangi bir dağılımla ifade edilebilmesi, probleme ilişkin önsel bilginin çok daha etkin kullanılabilmesini sağlamaktadır. Etkin bir çıkarsama algoritmasının tasarlanabilmesi için model parametreleri üzerinde tanımlanan önsel dağılım olabirlik fonksiyonuna konjuge seçilmiştir. Problemin yapısı itibarı ile çevrimiçi çözüm bulunması özellikle parametre değişim zamanlarının kestirimi için çok önemlidir. Bu nedenle hesaplanabilir çözümü bulunmayan modelin sonsal dağılımı ardışık Monte Carlo tekniği kullanılarak kestirilmiştir. Önerilen model, sürekli yada ayrık zamanda tanımlanan tüm zaman serilerine uygulanabilir niteliktedir. Tasarlanan model, Markov zincirlerinin sınıflandırılması problemine uyarlanmış, n durumlu bir zaman serisini oluşturan Markov zinciri sayısını, zincirler arasındaki geçiş zamanlarını ve her bir zincirin parametrelerini kestiren bir algoritma gerçekleştirilmiştir. Sentetik veri setleri üzerinde yapılan testlerde parametre kestirim performansı raporlanmıştır. Network analizi ve bioinformatik gibi konularda önemli bir uygulama alanı bulunan problemin gerçek hayat verilerindeki başarımı is ses sinyalleri üzerinde gösterilmiştir. Algoritmanın sentetik verileri ve ses sinyalini başarılı olarak sınıflandırdığı ve büyük veri setlerinde başarıyla çalıştığı gözlemlenmiştir.

1. INTRODUCTION

In applied and engineering problems involving analysis of real world datasets, researchers are often faced with two fundamental and challenging questions about model construction: what model class and model order to choose, and how to estimate optimal parameters to predict future observations. Parameter estimation is perhaps the better understood problem as there is an arsenal of optimization and integration techniques available. The problem of model construction is more obscure and considered more art than science. Even if a model class is chosen, it is still far from obvious what the model order should be. Typically if an overly simple model is chosen, the predictions are poor. Similarly, if an overly complex model is chosen, the predictions can be equivalently poor due to the so called 'overfitting' problem. While regularization techniques or full Bayesian treatment can remedy the latter, this may still lead to unnecessary waste of computational resources. It would be desirable to develop highly adaptive methods, that would tune their complexity and parameters without resorting to a fixed parametric family.

Modern computational techniques based on Monte Carlo simulation, provide a practical solution to this key problem. Intuitively, these techniques are able to adapt both model strength and computational cost, hence such methods potentially provide significant advantages in real life applications. However, batch methods are computationally infeasible for inference in dynamic models or when large datasets has to be processed. Therefore, in contrast to the batch methods, that apply the inference on the entire dataset, sequential Monte Carlo (SMC) methods recursively updates the posterior estimate upon arrival of each new observation. SMC provides a computationally efficient solution that especially suits real-time processing requirements.

1.1 Problem Statement

In our research we deal with sequential models and algorithms for sequential Bayesian Inference. First we focus on model construction and parameter estimation for tracking a maneuvering object under nonlinear and non-Gaussian conditions. Next, we deal with the sequential Bayesian estimation problem in Dirichlet process mixtures (DPM) model that has been one of the most widely used and popular approach to nonparametric probabilistic models (Antoniak, 1974). For both problems we utilize sequential Monte Carlo methods that allow us to treat any type of probability distribution and nonlinearity.

1.1.1 Maneuvering target tracking

In a target tracking problem, the aim is to estimate the target location and motion parameters accurately with the help of received noisy observations. Most widely used and promising approaches to tracking problem are based on stochastic filtering theory that was first established in early 1940's due to the pioneering work by Wiener (1949) and Kolmogorov (1941), and it culminated in 1960 for the publication of classic Kalman Filter (Kalman, 1960).

The target tracking problem faces two interrelated main challenges , these are target motion-mode uncertainty and nonlinearity (Li and Jilkov., 2005). Nonlinearity is handled by nonlinear filtering methods, and Multiple-model (MM) methods have been generally considered as the mainstream approach to the maneuvering target tracking under motion-mode uncertainty (Blom et al., 1998; McGinnity and Irwin, 2000; Doucet et al., 2001b; Kirubarajan et al., 2001). The performance of the tracking algorithms hinges in the modeling capability of the tracking model and the quality of the inference schemes in solving the model.

Conventional probabilistic tracking models define a discrete time state space model where the state sampling rate is determined by the rate at which the measurements arrive, thus known as fixed rate models. In fixed rate models, the time between two consecutive states (sojourn time) is fixed and the timing of the state variables (state arrival time) are determined under the assumption that a transition may occur at each observation time (Blom et al., 1998; McGinnity and Irwin, 2000;

Godsill and Vermaak, 2005). However, manned targets commonly execute short duration of sharp maneuvers following prolonged times of a smooth trajectories. Therefore, they have limited capability in modeling the sojourn times as well as the target parameters in a maneuvering target tracking problem. Unlike the fixed rate standard tracking models, recently introduced variable rate particle filters (VRPF), which models the state arrival times as a Markovian random process, enables the time between consecutive target states to be a random variable, hence can be considered as a more effective technique in target tracking (Godsill and Vermaak, 2005; Godsill et al., 2007).

The variable rate model tracks a maneuvering object with a small number of states by imposing a probability distribution on the sojourn times. In Figure 1.1 we show the localization of the states for fixed and variable rate models on a bearing-only target tracking scenario. The variable rate model (left figure) is much more efficient compared to a fixed rate model (right figure) since variable rate scheme allocates more state points to regions of rapid deviations and fewer points to smooth trajectories. However conventional variable rate models utilize a single motion model in order to characterize the state arrival times and the target parameters (Godsill and Vermaak, 2005). This limits the capability of estimating the maneuvering and smooth regions of the trajectory precisely. Therefore variable rate algorithms suffer from the poor estimate of the target parameters and the state arrival times.

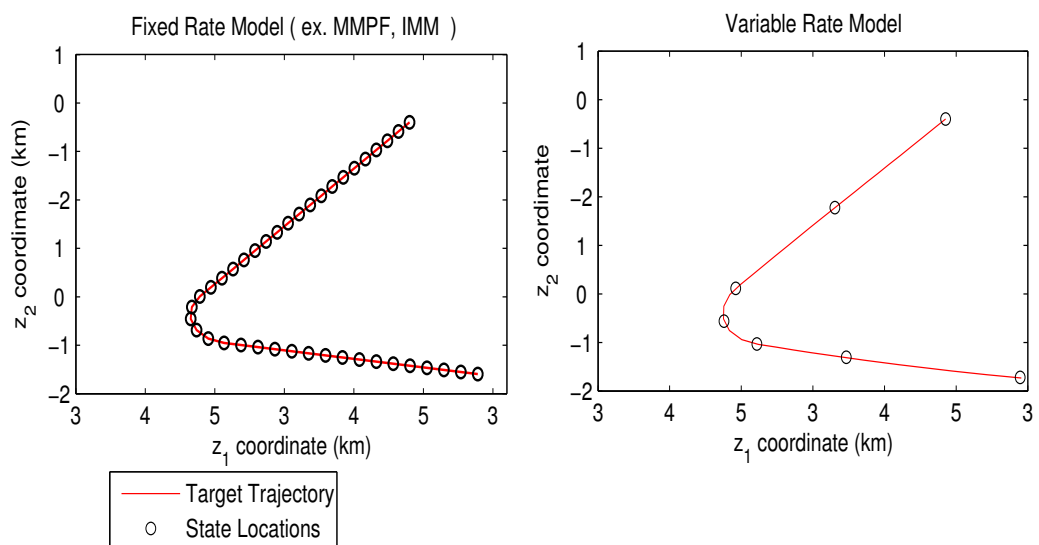


Figure 1.1: Localization of states for fixed rate (left side) and the variable rate models.

In order to overcome this drawback, in this thesis we developed a new model structure for variable rate models that aims to achieve a better characterization of the state arrival times and the target trajectory. We investigated adaptive tracking methods based on the multiple model approach and developed a multiple model structure for the variable rate models (Ulker et al., 2008; Ulker and Gonsel, 2012). The proposed model allows switching between candidate sojourn and motion parameter sets thus can precisely model the maneuver parameters as well as the state arrival times. Sequential inference for the proposed model is accomplished by the particle filtering algorithm hence named as multiple model variable rate particle filter (MM-VRPF). We evaluated the performance of the algorithm on several bearing only target tracking scenarios and improvements over conventional algorithms are reported.

In this thesis we focused on model construction problem for the variable rate models, therefore we applied conventional particle filtering for sequential inference. However, more efficient inference schemes based on sequential Monte Carlo samplers framework has been proposed by Whiteley et al. (2007) for variable rate models and it is possible to achieve improved results by applying similar sampling strategies to our model.

1.1.2 Sequential inference for Nonparametric Bayesian mixture models

Recently, Dirichlet process mixtures (DPM) have been widely used as a building block in hierarchical models for solving density estimation and clustering problems where the actual form of the data generation process is not constrained to a particular parametric family (Antoniak, 1974). Sophisticated applications involving DPM models has been studied in machine learning, signal processing, tracking or bioinformatics (Teh et al., 2004; Do et al., 2005; Caron et al., 2008; Fox et al., 2007). Provided that inference can be carried out effectively for the DPM, at least in principle, any density can be approximated with arbitrary precision. However, exact inference is unfortunately intractable. Yet due to the mentioned potential advantages of nonparametric approaches, there has been a surge of interest to the DPM model and efficient inference strategies based on variational techniques (Blei and Jordan, 2004, 2006) and Monte Carlo Markov Chain (MCMC) (S. Walker and Smith, 1999; MacEachern et al., 1999; Jain and Neal, 2000; Neal, 2000). Though, majority

of these methods perform batch algorithms that apply the inference on the entire dataset (Blei and Jordan, 2004; Neal, 2000), sequential methods that cluster each new observation upon its arrival have also been proposed (Quintana, 1996; MacEachern et al., 1999; Fearnhead, 2004).

Assuming that conjugacy condition is satisfied, methods that solve Bayesian inference problem based on sampling deals with sampling from an intractable discrete distribution. In literature, particle filtering has been proposed as a computationally efficient sequential sampling method that especially suits real-time processing requirements of dynamic models (Fearnhead, 2004). It is shown that particle filtering outperforms batch algorithms such as Gibbs sampler for small datasets (Fearnhead, 2004). However, it is argued that sequential importance sampling is not an appropriate method for models with static parameters and especially large datasets due to the degeneracy phenomenon and accumulated Monte Carlo error over time (Quintana and Newton, 1998). The sampler becomes 'sticky', meaning that previously assigned clusterings can never be updated according to the information provided by the latest observations. For static problems, degeneracy can be reduced but not avoided via resampling techniques (MacEachern et al., 1999; Fearnhead, 2004). Although particle filtering algorithm can even outperform batch algorithms such as the Gibbs sampler for small datasets, empirical evidence suggests that sequential methods achieve less satisfactory results particularly on larger datasets. In a particle filtering framework, this arises mainly due to the fact that discarded particles can never be reconsidered. These limitation causes the algorithms to get trapped in local modes of the posterior distribution and reduce the estimation performance. This is illustrated in Figure 1.2 for a two dimensional model where the data shown with the red dots is generated from a mixture of Gaussian distributions with three components. At the left side of Figure 1.2 we see that the conventional particle filter gets trapped in the local mode of the solution and represents the data with two mixture components whereas the true estimation result is shown at the right.

In this thesis we search for efficient sampling methodologies for posterior inference in DPM models by using sequential Monte Carlo techniques. We propose new sampling strategies in order to estimate the time evolving DPM model posterior. We take the advantaged of using the SMC sampler proposed by Del Moral et al. (2006) that enable

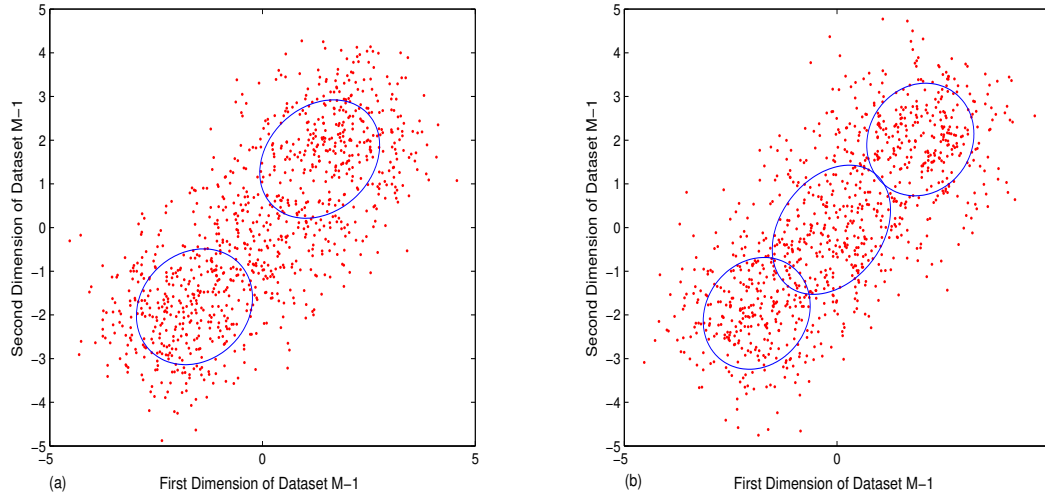


Figure 1.2: Observations (red dots) on 2-D space, generated from a Gaussian mixture density with three components. 50% confidence intervals of the Estimated mixture densities by the (a) Proposed SMC algorithm, (b) Conventional particle filtering algorithms.

us to design sophisticated particle updating schemes and ensures convergence to the true target distribution asymptotically.

1.2 Thesis Organization

The thesis is organized as follows: In Chapter 2, the basics of numerical Bayesian techniques and sequential probabilistic model structures are investigated. Following an introduction to non-parametric model construction, we seek to investigate the problem of maneuvering target tracking and parameter estimation in dynamic systems with numerical Bayesian techniques.

In Chapter 3 we introduce an adaptive tracking method for maneuvering target tracking of which incorporates multiple model approach with the variable rate model structure proposed by Godsill et al. (Godsill and Vermaak, 2005). The proposed model is based on semi-Markovian model structure and referred as multiple model variable rate particle filter (MM-VRPF) (Ulker et al., 2008; Ulker and Gonsel, 2008). MM-VRPF adaptively locates frequent state points to the maneuvering regions resulting in a much more accurate tracking while preserving the parsimonious representation for the smooth regions of the trajectory. This is achieved by including a mode variable into the conventional variable rate state vector which enables us to define a different

sojourn time and motion prior for each target motion mode using the multiple model structure.

Chapter 4 proposes efficient sampling strategies for sampling a time evolving DPM posterior distribution (Ulker et al., 2010b). Sequential inference schemes have limited success in maintaining an accurate approximation to the true target density. Particularly for large datasets, Monte Carlo error accumulates over time and the estimation variance increases (Quintana and Newton, 1998). This is due to the fact that past states of the particle trajectories (i.e., past clusterings) are not updated with new observations. The problem can be alleviated by a retrospective method that is able to reassign the previous clusterings at time n according to latest observations received. The SMC samplers framework enables us to accomplish this in practice and still ensures convergence to the true target posterior asymptotically. Unlike the existing methods proposed by Quintana (1996); MacEachern et al. (1999); Fearnhead (2004), we propose an efficient sequential Monte Carlo sampler that enables us to update past trajectories of the particles in the light of recent observations.

Chapter 5 further improves the method proposed in Chapter 4 by using annealing strategies under the sequential estimation context (Ulker et al., 2010a, 2011). Annealed proposal kernels are defined in order to draw efficient samples from the DPM posterior distribution that prevent the algorithm to get trapped to the local modes. The proposed method takes advantage of the SMC sampler framework in order to calculate the sample weights that ensures convergence to the true DPM posterior distribution. Due to importance of modeling the multidimensional dependencies in high dimensional datasets we extended the proposed algorithm to the multidimensional case. The efficiency of the proposed algorithms is analyzed on several univariate and multivariate synthetic datasets.

In Chapter 6 we develop a novel model for time series clustering, based on the DPM model structure under the semi-Markovian model formalism where the number of clusters and the parameters are priorly unknown. We constructed a Markovian process for modeling the change points in order to define an arbitrarily selected distribution on the sojourn times. We proposed a sequential Monte Carlo algorithm for online inference that can handle large datasets and applied the proposed model to the Markov

Chain clustering problem. We reported the estimation results for both synthetic and real world data.

The thesis is finalized with the conclusions and future research directions given in Chapter 7.

2. SEQUENTIAL MODELS AND SEQUENTIAL BAYESIAN INFERENCE

In this section, we give an introduction to the probabilistic filtering methods and basic numerical Bayesian techniques. After a brief introduction to the optimal filtering, we give the basics of Monte Carlo integration, importance sampling, particle filtering and the sequential Monte Carlo samplers framework that underlies the principles of sequential Bayesian estimation (Del Moral et al., 2006). We also present models and algorithms for mixture models with unknown number of components and maneuvering target tracking (Chen, 2003).

2.1 Sequential Bayesian Estimation

Kalman filters are optimal recursive Bayesian filters giving the Minimum Mean Square error under linear Gaussian conditions (Chen, 2003). However Bayesian techniques require integration of the product of probability density functions that cannot be accomplished in closed form for general nonlinear, non-Gaussian multivariate system. Necessity of linear Gaussian assumptions to execute Kalman filter equations is a highly restrictive constraint in many applications, thus several different approximations are to be introduced in literature such as extended Kalman filters (Chen, 1993), approximate grid based methods (Arulampalam et al., 2002), Monte Carlo Methods (Hanscomb and Hammersley, 1964) and etc. to solve the problem.

Main idea of extended Kalman filter (EKF) is to linearize the nonlinear equations around the prediction mean and assuming the posterior distribution as Gaussian. EKF is well suited for problems which are not highly nonlinear, unfortunately they fail to represent highly nonlinear environments such as multi modal distributions. Grid based method is another approximate Bayesian technique which discretizes the posterior evenly. Discretizing can be implied by constituting finite number of states as posterior. Hidden Markov Models (HMM) are an application of such approximate grid based methods and are used extensively. However grid based methods are faced with the

curse of dimensionality particularly in high dimensional state spaces which render these methods inappropriate for many applications.

Sequential Monte Carlo (SMC) methods, has been the most promising approaches among numerical recursive Bayesian techniques. The key idea is to represent required posterior distribution by set of random samples with associated weights and compute estimates based on these weights and samples. Sequential importance sampling (SIS) (Doucet et al., 2001a), sequential importance resampling (SIR) (Doucet et al., 2001a) are well known sequential Monte Carlo methods commonly named as particle filters. In the last decade several variants of particle filters, auxiliary particle filters (APF) (Pitt and Shephard, 1999), regularized particle filters (RPF) (Musso et al., 2001), Rao-Blackwellized particle filters (Doucet et al., 2001b) and etc. have been proposed. Although they have the capability to represent any posterior distribution, in practice they suffer from the degeneracy problem particularly in high dimensional spaces. Therefore, when sampling from a posterior density, they have limited success in maintaining an accurate approximation to the true target density. For example, a Rao Blackwellized particle filter based approach has been employed by Fearnhead (2004) to approximate the DPM target posterior sequentially as each new observation arrives. However the algorithm is only effective for small datasets due to accumulated Monte Carlo error over time because past states of the particle trajectories (i.e., past clusterings) are not updated with new observations (Quintana and Newton, 1998; Fearnhead, 2004). This problem can be alleviated by a retrospective method that is able to reassign the previous clusterings at time n according to latest observations received. Unfortunately, such a strategy will be intractable under the standard particle filtering framework. Yet, the SMC samplers framework proposed by Del Moral et al. (2006) will be detailed in Section 2.6 that enables us to update the previous clusterings by retrospection and still ensures convergence to the true target posterior asymptotically.

2.2 Bayesian Optimal Filters: Kalman Filtering

Kalman filter is an efficient recursive filter that estimates the state of a dynamical system from a series of incomplete and noisy measurements. The states follow a first-order Markov process and the observations are independent of the given states. Kalman filters, developed by Kalman (1960), are based on linear dynamic systems discretized

in the time domain as in Eq.(2.1),

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \quad (2.1)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k$$

where \mathbf{x}_k is the state vector, \mathbf{F}_k and \mathbf{H}_k are process and measurement matrices defining the linear equations at time index k . In Eq.(2.1), \mathbf{v}_{k-1} , \mathbf{n}_k are Gaussian random variables with covariances \mathbf{Q}_{k-1} and \mathbf{R}_k , respectively. Generally we consider they are zero mean and independent random variables. Since \mathbf{F}_k , \mathbf{H}_k and noise covariance parameters \mathbf{v}_{k-1} , \mathbf{n}_k shown in Eq.(2.1) are allowed to vary with time, Kalman filters can handle non stationary environments.

Kalman Filter consists of an iterative prediction and correction process. In the prediction step, the time update is taken where the one-step ahead prediction of observation is calculated; in the correction step, the measurement update is taken where the correction to the estimate of current state is calculated. Kalman filter assumes that posterior density $p(\mathbf{x}_{1:k}|\mathbf{y}_{1:k})$ is Gaussian hence parameterized with its mean and covariance. Recursive Kalman equations are derived by obtaining $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ from $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ in a recursive relationship. Define;

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) = N(\mathbf{x}_{k-1}; \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \quad (2.2)$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = N(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad (2.3)$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = N(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k}) \quad (2.4)$$

where $N(\mathbf{x}, \mathbf{m}, \mathbf{P})$ is a Gaussian density with mean \mathbf{m} and covariance \mathbf{P} . The prediction step is defined as,

$$\mathbf{m}_{k|k-1} = \mathbf{F}_k \mathbf{m}_{k-1|k-1} \quad (2.5)$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T \quad (2.6)$$

and the correction step is,

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}) \quad (2.7)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \quad (2.8)$$

where covariance of the innovation term $\mathbf{y}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}$ and the Kalman gain are defined as in Eq.(2.9) and Eq.(2.10), respectively.

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (2.9)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (2.10)$$

Kalman filter is optimal if the linear Gaussian assumptions hold true. The implication is that, no algorithm can ever do better than Kalman filter if the model assumptions exactly fit the problem considered. However, posterior is not necessarily Gaussian for nonlinear problems and Kalman filter is then not certain to be optimal. However Bayesian filtering theory is optimal in all conditions independent from the definition of the density functions (linear or nonlinear) concerning the model. In literature many approaches have been introduced to approximate the optimum Bayesian solution under nonlinear conditions. Particle filters and Extended Kalman Filters are good examples of these.

2.2.1 Extended Kalman filter

The use of Kalman filter is limited with linear problem where the posterior distribution is represented with a single Gaussian function. However Kalman filter can't find solution to a nonlinear system defined as in Eq.(2.11),

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \quad (2.11)$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \mathbf{n}_k$$

where $f_k(\cdot)$ and $h_k(\cdot)$ are known nonlinear functions, and \mathbf{v}_{k-1} , \mathbf{n}_k are Gaussian random variables with covariances \mathbf{Q}_{k-1} and \mathbf{R}_k respectively. As a solution, extended Kalman filter (EKF) linearizing the nonlinear functions using series expansion is proposed (Chen, 1993) . EKF approximates the posterior distribution as a Gaussian to handle nonlinear problems such that $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ approximates to Gaussian and following approximations will be valid,

$$p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) \approx N(\mathbf{x}_{k-1}; \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \quad (2.12)$$

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) \approx N(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad (2.13)$$

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx N(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k}) . \quad (2.14)$$

Consequently, recursive EKF equations are defined as,

$$\mathbf{m}_{k|k-1} = \mathbf{F}_k(\mathbf{m}_{k-1|k-1}) \quad (2.15)$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \hat{\mathbf{F}}_k \mathbf{P}_{k-1:k-1} \hat{\mathbf{F}}_k^T \quad (2.16)$$

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k(\mathbf{m}_{k|k-1})) \quad (2.17)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \hat{\mathbf{H}}_k \mathbf{P}_{k|k-1} \quad (2.18)$$

where $\hat{\mathbf{F}}_k$, $\hat{\mathbf{H}}_k$ are local linearizations of the nonlinear functions $f_k()$ and $h_k()$, respectively as defined in Eq.(2.19) and Eq.(2.20).

$$\hat{\mathbf{F}}_k = \left. \frac{df_k(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{m}_{k-1|k-1}} \quad (2.19)$$

$$\hat{\mathbf{H}}_k = \left. \frac{dh_k(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{m}_{k|k-1}} \quad (2.20)$$

Covariance of the innovation term and the Kalman gain are defined as in Eq.(2.21) and Eq.(2.22).

$$\mathbf{S}_k = \hat{\mathbf{H}}_k \mathbf{P}_{k|k-1} \hat{\mathbf{H}}_k^T + \mathbf{R}_k \quad (2.21)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \hat{\mathbf{H}}_k^T \mathbf{S}_k^{-1} \quad (2.22)$$

Because EKF always approximates the posterior $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ as a Gaussian, it works well for some types of nonlinear problems, but it may provide a poor performance in some cases when the true posterior is non-Gaussian (Chen, 2003). Bi-modal or heavily skewed posterior distributions are examples for this situation. In such cases particle filters yield improvement at performance over EKF.

2.3 Monte Carlo Integration

In literature many methods has been introduced to handle nonlinear filtering problem using numerical approximate techniques. Monte Carlo sampling approximation, Gaussian/Laplace approximation, iterative quadrature approximation, moment approximation, deterministic sampling approximation are some of these approaches. In this section we will focus our attention particularly on the sequential Monte Carlo technique which is one of the most promising approach in the field. First we will explain basics on Monte Carlo integration and probabilistic inference schemes using

Monte Carlo techniques, later we will describe more sophisticated sequential filtering schemes.

Monte Carlo methods estimate the integrals or other quantities that can be expressed as an expectation by averaging the results of a high number of statistical trials (Hanscomb and Hammersley, 1964). Computers are ideal for performing such trials, and the appearance of faster computers has driven the wide spread application of Monte Carlo methods today. The core problem is the computation of an integral,

$$\int_x f(x)p(x)dx \quad (2.23)$$

with x a possibly multi-dimensional variable. Suppose we also have a pdf $p(x)$ ($p(x) > 0$) according to which we can draw samples x^i using N_p independent samples, the integral can now be estimated as

$$\hat{f}_{N_p} = \frac{1}{N_p} \sum_{i=1}^{N_p} f(x^i). \quad (2.24)$$

This is the basic N_p sample Monte Carlo estimator using importance sampling with a pdf $p(x)$. Expected value and variance of the estimator can be expressed as,

$$E[\hat{f}_{N_p}] = E[f] \quad (2.25)$$

$$Var[\hat{f}_{N_p}] = \frac{1}{N_p} Var[f] = \frac{\sigma^2}{N_p} \quad (2.26)$$

where σ^2 is the variance of $f(x)$ and Eq.(2.25) indicates that the estimate is unbiased. By the Kolmogorov strong law of large numbers \hat{f}_{N_p} converges to $E[f]$ almost surely and convergence rate assessed by central limit theorem.

$$\sqrt{N_p}(\hat{f}_{N_p} - E[f]) = N(0, \sigma^2) \quad (2.27)$$

One crucial property of Monte Carlo approximation is the estimation accuracy is independent of the dimensionality of the state space, in contrast to most deterministic numerical methods. The variance of estimate is inversely proportional to the number of samples.

2.4 Importance Sampling

Importance sampling (IS) was first introduced by Marshall (Marshall, 1956). The objective of importance sampling is aimed to sample the distribution in the region of “importance” in order to achieve computational efficiency. This is important especially for the high-dimensional space where the data are usually sparse, and the region of interest where the target lies in is relatively small in the whole data space. The idea of importance sampling is to choose a proposal distribution $q(x)$ in place of the true probability distribution $p(x)$, which is hard-to-sample. The support of $q(x)$ is assumed to cover that of $p(x)$. Rewriting the integration in Eq.2.23 as,

$$\int_x f(x) \frac{p(x)}{q(x)} q(x) dx. \quad (2.28)$$

Monte Carlo importance sampling is to use a number of (say N_p) independent samples drawn from $q(x)$ to obtain a weighted sum to approximate

$$\hat{f}_{N_p} = \frac{1}{N_p} \sum_{i=1}^{N_p} w(x^i) f(x^i) \quad (2.29)$$

where $w(x^i) = p(x^i)/q(x^i)$ are called importance weights. If the normalizing factor is not known, importance weights can be evaluated up to normalizing constant

$$w(x^i) \propto \frac{p(x^i)}{q(x^i)} \quad (2.30)$$

hence to ensure that $\sum_1^{N_p} w(x^i) = 1$, importance weight are normalized. Normalized weights represents the probability density function $p(x)$ as in Eq.(2.31),

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i) \quad (2.31)$$

Variance of the importance sampler can be stated as (Chen, 2003),

$$Var[\hat{f}] = \frac{1}{N_p} \int \left[\left(\frac{(f(x)p(x))^2}{q(x)} \right) \right] dx - \frac{(E_p[f(x)])^2}{N_p}. \quad (2.32)$$

The variance can be reduced when an appropriate $q(x)$ is chosen to match the shape of $p(x)$ so as to approximate the true variance; or match the shape of $|f(x)|p(x)$ so as to further reduce the true variance. Importance sampling estimate given by (2.31) is biased (thus a.k.a. biased sampling) but consistent, namely the bias vanishes rapidly at a rate $O(N_p)$. Importance sampling is useful in two ways (Chen, 2003) (i) it provides

an elegant way to reduce the variance of the estimator (possibly even less than the true variance); and (ii) it can be used when encountering the difficulty to sample from the true probability distribution directly.

Although theoretically the bias of importance sampler vanishes at a rate $O(N_p)$, the accuracy of estimate is not guaranteed even with a large N_p . If $q(x)$ is not close to $p(x)$, it can be imagined that the weights are very uneven, thus many samples are almost useless because of their negligible contributions. In a high dimensional space, the importance sampling estimate is likely dominated by a few samples with large importance weights.

2.5 Sequential Importance Sampling (SIS)

Sequential importance sampling forms the bases of all particle filter algorithms derived over past decades in the literature. SIS is a sequential Monte Carlo approach known variously as bootstrap filtering, condensation algorithm, particle filtering (Arulampalam et al., 2002) and survival of the fittest. It is a recursive Bayesian filter implemented with Monte Carlo technique. Key idea is to represent the posterior density with random samples and associated weights to compute required estimates. As given with its name SIS is the sequential version of importance sampler. As the number of samples approaches infinity samples represent the true posterior density function and SIS becomes the optimal Bayesian estimator independent of the shape of the posterior.

Weighted approximation to a density $p(\mathbf{x})$ is given as in Section 2.4. The posterior can be stated as the weighted approximation

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i) \quad (2.33)$$

where samples $\mathbf{x}_{0:k}^i$ are drawn from importance density $q(\mathbf{x}_{0:k}^i|\mathbf{y}_{0:k})$. Weights are defined according to importance sampling as in Eq.(2.34) as,

$$w_k^i \propto \frac{p(\mathbf{x}_{0:k}^i|\mathbf{y}_{0:k})}{q(\mathbf{x}_{0:k}^i|\mathbf{y}_{0:k})}. \quad (2.34)$$

To obtain a sequential formulation of Eq.(2.33) we need to obtain $p(\mathbf{x}_{0:k}^i|\mathbf{y}_{0:k})$ recursively from the approximation $p(\mathbf{x}_{0:k-1}^i|\mathbf{y}_{0:k-1})$ with a new set of samples. If

the importance density is chosen to factorize as,

$$q(\mathbf{x}_{0:k}|\mathbf{y}_{0:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})q(\mathbf{x}_{0:k-1}|\mathbf{y}_{0:k-1}) \quad (2.35)$$

then one can obtain samples $\mathbf{x}_{0:k}^i \sim q(\mathbf{x}_{0:k}|\mathbf{y}_{0:k})$ by augmenting each of the existing samples $\mathbf{x}_{0:k-1}^i \sim q(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$ with new state $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})$. To derive the weight update equation, $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ is expressed in terms of $p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$, $p(\mathbf{y}_k|\mathbf{x}_k)$ and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. So the resultant equation representing the weight term is expressed as,

$$w_k^i \propto \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}^i|\mathbf{y}_{1:k-1})}{q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})q(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})} \quad (2.36)$$

$$= w_{k-1}^i \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})} \quad (2.37)$$

If $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)$, which is the conditional independency defined between states and observations in a Kalman model, importance density becomes only dependent on \mathbf{x}_{k-1} and \mathbf{y}_k . This is particularly useful in the common case when only a filtered estimate of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is required at each time step. Consequently weight update equation is,

$$w_k^i \approx w_{k-1}^i \frac{p(\mathbf{y}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_k)} \quad (2.38)$$

and the posterior filtered density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ can be approximated as shown in Eq.(2.39),

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto \sum_{i=1}^{N_p} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (2.39)$$

where the weight w_k^i is defined in equation (2.38). The SIS algorithm thus consists of recursive propagation of weights as each measurement is received sequentially. Pseudo code for the algorithm is given below.

SIS PARTICLE FILTER

$$\left[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_p} \right] = SIS \left[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_p}, \mathbf{y}_k \right]$$

- FOR $i = 1 : N_p$
 - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{y}_k)$
 - Assign the particle a weight, w_k^i , according to (2.38).
- END FOR

2.5.1 Degeneracy problem

A common problem of sequential importance sampler is the degeneracy phenomenon. At each sequential step it is shown (Doucet, 1998) that variance of importance weights increase and consequently except one or a few, samples will have negligible weight after a while. This degeneracy implies that required posterior density is represented with just a few samples and computational power is wasted to update weight which have negligible contribution to posterior $p(\mathbf{x}_k | \mathbf{y}_{1:k})$. An appropriate criterion on degeneracy is Effective sample size introduced in (Liu and Chen, 1998) as,

$$N_{eff} = \frac{N_p}{1 + Var(w_k^{*i})}, \quad (2.40)$$

where $w_k^{*i} = p(\mathbf{x}_k^i | \mathbf{y}_{1:k}) / q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{y}_k)$ is referred as true weight. This cannot be evaluated exactly, but an estimate \widehat{N}_{eff} can be obtained by

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p} (w_k^i)^2} \quad (2.41)$$

where w_k^i is the normalised weight obtained using (2.38). In equation (2.41) always $N_{eff} \leq N_p$ and small N_{eff} indicates severe degeneracy. Degeneracy is an undesirable effect in particle filter that is hard to prevent. Brute force approach to reducing its effect is to use a very large N_p . This is often impractical, and so there are two other methods to rely on which are good choice of importance density and use of resampling.

2.5.2 Importance density

In (Liu and Chen, 1998) optimal importance density function which minimizes the true weights, w_k^{*i} , has been shown to be as in Eq.(2.42)

$$q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{y}_k)_{opt} = p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{y}_k). \quad (2.42)$$

However sampling from distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_k)$ is often impossible, as a consequence it's impractical to use optimal importance distribution. In many problems it is convenient to choose the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ as importance distribution. The weight update yields

$$w_k^i \propto w_{k-1}^i p(\mathbf{y}_k|\mathbf{x}_k^i). \quad (2.43)$$

Choosing prior as importance function simplifies the implementation however it may give poor results where prior distributions are wide compared to likelihood. Choice of importance density is crucial step in particle filter design and many different has been proposed in literature.

2.5.3 Resampling

Resampling is the second method to reduce the degeneracy problem. Basic idea of resampling is to eliminate the low weighted particles and concentrate on particles which have large weight. Resampling is to generate a new set of points $\{\mathbf{x}_k^{i*}\}_{i=1}^{N_p}$ from discrete density distribution,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto \sum_{i=1}^{N_p} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (2.44)$$

by resampling N_p times so that $Pr(\mathbf{x}_k^{i*} = \mathbf{x}_k^j) = w_k^j$. Resulting sample is an i.i.d. sample from the density given by Eq.(2.44) and as a result of resampling weights are now reset to $w_k^i = 1/N_p$. In literature one can find different resampling algorithms such as systematic sampling, stratified sampling, residual sampling and etc. Systematic resampling is chosen the scheme for ease of implementation and widely use.

Although the resampling step reduces the effects of degeneracy problem it introduces new problems. First, it limits the opportunity to parallelize the algorithm and second particle which have high weights w_k^i are statistically selected many times. This leads to loss of diversity and the resultant density will contain many repeated points. This is a severe problem particularly in small process noise and known as sample impoverishment. Another side effect of resampling is the reduced number of paths with time. Smoothed estimates based on paths will degenerate.

A pseudocode for generic particle filter with combined sampling and resampling step is given below.

GENERIC PARTICLE FILTER

$$\left[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_p} \right] = SIS \left[\left[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_p}, \mathbf{y}_k \right] \right]$$

- FOR $i = 1 : N_p$
 - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{y}_k)$
 - Assign the particle a weight, w_k^i , according to Eq.(2.38).
- END FOR
- Calculate total weight $t = SUM[\{w_k^i\}_{i=1}^{N_p}]$
- FOR $i = 1 : N_p$
 - Normalise: $w_k^i = w_k^i/t$
- END FOR
- Calculate \widehat{N}_{eff} using Eq.(2.41)
- IF $\widehat{N}_{eff} < N_T$
 - Resample obtained discrete probability distribution.
- END IF

2.5.4 Regularization

In standard particle filtering resampling step was suggested to reduce the degeneracy problem. However resampling introduced new problems such as loss of diversity among the particles also known as sample impoverishment, that results in poor representation of posterior density. Sample impoverishment arises due to the fact that, in the resampling stage samples are drawn from a discrete representation rather than a continuous one thus unavoidably causes sample replication. This may lead to particle collapse which is severe case of sample impoverishment where all particles occupy the same point in the state space. A modified particle filter known as the regularized particle filter (RPF) was proposed (Musso et al., 2001) as a potential solution to the above problem.

RPF is identical to the SIR particle filter except for the resampling stage. RBF resamples from an approximated continuous filtering density, whereas the SIR resamples from discrete approximation. Continuous approximated filtering density is defined using a kernel function K and weights w_t^i for $(i = 1 \dots N_p)$ associated with samples \mathbf{x}_k^i according to Eq.(2.45),

$$p(\mathbf{x}_k | \mathbf{y}_{0:t}) \approx \sum_{i=1}^{N_p} w_t^i K_h(\mathbf{x} - \mathbf{x}_k^i). \quad (2.45)$$

Kernel density $K()$ is explicitly

$$K_h(\mathbf{x}) = \frac{1}{h^{n_x}} K\left(\frac{\mathbf{x}}{h}\right), \quad (2.46)$$

where $h > 0$ is the kernel bandwidth, n_x is the dimension of the state vector. Regularization kernel K is chosen to be symmetric probability density function such that (Silverman, 1986),

$$K \geq 0, \quad \int K(\mathbf{x}) d\mathbf{x} = 1, \quad \int \mathbf{x} K(\mathbf{x}) d\mathbf{x} = 0, \quad \int \|\mathbf{x}\|^2 K(\mathbf{x}) d\mathbf{x} < \infty. \quad (2.47)$$

Kernel bandwidth is chosen to minimize the mean integrated square error between the true posterior density and corresponding regularized empirical representation in Eq.(2.45) defined as,

$$MISE(\hat{p}) = E \left[\int [\hat{p}(\mathbf{x}_k | \mathbf{y}_{1:k}) - p(\mathbf{x}_k | \mathbf{y}_{1:k})]^2 dx_k \right] \quad (2.48)$$

where $\hat{p}()$ denotes the approximated density given by Eq.(2.45). In the special case where all samples are equally weighted $w_k^i = 1/N_p$, $(i = 1 \dots N_p)$, optimal choice of Kernel is Epanechnikov Kernel (Silverman, 1986),

$$K_{opt} = \begin{cases} \frac{n_x + 2}{2c_{n_x}} (1 - \|\mathbf{x}^2\|) & \text{if } \|\mathbf{x}^2\| < 1 \end{cases} \quad (2.49)$$

where c_{n_x} is the volume of the unit hypersphere. Assuming that underlying density is Gaussian with a unit covariance matrix, optimal choice for the bandwidth is (Silverman, 1986)

$$h_{opt} = A(k) N^{\frac{-1}{n_x+4}}, \quad \text{where } A(k) = (4/(n_x + 2))^{\frac{1}{n_x+4}} \quad (2.50)$$

However when multimodal distributions is in question it is convenient to choose the bandwidth $h = h_{opt}/2$. Although results in Eq.(2.49) are optimal under the assumption

that weights are equally weighted, they can be used in the general case to obtain a suboptimal filter. One iteration of RPF is described in (Arulampalam et al., 2002) as described below.

REGULARIZED PARTICLE FILTER

$$\left[\{\mathbf{x}_k^{i*}, w_k^i\}_{i=1}^{N_p} \right] = RPF \left[\{\mathbf{x}_k^{i*}, w_k^i\}_{i=1}^{N_p}, \mathbf{y}_k \right]$$

- For $i = 1 : N_p$
 - Draw $\mathbf{x}_k^i \propto q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{y}_k)$
 - Assign the particle a weight, w_k^i according to Eq.(2.38).
- End For
- Normalize the weights such that, $\sum_{i=1}^{N_p} w_t^i = 1$
- Calculate \hat{N}_{eff} using Eq.(2.41)
- If $\hat{N}_{eff} < N_t$
 - Calculate the empirical covariance matrix \mathbf{S}_k of $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_p}$
 - Compute \mathbf{D}_k such that $\mathbf{D}_k \mathbf{D}_k^T = \mathbf{S}_k$
 - Resample
 - * $\left[\{\mathbf{x}_k^{i*}, w_k^i\}_{i=1}^{N_p} \right] = \text{RESAMPLE} \left[\{\mathbf{x}_k^{i*}, w_k^i\}_{i=1}^{N_p} \right]$
 - For $i = 1 : N_p$
 - * Draw $\epsilon^i \sim K()$ from the Kernel
 - * $\mathbf{x}_k^{i*} = \mathbf{x}_k^i + h \mathbf{D}_k \epsilon^i$
 - End For
- End If

2.6 Sequential Monte Carlo (SMC) Samplers

In sequential Monte Carlo algorithms such as particle filtering given in Section.2.5, we sample from a sequence of target densities evolving with a countable index n , $\pi_1(x_1) \dots \pi_n(x_n)$, each defined on a common measurable space (E_n, \mathcal{E}_n) where

$x_n \in E_n$. Conventionally the particle filter is defined on the sequence of target densities $\pi_1(x_1) \dots \pi_n(x_n)$ where corresponding proposal distributions are defined as $\eta_1(x_1) \dots \eta_n(x_n)$. The importance weight W_n at time n can be defined as

$$W_n = \frac{\gamma_n(x_n)}{\eta_n(x_n)} \quad (2.51)$$

γ_n is the unnormalized target distribution according to $\pi_n = \gamma_n/Z$ where Z is the normalizing constant.

In order to derive the importance weights sequentially one needs to calculate the proposal distribution $\eta_n(x_n)$ pointwise which can be explicitly defined for $k = 1 \dots n$ by Eq.(2.52).

$$\eta_n(x_n) = \int \eta_1(x_1) \prod_{k=2}^n K_k(x_k|x_{k-1}) dx_{1:n-1} \quad (2.52)$$

As it is shown in Eq.(2.52), computation of the importance distribution $\eta_n(x_n)$ for $n > 1$ requires an integration with respect to $x_{1:n-1} = \{x_1 \dots x_{n-1}\}$ thus a closed form solution to $\eta_n(x_n)$ is not available except for specifically designed kernels i.e. independently selected proposal kernels where $K(x_k|x_{k-1}) = K(x_k)$. This constitutes a central limitation of particle filtering.

To appreciate this limitation, we will consider the form of the kernel explicitly. The particle filtering is a common framework for sampling from a target distribution with an increasing dimension over time that admits the integration in Eq.(2.52) to be trivial. Whenever a new observation arrives at time n , the vector x_{n-1} needs to be extended to $x_n = \{x_{n,1} \dots x_{n,n}\}$ resulting in an increase in the dimension of target distribution by one. Hence the particle filtering proposal kernel is defined as follows,

$$K_n(x_{n-1}, x_n) = \delta_{x_{n-1}}(x_{n,-n}) \eta_n(x_{n,n}|x_{n,-n}). \quad (2.53)$$

where the notation $x_{n,-n} = \{x_{n,1} \dots x_{n,n-1}\}$ denotes the components of x_n excluding $x_{n,n}$. The dirac delta function $\delta_{x_{n-1}}(x_{n,-n})$ shown in Eq.(2.53) just copies $x_{n-1} = \{x_{n-1,1} \dots x_{n-1,n-1}\}$ as it is. Hence the particle filtering proposal given in Eq.(2.53) only needs to label the latest component $x_{n,n}$ while preserving the rest, $x_{n,-n} = \{x_{n,1} \dots x_{n,n-1}\}$. In other words, the conventional particle filter is not capable of defining local moves on previous labels. This can be expressed as one of the major drawback that explains why conventional particle filtering methods are not efficient in fixed parameter estimation in general, and for DPM models in particular.

To eliminate this limitation, Del Moral et al. (Del Moral et al., 2006) proposed an auxiliary variable technique which solves the sequential importance sampling problem in an extended space $E^n = \{E_1 \times \dots \times E_n\}$. SMC sampler performs importance sampling between the joint importance distribution $\eta_n(x_{1:n})$ and the artificial joint target distribution defined by $\tilde{\pi}_n(x_{1:n}) = \tilde{\gamma}_n(x_{1:n})/Z_n$ where Z_n denotes the normalizing constant. The algorithm enables us to calculate efficient weight update equations for a given valid proposal kernel $K_n(x_{n-1}, x_n)$.

The proposal distribution $\eta_n(x_{1:n})$ of SMC is defined on extended space E^n as follows,

$$\eta_n(x_{1:n}) = \eta_1(x_1) \prod_{k=2}^n K(x_k|x_{k-1}) \quad (2.54)$$

Note that here an integration is no longer required. However, this comes with the expense of a extended artificial target density on the same extended space E^n defined by

$$\tilde{\gamma}_n(x_{1:n}) = \gamma_n(x_n) \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k). \quad (2.55)$$

Here, we introduced a sequence of backward kernels $L_k(x_{k+1}, x_k)$, $k = \{1 \dots n - 1\}$ to define the artificial target distribution shown in Eq.(2.55). Consequently $\tilde{\pi}_n(x_{1:n})$ defined on extended space E^n admits $\pi_n(x_n)$ as a marginal by construction therefore the resultant weighting function ensures convergence to the true target density.

The generic SMC algorithm which is used to sample from a sequentially evolving target posterior $\tilde{\pi}_n$ is presented as follows (Del Moral et al., 2006).

Assume that a set of weighted particles $\{w_{n-1}^i, x_{1:n-1}^i\}_{i=1}^{N_p}$ approximate to $\tilde{\pi}_{n-1}$ at time $n - 1$. At time n the path of each particle can be extended using a Markov kernel, $x_n^i \sim K_n(x_{n-1}^i, x_n)$. The unnormalized importance weights associated with the extended particles are calculated according to Eq.(2.56),

$$\begin{aligned} W_n(x_{1:n}) &= W_{n-1}(x_{1:n-1})v_n(x_{n-1}, x_n) \\ &= \frac{\tilde{\gamma}_n(x_{1:n})}{\eta_n(x_{1:n})} \end{aligned} \quad (2.56)$$

where the incremental term of weight equation, $v_n(x_{n-1}, x_n)$, is equal to

$$v_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n)L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}. \quad (2.57)$$

As the discrepancy between η_n and $\tilde{\gamma}_n$ tends to increase with n , variance of the unnormalized importance weights tends to increase that yields degeneracy. A

resampling scheme is used if degeneracy is above a certain level as measured by, e.g. effective sample size (ESS) (Liu and Chen, 1998).

2.6.1 Backward kernels

Design of efficient sampling schemata hinges on properly choosing L_n with respect to K_n . The introduction of the L_n extends the integration domain from E to E^n and eliminates the necessity of calculating $\eta_n(x_n)$. However increasing the integration domain also increases the variance of the importance weights. In (Del Moral et al., 2006) it is shown that the optimal backward Markov kernel L_k^{opt} ($k = 1, \dots, n$) minimizing the variance of the unnormalized importance weight $w_n(x_{1:n})$ is given for any k by,

$$L_{k-1}^{opt}(x_k, x_{k-1}) = \frac{\eta_{k-1}(x_{k-1})K_k(x_{k-1}, x_k)}{\eta_k(x_k)} \quad (2.58)$$

Replacement of the optimal backward kernel in Eq.(2.58) with the one defined in Eq.(2.56) yields the importance weight,

$$W_n(x_{1:n}) = \frac{\gamma_n(x_n)}{\eta_n(x_n)}. \quad (2.59)$$

Eq.(2.59) states that, using the optimal backward kernel, importance sampling is performed on E instead of E^n . However, the marginal distribution, $\eta_n(x_n)$, shown in Eq.(2.59) usually does not admit a closed form solution therefore it is almost never possible to use the optimal backward kernel. The common strategy is to approximate the optimal kernel as close as possible to provide asymptotically consistent estimates (Del Moral et al., 2006). A sensible approximation can be obtained by substituting π_{k-1} for η_{k-1} , $k = 2 \dots n$ where the approximate kernel L_{k-1} can be expressed as in Eq.(2.60),

$$L_{k-1}(x_k, x_{k-1}) = \frac{\pi_{k-1}(x_{k-1})K_k(x_{k-1}, x_k)}{\int \pi_{k-1}(x_{k-1})K_k(x_{k-1}, x_k)dx_{k-1}} \quad (2.60)$$

yielding the weight update equation,

$$v_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n)}{\int_E \gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)dx_{n-1}}. \quad (2.61)$$

Since γ_n is known analytically, it is convenient to use Eq.(2.61) rather than Eq.(2.59)

2.6.2 Forward kernels

Another important issue that needs to be discussed is specification of forward kernels. Gibbs type kernels and MCMC kernels are commonly used kernel type in SMC sampler. In practice it is often useful to define Gibbs kernels where partial state space is updated particularly for sequential applications. Let $x_{n,u}$ denotes the u^{th} component x_u of $x = (x_1, \dots, x_j)$ at time n . It is straightforward to establish the proposal minimizing the variance of Eq.(2.61) conditional on x_{n-1} as shown in Eq.(2.62)

$$K_n(x_{n-1}, dx_n) = \delta_{x_{n-1,-u}}(dx_{n,-u})\pi_n(dx_{n,u}|x_{n,-u}) \quad (2.62)$$

where $x_{n,-u} = (x_{n,1}, \dots, x_{n,u-1}, x_{n,u+1}, \dots, x_{n,j})$. In this case the backward kernel can be represented as in Eq.(2.63)

$$L_{n-1}(x_n, dx_{n-1}) = \delta_{x_{n,-u}}(dx_{n-1,-u})\pi_{n-1}(dx_{n-1,u}|x_{n-1,-u}) \quad (2.63)$$

and the incremental weight update equation for a Gibbs type kernel is,

$$\tilde{w}_n(x_{n-1}, x_n) = \frac{\gamma_n(x_{n-1,-u}, x_{n,u})}{\gamma_{n-1}(x_{n-1,-u})\pi_n(x_{n,u}|x_{n-1,-u})} \quad (2.64)$$

Assuming K_n is an MCMC kernel of invariant distribution π_n , an approximation of Eq.(2.60) can be obtained as shown in Eq.(2.65)

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1})K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \quad (2.65)$$

Eq.(2.65) is the reversal Markov kernel associated with K_n . Therefore a good approximation to Eq.(2.61) can be derived for $\pi_{n-1} \approx \pi_n$ and the unnormalized incremental weight equation is,

$$\tilde{w}_n(x_{n-1}, x_n) = \frac{\gamma_n(x_{n-1})}{\gamma_{n-1}(x_{n-1})} \quad (2.66)$$

The inference framework is fairly general and several methods proposed in the literature appear as special cases of the SMC sampler (Del Moral et al., 2006), including the sequential Monte Carlo algorithms using MCMC kernels are proposed by Chopin (Chopin, 2002), Jarzynski (Jarzynski, 1997), Neal (Neal, 2001) and MacEachern (MacEachern et al., 1999).

2.7 Dirichlet Process Mixture Models

In recent years, Dirichlet Process Mixtures (DPM) model has been one of the most widely used and popular approach to nonparametrical probabilistic models (Antoniak, 1974). Originally, DPM have been widely used as a building block in hierarchical models for solving density estimation and clustering problems where the actual form of the data generation process is not constrained to a particular parametric family such as mixture problems with unknown number of components (Blei and Jordan, 2004). Now we will explain the DPM model structure and the notation used throughout the thesis. Let us first define the notation for a finite mixture model, and then it will be extended to a DPM model.

In a batch Bayesian setting, the joint distribution corresponding to a finite mixture model over N observations $y = \{y_i\}, i = 1 \dots N$, can be defined as follows:

$$p(\theta, z, y) = \left(\prod_{i=1}^N p(z_i)g(y_i|\theta_{z_i}) \right) \prod_{j=1}^k p(\theta_j) \quad (2.67)$$

Here, for $i = 1 \dots N$, $z_i \in \{1 \dots k\}$ denotes the cluster index of the i th observation and $\theta = \{\theta_j\}, j \in \{1 \dots k\}$ denote the cluster conditional parameters. Here, k denotes the maximum number of clusters. We will use $z = \{z_i\}, i = 1 \dots N$ to refer to clustering variables, that we also call cluster labels or simply labels. The mixture density $p(y)$ can be obtained as the marginal by summing over the clustering variables z and integrating over mixture component parameters θ .

Given a set of observations y , in order to calculate the posterior probability $p(z, \theta|y)$, it is necessary to know k , the number of mixture components. However, this is rarely the case in practice and k needs to be estimated using more sophisticated inference techniques (Green, 1995). The DPM model introduced by Antoniak (1974) provides, among others, an elegant alternative for construction of mixture models with unknown number of components. In the sequel, we will refer to the target posterior as

$$\pi(x) \equiv p(z, \theta|y) \quad (2.68)$$

where $x = \{z, \theta\}$. It is advantageous to construct a mixture model sequentially, where data arrives one by one. Note that unlike a time series model, the likelihood will be invariant with respect to the actual order. This will also highlight the connection to a DPM which is easily constructed as a sequential process. To denote the sequential

construction, we extend our notation with an explicit 'time' index n . Note that this index does not correspond to the wall clock, it just denotes the number of observations seen so far.

We denote the observation sequence at time n by $y_n = \{y_{n,1} \dots y_{n,n}\}$. Each observation $y_{n,i}$, $i = 1, \dots, n$, is assigned to a cluster where $z_{n,i} \in \{1, \dots, k_n\}$ is the cluster label and, $k_n \in \{1 \dots n\}$ represent the number of clusters at time n . The vector of cluster variables is defined as $z_n = \{z_{n,1} \dots z_{n,n}\}$ and corresponding cluster parameters are represented with the parameter vector $\theta_n = \{\theta_{n,1} \dots \theta_{n,k_n}\}$.

The reader may find this notation redundant, as clearly $y_{n_1,j} = y_{n_2,j}$ for all $j \leq \min(n_1, n_2)$. However, the notation will be justified in the SMC samplers framework where we will target a joint density on all times $n = 1 \dots N$.

The DPM model assumes that the cluster parameters are independently drawn from the prior $\pi(\theta)$ and the observations are independent of each other conditional on the assignment variable $z_{n,i}$. Hence the DPM posterior density $\pi(x_n)$ can be expressed as,

$$\pi_n(x_n) \propto p(z_n) \prod_{j=1}^{k_n} p(\theta_{n,j}) \prod_{i=1}^n g(y_{n,i} | \theta_{n,z_{n,i}}) \quad (2.69)$$

where $x_n = \{z_n, \theta_n\}$. The prior on clustering variable vector z_n is formulated by Eq.(2.70) in a recursive way,

$$p(z_{n,i+1} = j | z_{n,\{1:i\}}) = \left\{ \begin{array}{ll} \frac{l_j}{i+\alpha}, & j = 1, \dots, k_i \\ \frac{\alpha}{i+\alpha}, & j = k_i + 1 \end{array} \right\} \quad (2.70)$$

where k_i is the number of clusters in the assignment $z_{n,\{1:i\}}$. l_j is the number of observations that $z_{n,\{1:i\}}$ assigns to cluster j and α is a 'novelty' parameter ; when α is large the process has the tendency to generate new clusters. An interesting mathematical property of this construction is that it assigns the same prior probability to similar partitions regardless of the order of observations as l_j are clearly independent of the actual order. For a rigorous development from first principles, see the so called stick breaking construction (Sethuraman, 1994).

We assume that conjugate prior is chosen for the parameters to ensure the conditions described in (Fearnhead, 2004). Typically given z_n , under conjugacy assumption the parameter θ_n can be integrated out and the DPM posterior distribution can be calculated up to a normalizing constant.

2.8 Models and Algorithms for Maneuvering Target Tracking

In this section, we give basic information on target motion models that defines the prior structure of the tracking algorithms and introduce well known state of the art maneuvering target tracking models introduced in the literature. We will present three models, interacting multiple model (IMM), multiple model particle filter (MMPF) and jump Markov system particle filter (JMS-PF), that all regard on the multiple model structure. These models differ in the way they unify the inference scheme with the multiple model approach. We also explain the variable rate particle filters (VRPF) which is one of the recent advances in the field of maneuvering target tracking.

Multiple models can be stated as a Hybrid scheme, including continuous and discrete random variable in a Bayesian framework. In maneuvering Target tracking, discrete variables represents the hypothesis of target mode sequence. However exponential increment in the number of hypothesis with time will render the optimum solution infeasible. Common approach taken is to reduce the number of hypothesis by some pruning or merging techniques. Final estimate is usually constructed with mixing or selecting the models.

Although, MM approaches defined in this section have quite different inference schemes, they try to solve the similar probabilistic structures regarding hybrid Markov models. A distinct approach proposed recently is Variable Rate Particle filters which models the maneuvering times as state process independent of observation times (Godsill and Vermaak, 2005). In contrast to other models state arrival times are assumed as random processes. The resultant model refers to a semi Markov model for tracking. The model claims that nature of a maneuvering target motion, particularly manned vehicles, fits better to the variable rate scheme.

2.8.1 Target dynamic motion models

Model based tracking methods assume that the target motion and its observations can be represented by some known mathematical models accurately. The most known models are state space models in the discrete form,

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, u_k) + \mathbf{w}_k \quad (2.71)$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k \quad (2.72)$$

where f_k and h_k are time indexed functions which defines the motion and the observation model. $\mathbf{x}_k, \mathbf{y}_k, u_k$ are target state vector, observation vector and control input at discrete time t_k respectively, and w_k, v_k are process and measurement noise sequences.

One of the major challenge for target tracking arises from motion uncertainty (Li and Jilkov, 2003). This uncertainty is caused by the lack of accurate knowledge of the target dynamic model. Although adequate models can be considered in state space form as in Eq.(2.71), a tracker has to confront the absence of actual control input u of the target and possibly actual form of $f()$, its parameters and statistical properties of the noise w for the particular target. In the case of maneuvering target tracking a motion model that accounts well for the true dynamic behavior of the target has to be considered. Most of the relevant work with the subject focused on 2 main tasks (Li and Jilkov, 2003).

- Approximate the actually nonrandom control input u as a random process of certain properties
- Describe typical target trajectories by some representative motion models with properly designated parameters.

Target motion can be classified into two major regimes, one is maneuver and the other is non-maneuver. In the sequel we will give a short introduction on non-maneuver and maneuver motion models. There are numerous motion models in literature used for target tracking that aims to describe the target kinematics explicitly to increase the tracker performance. Models can be clustered into 2 main types of which based on random processes and the ones based on target kinematics. Appropriate choice of motion model is highly dependent on the application and the type of the tracking algorithm. Thus we will just explain a few basic examples of such models and a sophisticated one which will be used in the presented work.

2.8.1.1 Non maneuvering motion model

Non maneuvering motion is commonly referred as the straight motion at constant velocity. It is generally expressed in the discrete form as (Bar-Shalom et al., 2001),

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k \tag{2.73}$$

where state vector $\mathbf{x} = [x, y, \dot{x}, \dot{y}]$ constitutes of coordinate positions x, y and velocities along x and y axis \dot{x}, \dot{y} respectively. Likewise, process noise vector \mathbf{w}_k constitutes of $[w_x, w_y]$ which corresponds to noisy accelerations along the x any y axes. Process noise vector is commonly chosen as $\mathbf{w}_k \sim N(0, \mathbf{Q})$ where covariance matrix is chosen to be $\mathbf{Q} = \sigma_w \mathbf{I}$ which defines an uncoupled acceleration across x and y directions. In constant velocity model, parameter σ_w is selected as a small value to model the uncertainties. The matrix \mathbf{F} also named as transition matrix is,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.74)$$

and matrix \mathbf{G} is defined as,

$$\mathbf{G} = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix} \quad (2.75)$$

where T is the sampling period. Since constant velocity model assumes independent constant acceleration values with a very small process noise σ_w between state transitions, by increasing the value of process noise σ_w , it can converted to a maneuvering target model and is so called constant white acceleration model. However a maneuver by its nature aims at accomplishing a certain task and thus is rarely independent with respect to time thus this model can be represented as the simplest maneuver model and usually used when maneuver is quite small.

A maneuvering motion can be expressed as the motion mode, exempt the straight constant velocity. Variation of target acceleration is accounted as a state variable in Maneuvering Target models. They differ in the way they deal with the acceleration. In the following sections maneuvering motion models of which are used extensively in the literature will be described.

2.8.1.2 Constant acceleration model (CA)

Constant acceleration model assumes the acceleration is a process with independent increments (Bar-Shalom et al., 2001). State vector $\mathbf{x} = [x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}]$ also includes acceleration deviation \ddot{x}, \ddot{y} respectively along x and y axis. This model is most conveniently expressed in the discrete time given as,

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k \quad (2.76)$$

In Eq.(2.76), the transition matrix \mathbf{F} is,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 & T^2/2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2/2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.77)$$

and the matrix \mathbf{G} is,

$$\mathbf{G} = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2.78)$$

where T is sampling period and \mathbf{w}_k constitutes the noise vector $[w_x, w_y]$. The assumption of the direct discrete-time CA model, that the acceleration increment $\Delta a_k = a_{k+1} - a_k$ is independent across different sampling intervals, is hardly justifiable. However more complex models are proposed by Singer (1970) that assumes target acceleration is a zero mean first order stationary Markov process. The process has autocorrelation $R_a = E[a(t + \tau)a(t)] = \sigma^2 e^{-\alpha|\tau|}$ and should be expressed as a linear time invariant discrete model $a_{k+1} = e^{-\alpha T} a_k + w_k^a$ where w_k^a is a zero mean white noise sequence with variance $\sigma^2(1 - e^{-\alpha 2T})$.

2.8.1.3 Constant turn model (CT)

Constant Turn model is one of the well known models based on target kinematic unlike the others we have represented before (Bar-Shalom et al., 2001). CT comprises of curvilinear motion model from kinematics. This is a constant speed and constant turn rate model, and defined with the state vector $\mathbf{x} = [x, y, \dot{x}, \dot{y}]$ as,

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k \quad (2.79)$$

CT motion transition matrix \mathbf{F} is given as a function of sampling time T and angular rate w ,

$$\mathbf{F} = \begin{bmatrix} 1 & \sin(wT)/wT & 0 & -(1 - \cos(wT))/w \\ 0 & \cos(wT) & 0 & -\sin(wT) \\ 0 & (1 - \cos(wT))/w & 1 & \sin(wT)/wT \\ 0 & \sin(wT) & 0 & \cos(wT) \end{bmatrix} \quad (2.80)$$

and usually matrix \mathbf{G} chosen is identical to Eq.(2.78). An important issue we have to indicate is that CT model is completely linear owing to angular rate w is known

priorly. However as a practical consideration this assumption is not justifiable. As a consequence, models are developed assuming angular rate w as a random variable which is included as a state component to be estimated. Angular rate is defined by an wiener process or a first order Markov process when interpreted as a random variable. Note that under this circumstance the model should no longer be linear thus linearization techniques may be required to reach a solution depending on the algorithm used.

2.8.1.4 Curvilinear motion models (CL)

Curvilinear motion models is one of the most sophisticated model considered in the literature (Li and Jilkov, 2003). It accounts for possibly non-zero normal (cross-track) and tangential (along-track) target maneuver accelerations simultaneously. In (Li and Jilkov, 2003) continuous and discrete form equations for curvilinear motion model in cartesian coordinates has been presented. However we will represent the curvilinear motion a slightly different way, represented using intrinsic coordinate system (Godsill and Vermaak, 2005). Note that a special case of curvilinear motion dynamics for a particular scenario is presented here which is in accord with the model used in Variable Rate particle filters.

In the intrinsic coordinate system applied forces are represented relative to heading of the object, distance traveled along the path of motion is denoted by s , while the angle of the path relative to x axis is denoted by ψ . Dynamics of motion is determined by the tangential acceleration a_T and perpendicular acceleration a_P defined as in Eq.(2.81) as,

$$a_T(t) = \frac{d^2s}{dt^2}, \quad a_p(t) = \frac{ds}{dt} \frac{d\psi}{dt} = \frac{1}{R} \left(\frac{ds}{dt} \right)^2 \quad (2.81)$$

where R is the instantaneous radius of curvature of the path. Piecewise constant force relative to the direction of heading ψ is assumed to be applied having tangential component $T_{T,k}$ and perpendicular component $T_{P,k}$ over time interval $(\tau_k, \tau_{k+1}]$. This model is suitable particularly scenarios where applied forces remains constant relative to the direction of flight between any two times τ_k and $\tau_k + 1$. Once the initial conditions are set, it is possible to define the target trajectory deterministically. In the scenario proposed, object being tracked makes discrete changes at random times to the controls and follows a deterministic path between these change times. To model the

external effects such as air resistance, a frictional force $\lambda(ds/dt)$ is assumed to apply in the opposite direction to the path. Applied forces in tangential and perpendicular direction is represented in Eq.(2.82).

$$T_{T,k} = \lambda \frac{ds}{dt} + m \frac{d^2s}{dt^2}, \quad T_{P,k} = m \frac{ds}{dt} \frac{d\psi}{dt}, \quad \tau_k < t \leq \tau_{k+1} \quad (2.82)$$

where m is the mass of the object and λ the coefficient of resistance. To represent the speed $v(t)$ along the path at time $t = \tau_k + \Delta\tau$ integration from time τ_k to $\tau_k + \Delta\tau$ is accomplished as in Eq.(2.83),

$$v(\tau_k + \Delta\tau) = \frac{1}{\lambda} (T_{T,k} - (T_{T,k} - \lambda v(\tau_k))e^{-\Delta\tau\lambda/m}) \quad (2.83)$$

Distance along the path is,

$$s(\tau_k + \Delta\tau) = s(\tau_k) + \frac{\delta\tau}{\lambda} T_{T,k} + \frac{m}{\lambda^2} (T_{T,k} - \lambda v(\tau_k))(e^{-\Delta\tau\lambda/m} - 1). \quad (2.84)$$

Then using equation (2.83) representing $v(t)$, perpendicular equation is rearranged and integrated as,

$$\frac{d\psi}{dt} = \frac{T_{P,k}}{mv(t)}, \quad \tau_k < t \leq \tau_{k+1}, v(t) \neq 0 \quad (2.85)$$

$$\psi(\tau_k + \Delta\tau) = \psi(\tau_k) + \int_{\tau_k}^{\tau_k + \Delta\tau} \frac{T_{P,k} dt}{mv(t)} \quad (2.86)$$

$$= \psi(\tau_k) + T_{P,k}/T_{T,k} \left(\Delta\lambda/m - \log \left| \frac{v(\tau_k)}{v(\tau_k + \Delta\tau)} \right| \right). \quad (2.87)$$

Results representing speed $v(t)$ and angle of the path $\psi(t)$ are in closed form however closed form solution is not available for position in cartesian coordinates which requires a numerical integration technique to compute. In (Godsill and Vermaak, 2005) a simple Euler approximation is used on a fine time grid, calculating the changes in x and y coordinates over a time interval δt as shown in Eq.(2.88)

$$\begin{aligned} \delta x &\approx v(t) \cos(\psi(t)) \delta t, \\ \delta y &\approx v(t) \sin(\psi(t)) \delta t, \\ \mathbf{z}(t + \delta t) &\approx \mathbf{z}(t) + [\delta x \quad \delta y]^T. \end{aligned} \quad (2.88)$$

2.8.2 Interacting multiple models (IMM)

Interacting Multiple Model represented by Blom (Blom et al., 1998) is one of the best known algorithm that deserved this popularity by having satisfactory results with a

reasonable complexity. Although it is not mandatory, mode sequence is assumed as first order Markov process that helps to reveal a tractable algorithm.

Assume S different models at time k are used, denoted by $r_k^{(i)}$, $\{i = 1 \dots S\}$, where probability for each model is $\gamma_k^{(i)} = p(r_k^{(i)} | \mathbf{y}_{0:k})$. The posterior density distribution at time k is given by the total probability theorem using S different models as,

$$p(\mathbf{x}_k | \mathbf{y}_{0:k}) = \sum_{j=1}^S p(\mathbf{x}_k | r_k^{(j)}, \mathbf{y}_{0:k}) P(r_k^{(j)} | \mathbf{y}_{0:k}). \quad (2.89)$$

Applying Bayes theorem to the first factor in (2.89) using $\mathbf{y}_{0:k} = \{\mathbf{y}_k, \mathbf{y}_{0:k-1}\}$ gives

$$p(\mathbf{x}_k | r_k^{(j)}, \mathbf{y}_{0:k}) \propto p(\mathbf{y}_k | r_k^{(j)}, \mathbf{x}_k) p(\mathbf{x}_k | r_k^{(j)}, \mathbf{y}_{0:k-1}). \quad (2.90)$$

Applying the total probability theorem to the last factor in (2.90) gives

$$p(\mathbf{x}_k | r_k^{(j)}, \mathbf{y}_{0:k-1}) = \sum_{i=1}^S p(\mathbf{x}_k | r_{k-1}^{(j)}, r_{k-1}^{(i)}, \mathbf{y}_{0:k-1}) \underbrace{p(r_{k-1}^{(i)} | r_{k-1}^{(j)}, \mathbf{y}_{0:k-1})}_{\gamma_{k-1|k-1}(i,j)} \quad (2.91)$$

IMM can be integrated with any inference scheme that can represent the distribution,

$$p(\mathbf{x}_k | r_{k-1}^{(j)}, r_{k-1}^{(i)}, \mathbf{y}_{0:k-1}). \quad (2.92)$$

If we assume the state variables as Gaussian distribution conditioned on mode then Eq.(2.91) can be approximated as,

$$p(\mathbf{x}_k | r_k^{(j)}, \mathbf{y}_{0:k-1}) \approx \sum_{i=1}^S p(\mathbf{x}_k | r_{k-1}^{(j)}, r_{k-1}^{(i)}, \left\{ \hat{\mathbf{x}}_{k-1|k-1}^l, \hat{\mathbf{P}}_{k-1|k-1}^l \right\}_{l=1}^S) \gamma_{k-1|k-1}(i, j) \quad (2.93)$$

$$= \sum_{i=1}^S p(\mathbf{x}_k | r_{k-1}^{(j)}, r_{k-1}^{(i)}, \hat{\mathbf{x}}_{k-1|k-1}^{(i)}, \hat{\mathbf{P}}_{k-1|k-1}^{(i)}) \gamma_{k-1|k-1}(i, j). \quad (2.94)$$

Approximation is due to the fact that the models summarize the history through the estimates and covariances. Mixing probabilities are expressed using Bayes theorem as

$$\gamma_{k-1|k-1}(i, j) = p(r_{k-1}^{(i)} | r_k^{(j)}, \mathbf{y}_{0:k-1}) \quad (2.95)$$

$$\propto \underbrace{p(r_k^{(j)} | r_{k-1}^{(i)}, \mathbf{y}_{0:k-1})}_{p(i,j)} \underbrace{p(r_{k-1}^{(i)} | \mathbf{y}_{0:k-1})}_{\gamma_{k-1}^{(i)}} \quad (2.96)$$

where $p(i, j)$ in practice is used as a design parameter. Eq. (2.93) is defined as a Gaussian mixture. An approximation is required to define it with a single Gaussian.

Moreover a similar methodology is adopted to estimate the joint mode distribution. This situation well defines the algorithm that integrates the EKF with IMM model and usually referred as IMM-EKF algorithm. However we wont give further details of the algorithm in this thesis.

2.8.3 Multiple model particle filter (MMPF)

The MMPF has been used to solve various maneuvering target tracking problems (McGinnity and Irwin, 2000). MMPF considers an augmented hybrid state vector, $\mathbf{x}_k = [x_k^*, r_k]$ where x_k^* is the continuous state vector and the r_k is the discrete mode variable. In order to recursively compute PF estimates, the MC representation of $p(\mathbf{x}_k | \mathbf{y}_{0:k})$ has to be propagated in time using posterior pdf $p(\mathbf{x}_{k-1} | \mathbf{y}_{0:k-1})$ of which is approximated by the samples and associated weights $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_p}$ according to,

$$p(\mathbf{x}_{k-1} | \mathbf{y}_{0:k-1}) \approx \sum_{i=1}^{N_p} w_{k-1}^i \delta(\mathbf{y}_{k-1} - \mathbf{y}_{k-1}^i) \quad (2.97)$$

where $\delta(\cdot)$ is the dirac delta measure. Posterior pdf at k can be written as in Eq.(2.98).

$$p(\mathbf{x}_k | \mathbf{y}_{0:k}) \propto p(\mathbf{y}_k | \mathbf{x}_k) \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{0:k-1}) d\mathbf{x}_{k-1} \quad (2.98)$$

To represent the posterior given by Eq.(2.98) using particles, importance sampling is employed. By choosing the importance density to be prior $p(\mathbf{x}_k | \mathbf{x}_{k-1})$, one can draw samples from $\mathbf{x}_k^i \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$, for $i = 1, \dots, N_p$. First sample is drawn from the discrete probability mass function $p(r_k | r_{k-1}^i)$ which indicates the transition probabilities between modes defined as an first order markov chain. The samples are represented as,

$$r_k^i \sim p(r_k | r_{k-1}^i) \quad (2.99)$$

Next, given the sample r_k^i , one can easily sample $\mathbf{x}_k^i \sim p(\mathbf{x}_k | x_{k-1}^i, r_k^i)$ through the dynamic model with the given process noise. This gives us the sample space $\{\mathbf{x}_k^i\}$ of the posterior $p(\mathbf{x}_k | \mathbf{y}_{0:k})$ and the weights associated with the samples are computed recursively using the measurement equation as,

$$w_k^i \propto w_{k-1}^i p(\mathbf{y}_k | \mathbf{x}_k^i). \quad (2.100)$$

The degeneracy phenomenon which is common problem with Particle filters also arise in MMPF. A conventional resampling step to overcome degeneracy is used as in particle filters.

2.8.4 Jump Markov system particle filter (JMS-PF)

The JMS-PF is based on jump Markov linear system proposed in (Doucet et al., 2001b). Standard particle filtering techniques focused on estimating the pdf of the state variable \mathbf{x}_k as described in MM-PF. However, idea in the JMS-PF is to emphasize the estimation of mode sequence $\mathbf{r}_{0:k}$, given the measurements $\mathbf{y}_{0:k}$. The density of latent variables $p(\mathbf{x}_{0:k}, \mathbf{r}_{0:k} | \mathbf{y}_{0:k})$ can be factorized into,

$$p(\mathbf{x}_{0:k}, \mathbf{r}_{0:k} | \mathbf{y}_{0:k}) = p(\mathbf{x}_{0:k} | \mathbf{r}_{0:k}, \mathbf{y}_{0:k}) p(\mathbf{r}_{0:k} | \mathbf{y}_{0:k}). \quad (2.101)$$

Eq. (2.101) indicates that given a specific mode sequence and the measurements, $p(\mathbf{x}_{0:k} | \mathbf{r}_{0:k}, \mathbf{y}_{0:k})$ can be estimated with any conventional nonlinear filtering method. Therefore mode sequence estimation is considered as the weak spot of the state density estimation. Methodology is to estimate mode sequence $p(\mathbf{r}_{0:k} | \mathbf{y}_{0:k})$ using a PF. Using Bayes rule, the equation,

$$p(\mathbf{r}_{0:k} | \mathbf{y}_{0:k}) = \frac{p(\mathbf{y}_{0:k} | \mathbf{y}_{0:k-1}, \mathbf{r}_{0:k}) p(r_k | r_{k-1})}{p(\mathbf{y}_k | \mathbf{y}_{0:k-1})} p(\mathbf{r}_{0:k-1} | \mathbf{y}_{0:k-1}) \quad (2.102)$$

provides a useful recursion for the estimation of $p(\mathbf{r}_{0:k} | \mathbf{y}_{0:k})$ with the generated N_p particles $\{\mathbf{r}_{0:k}^i\}_{i=1}^{N_p}$ at time k . An importance function of the form $q(r_k | \mathbf{y}_{0:k}, \mathbf{r}_{0:k-1})$ is required. Suppose at time $k-1$ we have set of particles $\{\mathbf{r}_{0:k-1}^i\}_{i=1}^{N_p}$ that characterizes the pdf $p(\mathbf{r}_{0:k-1} | \mathbf{y}_{0:k-1})$. Drawing N_p samples $r_k^i \sim q(r_k | \mathbf{y}_{0:k}, \mathbf{r}_{0:k-1}^i)$ will be sufficient to represent the mode distribution as in Eq.(2.103),

$$p(\mathbf{r}_{0:k} | \mathbf{y}_{0:k}) \approx \sum_{i=1}^{N_p} w_k^i \gamma(\mathbf{r}_{0:k} - \mathbf{r}_{0:k}^i), \quad (2.103)$$

where the weight is

$$w_k^i \propto \frac{p(\mathbf{y}_k | \mathbf{y}_{0:k-1}, \mathbf{r}_{0:k}^i) p(r_k^i)}{q(r_k^i | \mathbf{y}_{0:k}, \mathbf{r}_{0:k-1}^i)} \quad (2.104)$$

It shown in (Doucet, 1998) that optimal importance density which minimize the variance of weights is $p(r_k | \mathbf{y}_{0:k}, \mathbf{r}_{0:k-1}^i)$. Optimal importance density can be represented as,

$$p(r_k | \mathbf{y}_{0:k}, \mathbf{r}_{0:k-1}^i) = \frac{p(\mathbf{y}_k | \mathbf{y}_{0:k-1}, \mathbf{r}_{0:k-1}^i, r_k) p(r_k | r_{k-1}^i)}{p(\mathbf{y}_k | \mathbf{y}_{0:k-1}, \mathbf{r}_{0:k-1}^i)}. \quad (2.105)$$

In this form denominator is independent of r_k and $p(r_k | r_{k-1})$ is the Markov transition probability defined between modes. The $p(r_k | \mathbf{y}_{0:k}, \mathbf{r}_{0:k-1}^i)$, featuring the denominator

is the only term that restricts to sample the optimal distribution. Hopefully, it can be estimated by one step ahead EKF estimator. Weight update equation using Eq.(2.105) as the proposal distribution is achieved by

$$w_k^i \propto p(\mathbf{y}_k | \mathbf{y}_{0:k-1}, \mathbf{r}_{0:k-1}^i) = \sum_{j=1}^s p(\mathbf{y}_k | \mathbf{y}_{0:k-1}, \mathbf{r}_{0:k-1}^i, r_k = j) p(r_k = j | r_{k-1}^i). \quad (2.106)$$

Since $r_k \in \{1, \dots, S\}$ S one step ahead EKF innovations is required to compute the importance weights.

The performance of MMPF, IMM-EKF and JMS-PF has been investigated by Arulampalam et al. (2004). EKF algorithm has been unified with IMM and JMS-PF to estimate the posterior conditioned on the mode sequence. Comparison of bearing only single target tracking under non-cluttered environment has been evaluated and for comparison MMSE of each algorithm is calculated for a certain scenario. They have reported MMPF is superior compared to JMS-PF and IMM-EKF. Among the other methods IMM-EKF was inferior to others while JMS-PF has modest performance.

2.8.5 Variable rate particle filters

Variable rate particle filters (VRPF) mostly rely on the semi-Markov model framework. VRPF introduced by Godsill and Vermaak (2005) proposes an efficient generalized nonlinear inference scheme using particle filters in a semi-Markov model.

Standard particle filters assume states are associated each other in a Markov process and the state sampling rate is determined with the rate measurements arrive. However in real data sets trajectories are characterized prolonged period of smoothness with infrequent sharp changes. This is specially true for manned targets which moves along a course for a while and then change its direction according to intercept new heading. Thus, this representation makes it is possible to achieve a parsimonious representation of the target trajectory if the state sampling is adapted to nature of the data. Variable rate scheme allocates more state points to regions of rapid deviations and fewer points to smooth trajectories thus models the times when maneuver occurs and the parameter of this maneuver.

In contrast to the fixed and known sampling rate of standard tracking techniques, Variable Rate particle filter models the state arrival times as a Markovian random process. This allows time between consecutive target states to be variable thus number

of target states and their timings are unknown. Although not necessary in a typical scenario number of state point will be much lower than the number of observations. An interpolation function is constructed to match the observation with variable state sequence and formulate the likelihood. The model assumes likelihood function is only dependent with the local neighboring state points to lead an efficient recursive implementation.

2.8.5.1 Variable rate models

In standard state-space models a state variable \mathbf{x}_t evolves with time index t . Diversely, variable rate state is defined as $\mathbf{x}_k = (\tau_k, \theta_k)$ where k is a discrete index, $\tau_k > \tau_{k-1} > \dots > \tau_0$ denotes the state arrival time for state k and θ_k denotes the vector of variables necessary to parameterize the target state. In the tracking application state vector will include variables like position, velocity, heading etc.

Variable state sequence follows a Markovian process such that states are independently generated according to density function in Eq.(2.107),

$$\mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}) = p(\theta_k | \theta_{k-1}, \tau_k, \tau_{k-1}) p(\tau_k | \theta_{k-1}, \tau_{k-1}) \quad (2.107)$$

where we constraint that $\tau_k > \tau_{k-1}$ and finite. Measurement vectors \mathbf{y}_t relating to time t are assumed to occur on a regular time grid and in correct time ordering although this assumption can be removed for cases of irregular sampling or out of sequence measurements.

In standard models, timings of the state process and the measurements do match, so that each measurement \mathbf{y}_t is uniquely associated with a state variable \mathbf{x}_t . However, in a variable rate model state timing can be asynchronous with the measurement process. The rate of the measurement arrival will be typically higher than that of the state process although not necessary. In need, a likelihood is defined with the assumption that \mathbf{y}_t is independent of all other data points, conditionally upon the neighborhood of states $\mathbf{x}_{\mathcal{N}_t} = \{\mathbf{x}_k; k \in \mathcal{N}_t(\mathbf{x}_{0:\infty})\}$. Note that each neighborhood $\mathcal{N}_t(\mathbf{x}_{0:\infty})$ is constructed as a deterministic function of the time index t and the state sequence $\mathbf{x}_{0:\infty}$. Thus it is a random variable itself, a feature that is not present in standard state space models. Throughout the document $\mathcal{N}_t(\mathbf{x}_{0:\infty})$ will be denoted by \mathcal{N}_t . Consequently likelihood is represented as a density distribution $p(\cdot)$ for consecutive t as,

$$\mathbf{y}_t \sim p(\mathbf{y}_t | \mathbf{x}_{0:\infty}) = p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) \quad (2.108)$$

For practical computational purposes the neighborhood will need to have a local structure and \mathcal{N}_t will contain only states whose times τ_k are close to the observation time t . Neighborhood structure will often be determined directly by logical or physical considerations of the model. Interpolated state vector $\hat{\theta}_t = f_t(\mathbf{x}_{\mathcal{N}_t})$, which is deterministic function $f_t(\cdot)$ of the states in the neighborhood $\mathbf{x}_{\mathcal{N}_t}$, is defined that leads the calculation of the observation density at time index t . Observation density is then expressed as

$$p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) = p(\mathbf{y}_t | \hat{\theta}_t) \quad (2.109)$$

which allows pointwise evaluation of the likelihood $p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t})$ and the process. It will usually be natural to arrange for the neighborhood to be “strictly monotonic” with t . Largest and smallest elements of \mathcal{N}_t is defined in Eq.(2.110) as,

$$\mathcal{N}_t^+ = \max(\mathcal{N}_t) \quad \mathcal{N}_t^- = \min(\mathcal{N}_t) \quad (2.110)$$

and for monotonicity assumed that,

$$\mathcal{N}_t^+ \geq \mathcal{N}_{t-1}^+, \quad \mathcal{N}_t^- \geq \mathcal{N}_{t-1}^- \quad (2.111)$$

Although the monotonicity assumption is not required for a valid estimation procedure, it is preferred as it simplifies the notation. Combining the Markovian assumptions it is possible to represent the joint density of states and observations as follows:

$$p(\mathbf{x}_{0:K}, \mathbf{y}_{0:T}) = p(\mathbf{x}_0) \prod_{k=1}^K p(\mathbf{x}_k | \mathbf{x}_{k-1}) \prod_{t=0}^T p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}), \quad K \geq \mathcal{N}_T^+ \quad (2.112)$$

where the condition $K \geq \mathcal{N}_T^+$ ensures the “complete” neighborhood for the calculation of the observation density at the final time index T . For simplicity, K will be considered as a constant “sufficiently” large that \mathcal{N}_T^+ can never exceed it. This is an important concept in construction of a sequential estimation procedure that will be detailed in subsequent paragraphs. Another requirement to determine neighborhood properly is that, the neighborhood \mathcal{N} will not be modified by addition of future state points $\mathbf{x}_{\mathcal{N}_t^+}$. Thus for each time t for which an observation \mathbf{y}_t is available we can always simulate a sequence of new states from the dynamic model until such a time as a valid neighborhood \mathcal{N}_t has been obtained. If k state will be simulated one has to wait until $k = \mathcal{N}_t^+$. In a target tracking application neighborhood is constructed as $\mathcal{N}_t = \{k, k-1\}$ of which provides simplicity in update equations.

2.8.5.2 Variable rate state estimation using particle filters

Purpose in this section is to derive an sequential algorithm that estimates the sequence of variable rate state points as new measurements become available. The conditional distribution referred as variable rate filtering distribution, written as

$$p(\mathbf{x}_{0:\mathcal{N}_t^+}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) \quad (2.113)$$

where $\mathbf{y}_{0:t} = (\mathbf{y}_0, \dots, \mathbf{y}_t)$ is the sequence of available measurements up to time t and $\mathbf{x}_{0:k} = (\mathbf{x}_0, \dots, \mathbf{x}_k)$ denotes a sequence of k hidden state variables. Variable rate filtering distribution is interpreted as a joint distribution over the number of state points \mathcal{N}_t^+ and their values $\mathbf{x}_{0:\mathcal{N}_t^+}$. \mathcal{N}_t^+ is a random variable thus variable rate distribution has variable dimension support. For recursive state estimation an update rule of the form,

$$p(\mathbf{x}_{0:\mathcal{N}_{t-1}^+}, \mathcal{N}_{t-1}^+ | \mathbf{y}_{0:t-1}) \rightarrow p(\mathbf{x}_{0:\mathcal{N}_t^+}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) \quad (2.114)$$

is required. As the new measurement is received, variable rate distribution at the current time step will be updated. Filtering distribution at time t is related to that at time $t - 1$ according to the Eq.(2.115) obtained using Bayes rule.

$$p(\mathbf{x}_{0:\mathcal{N}_t^+}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) = \frac{p(\mathbf{x}_{0:\mathcal{N}_{t-1}^+}, \mathcal{N}_{t-1}^+ | \mathbf{y}_{0:t-1}) p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) p(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+})}{p(\mathbf{y}_t | \mathbf{y}_{0:t-1})} \quad (2.115)$$

Note that Eq.(2.115) is derived under the assumption that, new state points beyond \mathcal{N}_{t-1} does not alter the neighborhoods from times 0 to $t - 1$. Sequential update rule is similar to standard particle filtering except that number of states to represent the trajectory is an unknown random variable of which is a very important issue when developing recursive numerical techniques. An important property of variable rate structure is that neighborhood may not need to increase at every time step, which implies $\mathcal{N}_t^+ = \mathcal{N}_{t-1}^+$ for some or many values of t . In this case the term $p(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+})$ disappears from the update expression.

Variable rate structure is analytically intractable for most models of practical interest therefore sequential Monte Carlo methods provide an efficient update rule. However standard particle filtering algorithm can not be applied to the update rule given by Eq. (2.115) owing to variable dimensionality of the hidden state sequence at each time index t . An intuitive method is applied by the authors (Godsill and Vermaak, 2005) to obtain a fixed dimensional state sequence. They assume that state sequence

comprise fixed number of state points K , that extend beyond the current time horizon so that neighborhood including \mathcal{N}_t can be unambiguously completed. K is chosen as an arbitrarily large number that satisfies the constraint. Value of K does not affect the final algorithm and having set up a fixed dimensional problem, sequential importance sampling can be applied to the variable rate state distribution. Fixed dimensional target distribution can be factorized as follows,

$$p(\mathbf{x}_{0:K}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) = p(\mathbf{x}_{0:\mathcal{N}_t^+}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) \pi(\mathbf{x}_{\mathcal{N}_t^++1:K} | \mathbf{x}_{0:\mathcal{N}_t^+}) \quad (2.116)$$

thus by the construction, desired variable rate filtering distribution is the marginal with respect to $\mathbf{x}_{\mathcal{N}_t^++1:K}$. The conditional distribution extends the variable state sequence to the fixed horizon K , and its choice is arbitrary. Now we can define the sequential importance sampler assuming that we have large number N_p of weighted Monte Carlo samples approximating the variable rate filtering distribution at the previous time step as in Eq.(2.117).

$$\left\{ \mathbf{x}_{0:\mathcal{N}_{t-1}^+}^{(i)}, w_{t-1}^{(i)} \right\}_{i=1}^{N_p}, \quad \sum_{i=1}^{N_p} w_{t-1}^{(i)} = 1 \quad (2.117)$$

and the weight update is,

$$w_t = \frac{p(\mathbf{x}_{0:K} | \mathbf{y}_{0:t})}{q(\mathbf{x}_{0:K} | \mathbf{y}_{0:t})} \quad (2.118)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) p(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}) p(\mathbf{x}_{0:\mathcal{N}_{t-1}^+}, \mathcal{N}_{t-1}^+ | \mathbf{y}_{0:t-1}) \pi(\mathbf{x}_{\mathcal{N}_t^++1:K} | \mathbf{x}_{0:\mathcal{N}_t^+})}{p(\mathbf{y}_t | \mathbf{y}_{0:t-1}) q(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}, \mathbf{y}_{0:t}) q(\mathbf{x}_{0:\mathcal{N}_{t-1}^+}, \mathcal{N}_{t-1}^+ | \mathbf{y}_{0:t}) \pi(\mathbf{x}_{\mathcal{N}_t^++1:K} | \mathbf{x}_{0:\mathcal{N}_t^+})} \quad (2.119)$$

$$\propto \frac{p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) p(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}) p(\mathbf{x}_{0:\mathcal{N}_{t-1}^+}, \mathcal{N}_{t-1}^+ | \mathbf{y}_{0:t-1})}{q(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}, \mathbf{y}_{0:t}) q(\mathbf{x}_{0:\mathcal{N}_{t-1}^+}, \mathcal{N}_{t-1}^+ | \mathbf{y}_{0:t})} \quad (2.120)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) p(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}) w_{t-1}}{q(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}, \mathbf{y}_{0:t}) v_{t-1}} \quad (2.121)$$

According to weight update rule in (2.121) particles are extended to a fix horizon K by first sampling from proposal distribution,

$$q(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}, \mathbf{y}_{0:t}) \quad (2.122)$$

that generates randomly a new complete neighborhood of state points $\mathbf{x}_{\mathcal{N}_t^+}$ and then complete the fixed horizon state space by drawing a sample form extension distribution,

$$q(\mathbf{x}_{\mathcal{N}_t^++1:K} | \mathbf{x}_{0:\mathcal{N}_t^+}) = \pi(\mathbf{x}_{\mathcal{N}_t^++1:K} | \mathbf{x}_{0:\mathcal{N}_t^+}). \quad (2.123)$$

However in practice one won't need to extend the state variables to the fixed horizon K since marginalization of a sampled joint random variable is utilized just by discarding the components which are not of interest. Further, weight update can be simplified by choosing the proposal distribution as prior. Similar to standard particle filter choosing proposal as in Eq.(2.124),

$$q(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}, \mathbf{y}_{0:t}) = p(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}) \quad (2.124)$$

results the variable rate of the standard bootstrap filter,

$$w_t \propto \frac{w_{t-1}}{v_{t-1}} p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}). \quad (2.125)$$

which approximates the variable rate state distribution as,

$$p(\mathbf{x}_{0:\mathcal{N}_t^+}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) = \sum_{i=1}^{N_p} w_t^i \delta(\mathbf{x}_{0:\mathcal{N}_t^+}^i - \mathbf{x}_{0:\mathcal{N}_t^+}). \quad (2.126)$$

2.8.5.3 Variable rate particle filtering algorithm

1. Initialize the VRPF state distribution $p(\mathbf{x}_0)$ at time $t = 0$ by drawing N_p samples from pre-defined initial distribution. The collection of samples are obtained sampling from the transition distribution $\dots p(\mathbf{x}_2 | \mathbf{x}_1) p(\mathbf{x}_1 | \mathbf{x}_0)$, until each particle contains a valid neighborhood $\mathbf{x}_{\mathcal{N}_0}$. Then initialized samples $\mathbf{x}_{\mathcal{N}_0}^i$ are equally weighted,

$$w_{t=0}^i = 1/N_p, \text{ for } i = 1 \dots N_p. \quad (2.127)$$

2. For $t = 1 \dots T$ where T is the index to last observation, sample from the proposal distribution defined in Eq.(2.124) which is generated according to Eq.(2.128) as,

$$\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+}^i \sim p\left(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}^i\right), \quad i = 1 \dots N_p \quad (2.128)$$

to complete the neighborhood of observation \mathbf{y}_t , and set state index $k \leftarrow k + 1$. Note that generation of particles are needed to complete the neighboring N_t at time t thus no particles will be generated when the neighborhood is already complete. More precisely a single step, where new state k is generated from the previous state $k - 1$ is shown as in Eq.3.6,

$$\mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}) = p(\theta_k | \theta_{k-1}, \tau_k, \tau_{k-1}) p(\tau_k | \theta_{k-1}, \tau_{k-1}). \quad (2.129)$$

In the original work, Godsill et al. (Godsill and Vermaak, 2005) adapted curvilinear model (CL), in VRPF framework and defined target state vector \mathbf{x}_k as follows,

$$\mathbf{x}_k = [T_{T,k} \ T_{P,k} \ v(\tau_k) \ \psi(\tau_k) \ \mathbf{z}(\tau_k) \ \tau_k] \quad (2.130)$$

where $T_{T,k}, T_{P,k}$ are independent tangential and perpendicular forces applied to target, $v(\tau_k), \psi(\tau_k)$ are the target speed and the course, $\mathbf{z}(\tau_k) = [x \ y]$ is the position vector constituting position in x and y coordinates, and the last variable is the arrival time τ_k for state k that is statistical conditioned on the previous arrival time τ_{k-1} .

Sampling state vector \mathbf{x}_k according to conditional distribution Eq.(2.129) is expressed in Eq.(2.131),

$$\begin{aligned} T_{T,k} &\sim N(\mu_t, \sigma_t^2), \quad T_{P,k} \sim N(0, \sigma_p^2) \\ \tau_k - \tau_{k-1} - \tau_{min} &\sim G(\alpha_\tau, \beta_\tau), \quad \tau_{min} > 0 \end{aligned} \quad (2.131)$$

where N is Gauss, G is Gamma distribution and τ_{min} is an operator to shift the gamma distributed sojourn time. $T_{T,k}$ and $T_{P,k}$ assumed independently distributed forces applied to target at state index k . Hyperparameters of these distributions are chosen to match the characteristics of object being tracked. The rest of required variables which are, position $\mathbf{z}(\tau_k)$, speed $v(\tau_k)$, and course $\psi(\tau_k)$ can be determined from the previous state variable \mathbf{x}_{k-1} with a deterministic function defined by curvilinear motion equations.

3. Update the weights associated with sample points $\{w_t^i, \mathbf{x}_{N_t}^i\}_{i=1}^{N_p}$ that approximates the filtering distribution,

$$p(\mathbf{x}_{N_t}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}). \quad (2.132)$$

Note that Eq.(2.132) is the marginalized version of the filtering distribution Eq.(2.113). In Variable Rate particle filtering framework weights w_k^i for each particle ($i = 1 \dots N_p$) are updated according to defined weight update equation expressed in Eq.(2.133) as,

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{y}_t | \mathbf{x}_{N_t}) p(\mathbf{x}_{N_{t-1}^+ + 1 : N_t^+} | \mathbf{x}_{N_{t-1}^+})}{q(\mathbf{x}_{N_{t-1}^+ + 1 : N_t^+} | \mathbf{x}_{N_{t-1}^+}, \mathbf{y}_{0:t})}. \quad (2.133)$$

Similar to standard particle filtering methods, simplest strategy to define the proposal distribution q is to use the prior distribution defined in Eq.(2.134),

$$p(\mathbf{x}_{\mathcal{N}_{t-1}^+ + 1 : \mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}) \quad (2.134)$$

as the proposal density. This results in a simple weight update equation in the form of Eq.(2.125) as

$$w_t^i \propto w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) \quad (2.135)$$

where we just have to evaluate the likelihood $p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t})$ to update the particle weights w_t . To express the likelihood in an analytic form, it can be redefined according to Eq.(2.109) as follows,

$$p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) = p(\mathbf{y}_t | \hat{\theta}_t) \quad (2.136)$$

using interpolated continuous time state process $\hat{\theta}_t$ at time t which is a deterministic function of $\mathbf{x}_{\mathcal{N}_t}$ where $\hat{\theta}_t = f_t(\mathbf{x}_{\mathcal{N}_t})$. The function $f_t(\cdot)$ is defined by the curvilinear motion equations, obtaining the continuous process at time t from the marked point processes $\mathbf{x}_k = (\theta_k, \tau_k)$. Resultantly the state vector $\hat{\theta}_t$ constitutes the interpolated variables of $\mathbf{x}_{\mathcal{N}_t^+}$ required to evaluate the likelihood function $p(\mathbf{y}_t | \hat{\theta}_t)$. In tracking applications it is suitable to choose $\hat{\theta}_t$ as the interpolated position vector $\mathbf{y}(t)$. New weights for each particle ($i = 1 \dots N_p$) are updated according to the update function

$$w_t^i \propto w_{t-1}^i p(\mathbf{y}_t | \hat{\theta}_t), \quad (2.137)$$

and weights are renormalized such that,

$$\sum_{i=1}^{N_p} w_t^i = 1. \quad (2.138)$$

4. Following the weight update procedure, calculate the estimate of effective sample size \widehat{N}_{eff} (Liu and Chen, 1998) as in Eq.(2.139).

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p} (w_t^i)^2} \quad (2.139)$$

If \widehat{N}_{eff} is below a pre-specified threshold than resample the approximate filtering distribution,

$$\{\mathbf{x}_{\mathcal{N}_t}^i, w_t^i\}_{i=1}^N \propto p(\mathbf{x}_{\mathcal{N}_t}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) \quad (2.140)$$

with a certain resampling scheme to form the replicated samples,

$$\left\{ \mathbf{x}_{0:\mathcal{N}_t^+}^i, w_t^i \right\}_{i=1}^N \longleftarrow \text{Resample} \left[\left\{ \mathbf{x}_{0:\mathcal{N}_t^+}^i, w_t^i \right\}_{i=1}^N \right] \quad (2.141)$$

else return to step 2 to evaluate the next observation.

3. MULTIPLE MODEL TARGET TRACKING WITH VARIABLE RATE PARTICLE FILTERS

Maneuvering target tracking has taken much attention in the past decade with the development of numerous numerical techniques. Multiple model (MM) approaches which characterize target motion dynamics with a set of models, have been the most widely used techniques in the field (Blom et al., 1998; McGinnity and Irwin, 2000; Doucet et al., 2001b). In MM modeling, a target state posterior at time t can be represented as the weighted sum of several parallel filter outputs, each representing a different target mode. However, the number of filters required to obtain the optimal solution increases exponentially as time evolves, hence it is common to use a MM structure therefore provides a suboptimal solution (Blom et al., 1998).

Kalman filter based interacting multiple model (IMM) is one of the commonly used suboptimal solution to the maneuvering target tracking problem (Blom et al., 1998; Kirubarajan et al., 2001). IMM approximates the weighted output of each filter with a single Gaussian distribution thus fixes the number of required filters. However due to the nature of the maneuvering target tracking problem, it is desirable to represent the state posterior with multi modal complex density function which reduce the effectiveness of the suboptimal solutions such as IMM. Therefore, an extension of the particle filters to the multiple model estimation problem has been proposed and is referred as multiple model particle filter (MM-PF) (McGinnity and Irwin, 2000). Since the particle filter solution is not restricted to a particular distribution, the MM-PF allows approximating to the non-linear posterior distribution of the multiple model state space and improve the tracking performance when compared to Kalman filter based IMM filters (Arulampalam et al., 2004).

Conventional tracking models, including the particle filter and Kalman filter based approaches, define a discrete time state space model where the state sampling rate is determined by the rate at which the measurements arrive, thus known as fixed rate models. In fixed rate models, the timing of the state variables (state arrival

time) is deterministically defined by the observation time and the time between two consecutive states (sojourn time) is fixed. The multiple model structure defined for fixed rate models denote the target motion mode by a time indexed latent variable and under the assumption that a transition may occur at each observation time, mode variables are represented as a first order Markov chain. However it is known that a manned target often executes straight motion followed by a short duration of sharp maneuver. Therefore, unlike the fixed rate standard tracking models, recently introduced variable rate particle filters (VRPF) model the state arrival times as a Markov random process that enables the time between consecutive target states to be a random variable. In contrast to the fixed rate modeling of the conventional methods, the VRPF tracks a maneuvering object with a small number of states by imposing a probability distribution on the state arrival times. In literature, variable rate models have been considered as a more effective technique in maneuvering target tracking problems even when compared to well known particle filter based IMM models (Godsill and Vermaak, 2005; Godsill et al., 2007).

The variable rate structure proposed by Godsill and Vermaak (2005) utilizes curvilinear motion dynamics that enable tracking a wide range of motion characteristics (Li and Jilkov, 2003). However, similar to the conventional models, depending on its parameterization, the VRPF with its single mode structure suffers from the poor estimate of the target trajectory and the state arrival times. In order to overcome this drawback, we introduce a multiple model variable rate particle filtering (MM-VRPF) scheme that integrates the multiple model structure with the variable rate filtering. Independent of the observation time, the MM-VRPF allows switching between candidate sojourn and motion parameter sets thus can precisely model the maneuver parameters as well as the state arrival times. The introduction of different sojourn modes enables a parsimonious representation for smooth regions of the trajectory while it adaptively locates frequent state points at high maneuver regions, resulting in a much more accurate tracking.

On the other hand, it is possible to construct an autonomous target tracking model where the number of modes and mode parameters are estimated under the Bayesian framework. In the literature the Dirichlet Process Mixtures (DPM) model has been used as a key building block particularly in modeling linear dynamic systems

with unknown model structure (Caron et al., 2008; Teh et al., 2006; Fox et al., 2007). Similarly, It is possible to construct a hierarchical Dirichlet process mixture model that can be considered as a prior for the semi-Markovian structured tracking model (Teh et al., 2006). However, online inference becomes a complicated task and very efficient sequential Monte Carlo schemes are required to accomplish the parameter estimation process for such models (Ulker et al., 2008, 2010b,a). Therefore in this thesis we particularly focus on multiple model approach that leads to simpler inference schemes based on conventional particle filters.

In our modeling, we preserve the continuous deterministic process proposed by Godsill and Vermaak (2005) while adapting the multiple model structure to the variable rate framework. In order to classify the type of the maneuvering objects, two sojourn distributions having different driving noise processes are defined by Maskell (2004). In our model we are also proposing usage of different sojourn distributions, but unlike Maskell (2004), our objective is efficiently modeling the parameters and arrival times of the maneuver rather than the noise.

In order to overcome the degeneracy problem which is a well known drawback of particle filtering we have also presented a regularization scheme for variable rate models. In a single target tracking problem, degeneracy is particularly observed if the unknown parameters are defined in a complex state space or when the parameters of the model do not fit the actual model, especially under low process noise. Regularization can be interpreted as sampling from a continuous probability density function instead of a discrete probability mass space as in the case of a conventional particle filter resampling mechanism (Musso et al., 2001). In order to eliminate degeneracy, we have adapted well known regularized fixed rate particle filtering (R-PF) scheme to the introduced MM-VRPF structure and named the new filter as regularized multiple model variable-rate particle filter (RMM-VRPF).

Test results reported for bearings only scenarios demonstrate that the proposed structure finely locates the target states onto critical maneuver change points thus improves the VRPF's trajectory estimation performance. It is shown that regularization significantly improves the tracking performance when the initial values are specified as far from the true values thus resulting in degeneracy.

This section is organized as follows. In Section 3.1 a summary of variable rate models is presented. Section 3.2 introduces the theoretical formulation of the proposed multiple model variable rate model. Section 3.3 explains the state posterior estimation of the proposed model and a regularization scheme is described in Section 3.4. The test results are given in Section 3.5.

3.1 Variable Rate Tracking

The conventional fixed rate state-space model proposes a state variable \mathbf{x}_t that evolves with time index t (Godsill and Vermaak, 2005). Let $\mathbf{x}_k = (\tau_k, \theta_k)$ defines a variable rate state where k is a discrete index, τ_k denotes the state arrival time for state k and θ_k denotes the vector of variables that parameterizes the target state. Variable state sequence follows a Markovian process such that states are independently generated according to a density function, $\mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1})$ where τ_k is finite and $\tau_k > \tau_{k-1}, \tau_{k-2}, \dots$

Unlike the fixed rate models, allocation of states in a variable rate model is asynchronous with the timing of observations. Therefore, tractable solution to the inference problem can only be obtained if the likelihood function depends on the local neighborhood of the states. Indeed, under the assumption that an observation \mathbf{y}_t is independent of all other data points except the neighborhood states $\mathbf{x}_{\mathcal{N}_t} = \{\mathbf{x}_k; k \in \mathcal{N}_t(\mathbf{x}_{0:\infty})\}$, the likelihood function at time t is defined as,

$$p(\mathbf{y}_t | \mathbf{x}_{0:\infty}) = p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}). \quad (3.1)$$

Practically, the neighborhood \mathcal{N}_t comprises the states that are closest to the observation time t . In order to calculate the likelihood function, an interpolated continuous time state process, $\hat{\theta}_t$, is defined as a deterministic function of $\mathbf{x}_{\mathcal{N}_t}$ as $\hat{\theta}_t = f_t(\mathbf{x}_{\mathcal{N}_t})$ where the function $f_t(\cdot)$ is determined by the selected target motion model. Finally, the observation density can be rewritten as $p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}) = p(\mathbf{y}_t | \hat{\theta}_t)$.

The first member of a neighborhood set \mathcal{N}_t is represented with \mathcal{N}_t^+ and the last one is \mathcal{N}_t^- . For monotonicity assumed that, $\mathcal{N}_t^+ \geq \mathcal{N}_{t-1}^+$, $\mathcal{N}_t^- \geq \mathcal{N}_{t-1}^-$. Under the Markovian assumptions, the joint density of states and observations can be expressed as in Eq.(3.2)

$$p(\mathbf{x}_{0:K}, \mathbf{y}_{0:T}) = p(\mathbf{x}_0) \prod_{k=1}^K p(\mathbf{x}_k | \mathbf{x}_{k-1}) \prod_{t=0}^T p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}), \quad K \geq \mathcal{N}_T^+ \quad (3.2)$$

where the condition $K \geq \mathcal{N}_T^+$ ensures the “complete” neighborhood for the calculation of the observation density at the final time index T . The requirement to determine neighborhood properly is that, the neighborhood \mathcal{N}_t will not be modified by addition of future state points.

Let $\mathbf{y}_{0:t} = (\mathbf{y}_0, \dots, \mathbf{y}_t)$ and $\mathbf{x}_{0:\mathcal{N}_t^+} = (\mathbf{x}_0, \dots, \mathbf{x}_{\mathcal{N}_t^+})$ denote the observations received until time instant t and the corresponding target states, respectively. Note that, the total number of states, \mathcal{N}_t^+ , is a random variable in the variable rate model, therefore the posterior distribution can only be expressed in transdimensional form. At each time instant t , the variable rate optimal filtering distribution can be expressed as a combination of N_p multi-dimensional Dirac delta functions as shown in Eq.(3.3),

$$p(\mathbf{x}_{0:\mathcal{N}_t^+}, \mathcal{N}_t^+ | \mathbf{y}_{0:t}) \approx \sum_{i=1}^{N_p} w_t^i \delta(\mathbf{x}_{0:\mathcal{N}_t^+}^i - \mathbf{x}_{0:\mathcal{N}_t^+}) \quad (3.3)$$

where w_t^i is the weight associated to i 'th particle $\mathbf{x}_{0:\mathcal{N}_t^+}^i$.

Approximation to the posterior density given by Eq.(3.3) is achieved by sampling from the proposal density $q(\cdot)$ and updating the particle weights w_{t-1}^i according to Eq.(3.4) whenever a new observation arrives.

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t^+}^i) p(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+}^i | \mathbf{x}_{\mathcal{N}_{t-1}^+}^i)}{q(\mathbf{x}_{\mathcal{N}_{t-1}^++1:\mathcal{N}_t^+}^i | \mathbf{x}_{\mathcal{N}_{t-1}^+}^i, \mathbf{y}_{0:t})} \quad (3.4)$$

3.2 Multiple Model Variable Rate Model

The conventional variable rate model defines the target motion and state arrival times with a single motion model. However when a maneuver is undertaken or a straight motion is in progress, the motion parameters and arrival times always show diversity due to the nature of maneuvering target tracking problem. Thus it is often not suitable to estimate the state arrival times and maneuver parameters with a single model. To overcome the drawbacks of the single motion model, we proposed a multiple model structured variable rate scheme in which the maneuver parameters and arrival times are modeled by three different parameter sets. The proposed model combines the advantage of adaptive motion estimation with variable rate filtering thus is expected to improve the target tracking performance.

Unlike the conventional variable rate model, the multiple model approach introduces a new discrete state variable m_k that denotes the mode of the motion dynamics, as shown

in Eq.(3.5),

$$\mathbf{x}_k = [\theta_k, \tau_k, m_k], \quad m_k \in \{1, \dots, r\} \quad (3.5)$$

where r refers to the total number of modes. In a target tracking scenario it is often appropriate to specify a set of dynamic modes where each mode is parameterized to handle different target motion characteristics such as a straight motion or the maneuvers. Therefore we defined three, $r = 3$, distinct target motion modes representing the straight motion and the maneuvers undertaken to each side. In the expense of computational complexity the number of modes can be increased, however it is known that increasing the number of modes do not necessarily improve the algorithm performance in a multiple model structure (Li and Jilkov., 2005).

Proposed structure yields the conditional state distribution expressed in Eq.(3.6),

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{x}_{k-1}) &= p(\theta_k | \theta_{k-1}, \tau_k, \tau_{k-1}, m_{k-1}, m_k) \\ &\quad p(m_k | m_{k-1}) p(\tau_k | \tau_{k-1}, m_{k-1}). \end{aligned} \quad (3.6)$$

According to the curvilinear motion model it is convenient to define the motion kinematics vector θ_k by the vector $[T_{T,k} \ T_{P,k} \ v(\tau_k) \ \psi(\tau_k) \ \mathbf{z}(\tau_k)]$ (Godsill and Vermaak, 2005). The tangential, $T_{T,k}$ and perpendicular, $T_{P,k}$ forces applied to target can be simply modeled with Gauss distributions as given in Eq.(3.7),

$$\begin{aligned} T_{T,k} &\sim N(\mu_{T,n}, \sigma_{T,n}^2) \\ T_{P,k} &\sim N(0, \sigma_{P,n}^2) \end{aligned} \quad (3.7)$$

where $n = m_k$, $m_k \in \{1 \dots r\}$ and the parameters, $\mu_{T,n}, \sigma_{T,n}^2, \sigma_{P,n}^2$ are specified to match the motion characteristics of the object being tracked. Note that the defined model has the ability to characterize a large set of maneuvers precisely when compared to a coordinated turn (CT) model driven with a process noise (Li and Jilkov, 2003). The rest of the state variables, i.e., the position $\mathbf{z}(\tau_k)$, speed $v(\tau_k)$, and course $\psi(\tau_k)$ are calculated from the previous state variable \mathbf{x}_{k-1} by using motion dynamics equations given in Section 2.8.1.4.

Manned target often undertakes sharp and short duration turns whereas smooth and straight motions are maintained for longer periods. Multiple model representation defines a set of Gamma distributions each parameterized to handle a different motion character. Thus unlike the single mode conventional variable rate model, the multiple

model structure allows modeling the sojourn times with r different distributions. The distribution defined on the sojourn time, $p(\tau_k|\tau_{k-1}, m_{k-1})$, is defined as a Gamma distribution where the parameters are selected according to the discrete mode variable. Dependent on the mode variable m_{k-1} , the sojourn times are distributed with a shifted Gamma distribution as,

$$\tau_k - \tau_{k-1} - \tau_n \sim G(\alpha_n, \beta_n), \quad (3.8)$$

where $n = m_{k-1}$, $m_{k-1} \in \{1 \dots r\}$, is the index to the motion mode, and τ_n is the sojourn time shifting parameter.

Proposed multiple model structure requires the representation of the transition matrix, more specifically $p(m_k|m_{k-1})$ shown in Eq.(3.6). Transition matrix represents the transition from a motion mode to another and the probability of staying at a particular mode thus it is an important task to define an appropriate transition matrix. The proposed multiple model structure defines the mode transition probability, $p(m_k|m_{k-1})$, by a time invariant mode transition probability matrix \mathbf{P} ,

$$\mathbf{P} = \begin{bmatrix} p_{11} & \dots & p_{1r} \\ \vdots & \ddots & \vdots \\ p_{r1} & \dots & p_{rr} \end{bmatrix} \quad (3.9)$$

where each element, p_{hl} , $\{h, l\} \in \{1, \dots, r\}$ denotes the transition probability from mode h to mode l .

In practice, the transition matrix \mathbf{P} is intuitively selected according to the maneuver characteristics of the object being tracked unless it is assumed to be known a priori. In our model we determined an appropriate transition matrix considering the properties of the target motion characteristics.

It is very difficult to determine a suitable transition matrix generalizing all types of maneuvers. Some techniques to determine the \mathbf{P} has been addressed by Blair and Watson (1992); Bloomer and Gray (2002). However these techniques assume that target sojourn times are known a priorly. A number of online transition matrix estimation methods has been proposed by Jilkov and Li (2004).

It is also possible to define a set of transition matrix or randomize it by using a non informative prior. In this case estimation of the transition matrix can be described as a parameter estimation problem and can be solved in a particle filter framework in the expense of model complexity .

If the target sojourn time τ of each motion mode are known priorly, diagonal elements of the transition matrix \mathbf{P} can be determined as follows. In a semi-Markov chain where sojourn time distribution is a Gamma distribution, $G(\alpha, \beta)$, the expected sojourn value is computed as $\mu_G = \alpha.\beta$. Consequently, τ , the expected amount of time spent at state h can be defined as in Eq.(3.10).

$$\tau = \frac{\mu_{G_n}}{1 - p_{hh}}. \quad (3.10)$$

Since the target sojourn time τ of each motion mode is known a priori, diagonal elements of \mathbf{P} can be computed by Eq.(3.10). Considering that the summation of each row should be equal to 1, off-diagonal elements of \mathbf{P} can be specified heuristically with respect to the prior belief on mode transitions. However prior information on sojourn time is often unavailable in target tracking problems thus this methodology can only help to determine an appropriate transition matrix intuitively.

Note that the proposed multiple model scheme do not change the constraints that apply on the neighborhood conditions therefore the likelihood function, as well as the joint distribution of the states and observations given in Eq.(3.2) is also valid for the proposed MM-VRPF algorithm.

3.3 Multiple Model Variable Rate State Estimation

In the following we will describe how to generate weighted samples that approximate to the variable rate model posterior distribution given by Eq.(3.2). Note that the algorithm given in this section is based on the variable rate particle filtering algorithm proposed by Godsill and Vermaak (2005) and valid for both MM-VRPF and VRPF algorithms.

In the first step all particles are initialized according to a predefined initial distribution. Next, the propagation step carries the particles forward in time if required. Finally, the update step calculates the sample weights for each particle that represent the model posterior distribution. A resampling step is performed if effective sample weights are below a certain threshold and the algorithm is iterated through step 2 whenever a new observation arrives. A summary of the algorithm is also described by a pseudo code given in Algorithm-1.

A. Initialization

At time $t = 0$, N_p samples are drawn from a predefined initial state distribution $p(\mathbf{x}_0)$ where \mathbf{x}_0 denotes the initial state vector of the target. It is assumed that the initialized samples \mathbf{x}_0^i , $i = 1 \dots N_p$, are equally weighted.

B. Propagation Step

At time step t , whenever a new observation arrives, for $i = 1 \dots N_p$, we generate N_p particles from the proposal distribution, $q(\cdot)$, that is chosen as the prior state distribution,

$$q(\mathbf{x}_{\mathcal{N}_{t-1}^+ + 1 : \mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}^i, \mathbf{y}_{0:t}) = p(\mathbf{x}_{\mathcal{N}_{t-1}^+ + 1 : \mathcal{N}_t^+} | \mathbf{x}_{\mathcal{N}_{t-1}^+}^i) \quad (3.11)$$

Note that we just need to generate new particles from the proposal distribution if the neighborhood \mathcal{N}_t of an observation \mathbf{y}_t is not complete. Therefore no particles will be generated when the neighborhood is already complete. More specifically, assuming that, $k = \mathcal{N}_{t-1}^+ + 1$, the $k - 1$ 'th state variable, x_{k-1}^i is propagated forward in time according to $x_k^i \sim p(x_k | x_{k-1}^i)$ until $\tau_k^i > t$ where $x_k = [\theta_k, \tau_k, m_k]$. The target motion state variable θ_k^i , the state arrival time τ_k^i , and the motion mode m_k^i , are sampled according to the equations defined in Eq.(3.7), Eq.(3.8) and Eq.(3.9) respectively.

C. Updating the Particle Weights

In this step, the particle weights w_t^i , $i = 1 \dots N_p$, are calculated according to the weight update equation that can be derived by replacing the selected proposal function in Eq.(3.4). When prior distribution is selected as the proposal, the bootstrap version of the variable rate particle filtering algorithm can be achieved as given in Eq.(3.12)

$$w_t^i \propto w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t}^i). \quad (3.12)$$

Finally, the true posterior $p(\mathbf{x}_{\mathcal{N}_t}, \mathcal{N}_t^+ | \mathbf{y}_{0:t})$, is approximated by the particles and associated weights, $\{\mathbf{x}_{\mathcal{N}_t}^i, w_t^i\}_{i=1}^{N_p}$. Following the weight update step, a resampling step is performed if the effective sample size, $\hat{N}_{eff} = 1 / \sum_{i=1}^{N_p} (w_t^i)^2$, is below a predefined threshold value.

In VRPF algorithm, following the resampling, multiple copies of the particles with identical state arrival times occur. This situation can result in poor estimates of the state arrival times and reduce the filter performance drastically. Therefore, under the

variable rate framework, a regeneration step is introduced that augments the latest arrival time of each particle fixing the past state timings (Godsill and Vermaak, 2005). However ability of the proposed multiple model scheme in modeling the state arrival times eliminates the necessity of this regeneration step.

Note that replicated particles with completed neighborhoods undergo the same weighting procedure for the explained variable rate algorithm. In order to reduce this redundancy, a modified algorithm is proposed by Godsill and Vermaak (2005) and more efficient particle filtering schemes that apply different proposal kernels are applied by Whiteley et al. (2007). However, for comparison purpose we only applied the basic variable rate particle filtering algorithm for both VRPF and MM-VRPF in our work .

Algorithm-1 : The VRPF and MM-VRPF algorithm

1. Initialization

- Set $t=0$
- For $i = 1$ to N_p , draw equally weighted samples, $x_0^i \sim p(\mathbf{x}_0)$, from the predefined initial state distribution.

2. Propagation step

- set $k = N_{t-1}^+$
- for $i = 1$ to N_p
 - While the neighborhood \mathcal{N}_t^i is incomplete
 - * Set $k = k + 1$ and draw samples form the proposal distribution, $x_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}, y_t)$, until $\tau_k \geq t$.

3. Weight update step

- calculate the particle weights according to

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{y}_t | \mathbf{x}_{\mathcal{N}_t^i}^i) p(\mathbf{x}_{\mathcal{N}_{t-1}^+ + 1 : \mathcal{N}_t^+}^i | \mathbf{x}_{\mathcal{N}_{t-1}^+}^i)}{q(\mathbf{x}_{\mathcal{N}_{t-1}^+ + 1 : \mathcal{N}_t^+}^i | \mathbf{x}_{\mathcal{N}_{t-1}^+}^i, \mathbf{y}_{0:t})}$$

- Normalize the weights.

4. Resampling step

- Resample the particles $\{\mathbf{x}_{\mathcal{N}_t}^i, w_t^i\}_{i=1}^{N_p}$ if effective sample size, $\hat{N}_{eff} = 1/\sum_{i=1}^{N_p} (w_t^i)^2$, is below a certain threshold.
- set $t = t + 1$
- iterate through step 2

3.4 Regularization

In particle filtering, degeneracy is an essential problem that causes variance of the particle weights decrease with time and results in a few particles to have non-zero importance weights. In order to reduce the effects of degeneracy, resampling is introduced as a solution that eliminates the particles that have small weights and concentrates on the the particles with large weights (Gordon et al., 1993; Arulampalam et al., 2002).

However resampling introduces new problems such as sample impoverishment that arises due to sampling from a discrete posterior distribution rather than a continuous one. If the problem is not addressed properly, it may lead to collapse of particles resulting in severe sample impoverishment.

In this section, we propose a regularization scheme for variable rate models in order to find a potential solution to the described problem. The regularization scheme guarantees approximation to the continuous filtering distribution with a kernel function and enables us to generate new samples from the approximated continuous distribution (Musso et al., 2001). We applied the regularization process for both, single and multiple model structure, and call the new algorithms as regularized variable rate particle filter (R-VRPF) and regularized multiple model variable rate particle filter (RMM-VRPF). Conventional fixed rate models perform the regularization on the particles approximating the marginal posterior of the latest state variable. However states are asynchronous in time for the variable rate models, hence we define a time indexed state variable and perform the regularization process on the defined state posterior.

In the following we will describe how to perform the regularization step for both VRPF and MM-VRPF algorithms. Recall that regularized algorithms execute same

steps with their conventional counterparts, except a regularization step is executed after resampling.

3.4.1 Regularized variable rate particle filters

Remark that, in variable rate models, states are asynchronous with the observations time, thus the states are indexed with k whilst t is the time index. Let us define a time indexed state variable vector \mathbf{r}_t that is a deterministic function of the neighborhood \mathcal{N}_t as expressed in Eq.(3.13),

$$\mathbf{r}_t = [T_{T,\tau_{\mathcal{N}_t^-}} \quad T_{P,\tau_{\mathcal{N}_t^-}} \quad v(t) \quad \psi(t) \quad \mathbf{z}(t) \quad \tau_{\mathcal{N}_t^+}] \quad (3.13)$$

where $T_{T,\tau_{\mathcal{N}_t^-}}$, $T_{P,\tau_{\mathcal{N}_t^-}}$ are tangential and perpendicular force parameters at neighborhood \mathcal{N}_t^- , and $v(t)$, $\psi(t)$, $\mathbf{z}(t)$ are velocity, course, position vector at time t and $\tau_{\mathcal{N}_t^+}$ is state arrival time at state \mathcal{N}_t^+ . Since we assume neighboring conditions of the past observations do not change with the arrival of a new observation, the variable $\tau_{\mathcal{N}_t^-}$ is not included into the definition of the state vector \mathbf{r}_t .

In target tracking applications to simplify the algorithm, at each observation time t only closest variable rate states are assumed as neighboring states according to $\mathcal{N}_t = (k, k-1, \tau_{k-1} < t \leq \tau_k)$ where k is state index. Therefore \mathbf{r}_t can be stated explicitly as in Eq.(3.14),

$$\mathbf{r}_t = [T_{T,k-1} \quad T_{P,k-1} \quad v(t) \quad \psi(t) \quad \mathbf{z}(t) \quad \tau_k] \quad (3.14)$$

Given the observation vector $\mathbf{y}_{0:t}$, the regularized distribution associated with the state vector \mathbf{r}_t is defined as in Eq.(3.15),

$$p(\mathbf{r}_t | \mathbf{y}_{0:t}) \approx \sum_{i=1}^{N_p} w_t^i K_b(\mathbf{r} - \mathbf{r}_t^i) \quad (3.15)$$

where

$$K_b(\mathbf{r}) = \frac{1}{b^{n_r}} K\left(\frac{\mathbf{r}}{b}\right) \quad (3.16)$$

is the rescaled Kernel density $K(\cdot)$.

In Eq.(3.16) $b > 0$ is the kernel bandwidth, n_r is the dimension of the state vector \mathbf{r}_t , and w_t^i , $i = 1 \dots N_p$, is the normalized particle weights. It is common to chose the kernel $K(\cdot)$ as a symmetric probability density function. In the special case of equally weighted samples, $w_t^i = 1/N_p$, $i = 1 \dots N_p$, it is appropriate to choose the

kernel as Epanechnikov kernel (Silverman, 1986). However due to the simplicity in generating samples we used Gaussian kernels. Under the assumption that underlying density is also Gaussian with unit covariance matrix, Eq.(3.17) defines the optimal kernel bandwidth (Silverman, 1986).

$$b_{opt} = AN_p^{\frac{-1}{n_r+4}}, \text{ where } A = (4/(n_r + 2))^{\frac{1}{n_r+4}}. \quad (3.17)$$

To cover the case of multimodal distributions, it is convenient to define the kernel bandwidth as $b = b_{opt}/2$ (Musso et al., 2001). To adjust the kernel covariance with respect to an empirical covariance matrix \mathbf{S}_t of the state vector \mathbf{r}_t , we compute the empirical covariance matrix

$$\mathbf{S}_t = \text{Cov} [\mathbf{r}_t \mathbf{r}_t^T] \quad (3.18)$$

where the matrix \mathbf{D}_t is extracted by decomposing the matrix \mathbf{S}_t according to $\mathbf{S}_t = \mathbf{D}_t \mathbf{D}_t^T$.

New samples from the regularized distribution given by Eq.(3.15) are obtained as expressed in Eq.(3.19)

$$\mathbf{r}_t^{*i} = \mathbf{r}_t^i + b \mathbf{D}_t \epsilon^i, \quad \epsilon^i \sim K_b(\mathbf{r}), \quad i = 1 \dots N_p \quad (3.19)$$

where ϵ^i is a sample of the kernel density estimate $K_b(\mathbf{r})$. Our claim in proposing a regularized scheme is that samples generated from the regularized distribution approximate the true posterior in a better way resulting in more accurate estimates. The regularized samples and associated weights represent the posterior distribution,

$$\{\mathbf{r}_t^i, w_t^i\}_{i=1}^N \propto p(\mathbf{r}_t | \mathbf{y}_{0:t}) \quad (3.20)$$

Finally the samples that represent the distribution $p(\mathbf{x}_{\mathcal{N}_t} | \mathbf{y}_{0:t})$ can be calculated deterministically. Since neighboring states $\mathbf{x}_{\mathcal{N}_t}$ at time t are deterministic function of \mathbf{r}_t , interpolated states can be obtained according to $\mathbf{x}_{\mathcal{N}_t} = f_t(\mathbf{r}_t)$, where function $f_t(\cdot)$ is specified with the curvilinear motion dynamic model. The interpolated particles $\mathbf{x}_{\mathcal{N}_t}^i$ represent the objective posterior distribution given by Eq.(3.21).

$$\{\mathbf{x}_{\mathcal{N}_t}^i, w_t^i\}_{i=1}^N \propto p(\mathbf{x}_{\mathcal{N}_t} | \mathbf{y}_{0:t}) \quad (3.21)$$

Algorithm-2 : The regularized VRPF algorithm

1. Apply step 1, 2 and 3 in algorithm-1, respectively. Replace step 4 with the regularization step given below.
2. Regularization step
 - if $N_{eff} < thr$
 - Calculate the empirical covariance matrix, \mathbf{S}_t , and decompose, \mathbf{D}_t where $\mathbf{S}_t = \mathbf{D}_t \mathbf{D}_t^T$.
 - resample the particles, $\{\mathbf{r}_t^i, w_t^i\}_{i=1}^{N_p}$
 - for $i = 1 \dots N_p$
 - * draw a sample from the Gaussian kernel, $\epsilon^i \sim K_b(\mathbf{r})$
 - * $\mathbf{r}_t^{*i} = \mathbf{r}_t^i + b\mathbf{D}_t \epsilon^i$
 - Iterate through step 2 of the Algorithm-1

3.4.2 Regularized multiple model variable rate particle filter

In this section we describe the regularization scheme for multiple model variable rate models that is a straightforward extension of the R-VRPF described in Section 3.4.1.

Let us define the time indexed multiple model variable rate state vector as,

$$\mathbf{h}_t = [T_{T, \tau_{N_t^-}} \quad T_{P, \tau_{N_t^-}} \quad v(t) \quad \psi(t) \quad \mathbf{z}(t) \quad \tau_{N_t^+} \quad m_{N_t^-}]. \quad (3.22)$$

For simplicity, a more compact representation can be written as,

$$\mathbf{h}_t = [\mathbf{r}_t \quad m_{N_t^-}] \quad (3.23)$$

by using the definition in Eq.(3.13) where $\mathbf{r}_t = [T_{T, \tau_{N_t^-}} \quad T_{P, \tau_{N_t^-}} \quad v(t) \quad \psi(t) \quad \mathbf{z}(t) \quad \tau_{N_t^+}]$. According to the Bayesian rule, the time indexed posterior distribution can be expressed as in Eq.(3.24),

$$p(\mathbf{h}_t | \mathbf{y}_{0:t}) = p(\mathbf{r}_t | m_{N_t^-}, \mathbf{y}_{0:t}) p(m_{N_t^-} | \mathbf{y}_{0:t}) \quad (3.24)$$

that is often a multimodal mixture density where mixture probabilities are defined with $p(m_{N_t^-} | \mathbf{y}_{0:t})$ and the mode conditioned posterior is expressed by $p(\mathbf{r}_t | m_{N_t^-}, \mathbf{y}_{0:t})$. Note that Kernel bandwidth calculation often assumes unimodal posterior densities

such as Gaussian pdf that can cause poor bandwidth selection for the mixture posterior $p(\mathbf{h}_t|\mathbf{y}_{0:t})$. Therefore, in contrast to R-VRPF that defines the regularized posterior as $p(\mathbf{r}_t|\mathbf{y}_{0:t})$, the regularization step in RMM-VRPF is carried out on the mode conditional posterior $p(\mathbf{r}_t|m_{\mathcal{N}_t^-}, \mathbf{y}_{0:t})$. Consequently the regularized distribution conditioned on mode index in continuous form is given by Eq.(3.25).

$$p(\mathbf{r}_t|, m_{\mathcal{N}_t^-}, \mathbf{y}_{0:t}) \approx \sum_{i=1}^{N_p} w_t^i K_b(\theta_t - \theta_t^i). \quad (3.25)$$

The regularization steps explained in Section 3.4.1 are carried out for each mode independently where the mode dependent empirical covariance matrix is \mathbf{S}_k^n , $n \in \{1 \dots r\}$ and the decomposed matrix is represented with \mathbf{D}_k^n .

Algorithm-3 : The regularized MM-VRPF algorithm

1. Apply step 1, 2 and 3 in algorithm-1, respectively. Replace step 4 with the regularization step given below.
2. Regularization step
 - if $N_{eff} < thr$
 - for $n = 1$ to r
 - * calculate the empirical covariance matrix, \mathbf{S}_k^n , and decompose, \mathbf{D}_k^n for each motion mode.
 - resample the particles, $\{\mathbf{r}_t^i, w_t^i\}_{i=1}^{N_p}$
 - for $i = 1 \dots N_p$
 - * draw a sample from the Gaussian kernel, $\epsilon^i \sim K_b(\mathbf{r})$
 - * Assign $n = m_{\mathcal{N}_t^-}^i$
 - * $\mathbf{r}_t^{*i} = \mathbf{r}_t^i + b\mathbf{D}_k^n \epsilon^i$
 - Iterate through step 2 of the Algorithm-1

3.5 Experimental Results

In this section we present experimental results obtained by the proposed MM-VRPF and compared its tracking performance with the conventional VRPF, MM-PF and

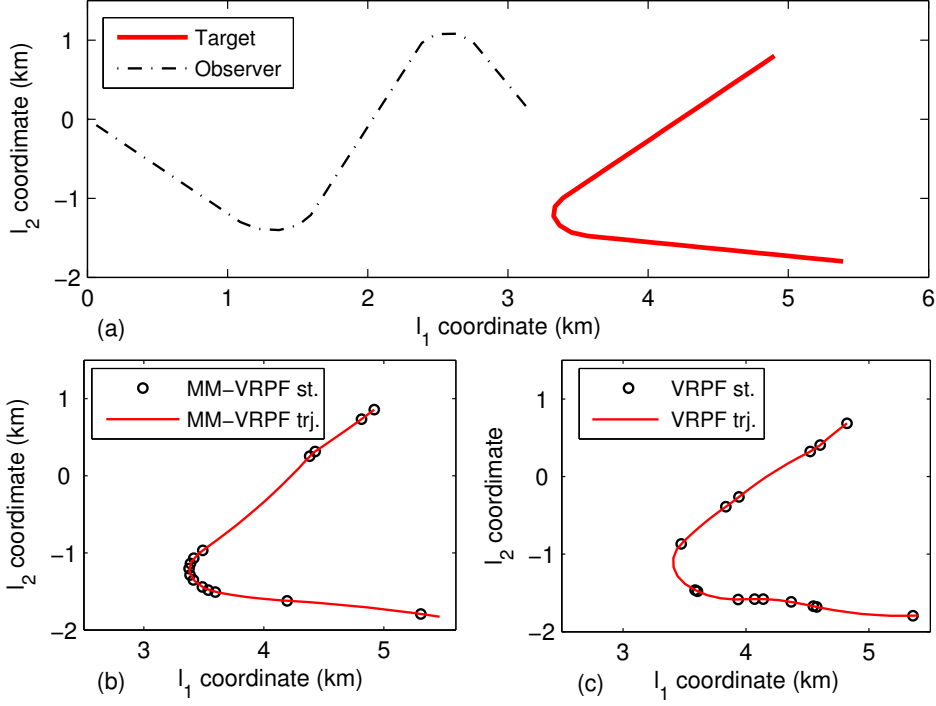


Figure 3.1: (a) Target and observer trajectories for the scenario-1. Trajectories and states of a particle generated by (b) the MM-VRPF, and (c) the VRPF.

IMM-EKF for bearings only maneuvering target tracking. We have also evaluated the improvements achieved in target tracking by integrating a regularization scheme into single model and multiple model variable rate structures named as regularized multiple model VPRF (RMM-VRPF) and regularized VPRF (R-VRPF), respectively. The tracking performances are reported for different scenarios having various turn rates and sensitivity of the performance to state initialization is also investigated.

Let y_t refers to a bearing measurement taken by a passive target tracking sensor at time instant t and is given by Eq.(3.26).

$$y_t = \arctan\left(\frac{l_1 - l_{1o}}{l_2 - l_{2o}}\right) + v_t \quad (3.26)$$

where, $\mathbf{z} = [l_1 \ l_2]^T$ refers to the target position vector and $v_t \sim N(0, \sigma_\theta^2)$ is the independently distributed Gaussian sensor noise. $[l_{1o} \ l_{2o}]^T$ defines the sensor position which is known by the observer.

Two conventionally used performance metrics defining the instant root mean square position error ($RMSE_t$) and time averaged root mean square position error (RMSE) are used in the evaluation of tracking performance (Arulampalam et al., 2004). Reported results are obtained by running $L = 100$ Monte Carlo simulations for each filter. For

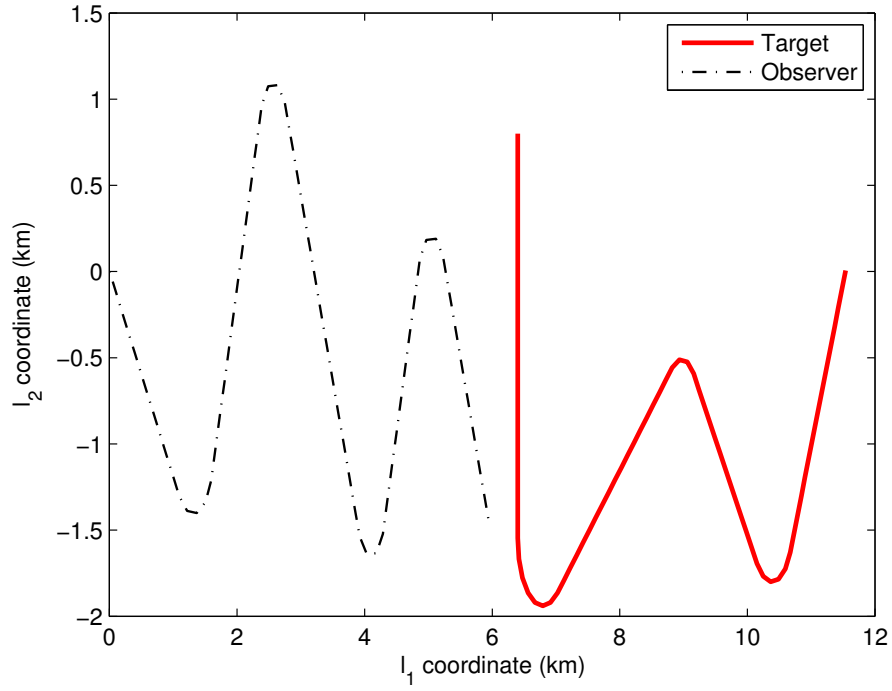


Figure 3.2: Target and observer trajectories for the scenario-2.

i 'th run, let $(\hat{l}_{1t}^i, \hat{l}_{2t}^i)$ and (l_{1t}^i, l_{2t}^i) denote the estimated and true positions obtained at time t , respectively. RMSE_t and RMSE values are computed by Eq.(3.27) and Eq.(3.28), respectively where T is the index to the latest observation.

$$\text{RMSE}_t = \sqrt{\frac{1}{L} \sum_{i=1}^L (\hat{l}_{1t}^i - l_{1t}^i)^2 + (\hat{l}_{2t}^i - l_{2t}^i)^2} \quad (3.27)$$

$$\text{RMSE} = \sqrt{\frac{1}{TL} \sum_{t=1}^T \sum_{i=1}^L (\hat{l}_{1t}^i - l_{1t}^i)^2 + (\hat{l}_{2t}^i - l_{2t}^i)^2} \quad (3.28)$$

We synthesized two bearing-only test scenarios that simulate a maneuvering target for testing. The first one is illustrated in Figure 3.1(a) which is akin to the one synthesized by Godsill and Vermaak (2005) and the second scenario is shown in Figure 3.2. In the first scenario ownship starts moving at the origin of the coordinate plain and travels at a constant speed of 5 knots for 11 minutes with a course of 140° . Thereafter executes a maneuver between time intervals $(12 - 16)$ at a constant turn rate $30^\circ/\text{min}$ attaining a new course of 20° . Ownship maintains this course for 15 minutes and makes a second maneuver at a constant turn rate of $45^\circ/\text{min}$ to attain a course of 155° . It maintains this course till the end of the scenario. Target starts moving at the position $(5400, 500)$ on the coordinate plane and travels at a course of -150° with a constant speed of 4

knots. It executes a maneuver between minutes (20 – 25) with a constant turn rate of $24^\circ/\text{min}$ to attain a new course of 100° , and maintains the same course for the rest of observation periods. An observation period is equal to 1 minute and total number of observations is equal to 40 for the presented scenario.

The second scenario shown in Figure 3.2 considers multiple turns for the target and the observer to evaluate the long term performance of the algorithms. Observer starting at the origin travels with a constant speed of 5 knots and executes four consecutive maneuvers with a rate of $30^\circ/\text{min}$, $-45^\circ/\text{min}$, $45^\circ/\text{min}$, and $-45^\circ/\text{min}$ within the time intervals, (12 – 16), (31 – 34), (52 – 55), and (66 – 69). The observation period is 1 minute where the number of observations is equal to 80 resulting in a 80 minutes long scenario. Target starts moving at the position (6400m, 800m) on the coordinate plane and travels at a course of -180° with a constant speed of 4 knots and it executes three maneuvers between the time intervals (20 – 27), (45 – 48), and (60 – 65) with the rate of $18^\circ/\text{min}$, $-28^\circ/\text{min}$, and $22^\circ/\text{min}$ consecutively. In both scenarios bearing measurements are assumed to be perturbed with a zero mean Gaussian noise having a standard deviation $\sigma_\theta = 1.5^\circ$.

In order to compare behavior of all methods clearly, two different test cases are considered. In the first test, performance of the proposed MM-VRPF is compared to VRPF, MM-PF and IMM-EKF. To evaluate capability of each model in characterization of the straight and maneuvering target motions, initial conditions are set to their true values. Therefore, all filters are initialized by a Gaussian distribution centered on the true mean value and variance, i.e., $\sigma_r = 100\text{m}$ for the range and $\sigma_\theta = 1.5^\circ$ for the bearing. Initial values of heading and velocity are also set to the true values. To observe the effects of particle size on the tracking performance, simulations has been carried out for $N_p = 2000$ and $N_p = 8000$ particles for both scenarios.

Aim of the second test on bearing only tracking was evaluation of improvement gained by incorporating a regularization scheme into VRPF and MM-VRPF models, and evaluation of the tracking performance under erroneous initial conditions. We have also demonstrated the performance of regularized MM-PF for comparison purposes. As we have stated before, regularization often improves the performance in case of degeneracy which is observed when the parameters are not appropriately specified or the algorithm is badly initiated. Thus we examined the effect of regularization for

Table 3.1: MM-VRPF and VRPF parameters for the bearing-only scenarios.

	MM-VRPF		VRPF
	mode-1	mode-2,3	
$\mu_{T,n}, \sigma_{T,n}$	(0, 100)	(0, 100)	(0, 100)
$\mu_{P,n}, \sigma_{P,n}$	(0, 500)	($\pm 11000, 3000$)	(0, 5000)
α_n, β_n	(1.5, 4)	(0.5, 0.35)	(0.5, 6.5)
τ_n	(0)	(0.5)	0

erroneous initial data where variations from the true range, course and velocity are assumed to be Gaussian distributed with standard deviation of 1500m, 10° , and 0.3 m/sn, respectively. Simulations has been carried out for $N_p = 2000$ and $N_p = 8000$ particles.

Model Parameters for the variable rate models MM-VRPF and VRPF used in tests are listed in Table 3.1. MM-PF and IMM-EKF use nearly constant turn (NCT) model with 3 possible turn rates where parameters are chosen for each mode as $w_1 = 0$, $w_2 = 0.5$, $w_3 = -0.5\text{rad/sn}$ ($w \in \Omega$) and state noise covariance matrix is set to $2 \times 10^{-6}\mathbf{I}_2$. Parameter sets are selected heuristically to make sure that different motion characteristics can be represented by the same parameter set.

The proposed multiple model variable rate structure defines three modes where the first mode ($n = 1$) models the straight target trajectories and the other modes, ($n = \{2, 3\}$) are parameterized in order to track the maneuvers performed to each direction. In our model we have chosen a fixed mode transition probability matrix \mathbf{P} as given in Eq.(3.29).

$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 0.35 & 0.45 & 0.2 \\ 0.35 & 0.2 & 0.45 \end{bmatrix} \quad (3.29)$$

Note that Eq.(3.29) is different than the one shown in Eq.(3.30) which is utilized by fixed rate models MM-PF and IMM-EKF (Arulampalam et al., 2004).

$$\mathbf{P} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.4 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \quad (3.30)$$

We always choose elements of the the matrix \mathbf{P} for variable rate models considering the object motion characteristics such that the model can track maneuvers at different rates. Roughly estimated values for the transition matrix can be calculated as explained in Section 3.2. Eq.(3.8) indicates that increasing the value at the diagonal elements of matrix \mathbf{P} also increases expected time spent at a particular motion mode.

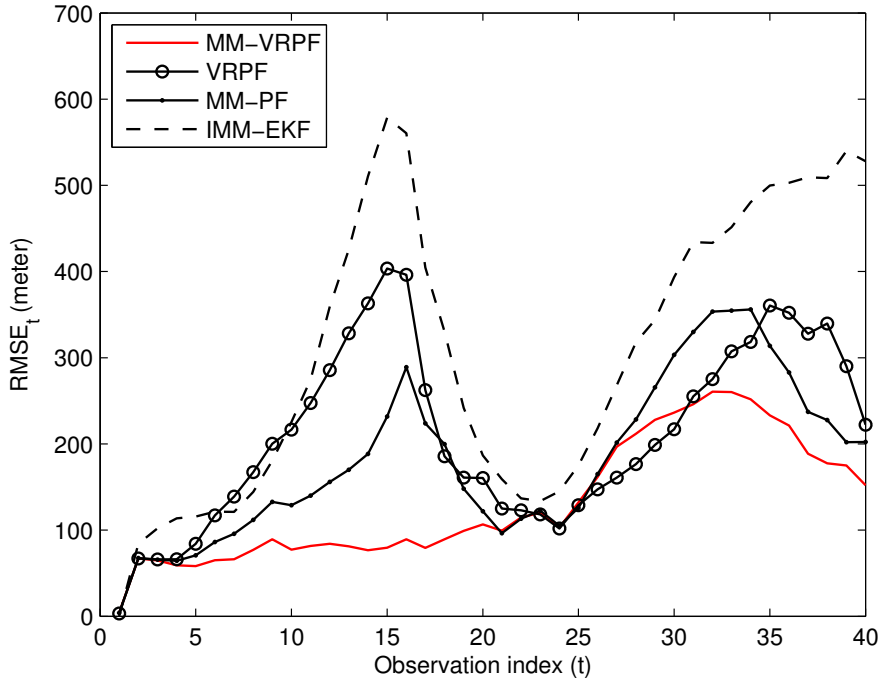


Figure 3.3: $RMSE_t$ versus time t for true initials ($N_p = 2000$).

In Eq.(3.29) and Eq.(3.30) probabilities of the transitions from maneuvering models are identical, however in contrast to the fixed rate models, straight motion model indexed as $n = 1$ allows switching to either of the adaptive maneuvering models indexed as $n = \{2, 3\}$ with higher probabilities.

Figure 3.1(b) and (c) illustrate the trajectory and state arrival points of a single particle generated by the VRPF and MM-VRPF, respectively for the Scenario-1. As it is shown in Figure 3.1(b), the proposed MM-VRPF is capable of locating frequent states at high maneuvering regions while using a parsimonious state representation for the smooth regions of the trajectory owing to the sojourn time distribution parameters of the adaptive models. Note that the continuous representation of the motion dynamics of a maneuver enables MM-VRPF to locate frequent state points even higher than the observation period. This yields better characterization of the maneuver parameters and arrival times independent from the observation time. It can be concluded that the proposed MM-VRPF, using a more flexible rate estimation procedure, is capable of estimating the target trajectory precisely.

Figure 3.3 and Figure 3.4 plot the $RMSE_t$ versus observation time index t for four different models: the VRPF, MM-VRPF, MM-PF and IMM-EKF simulated using

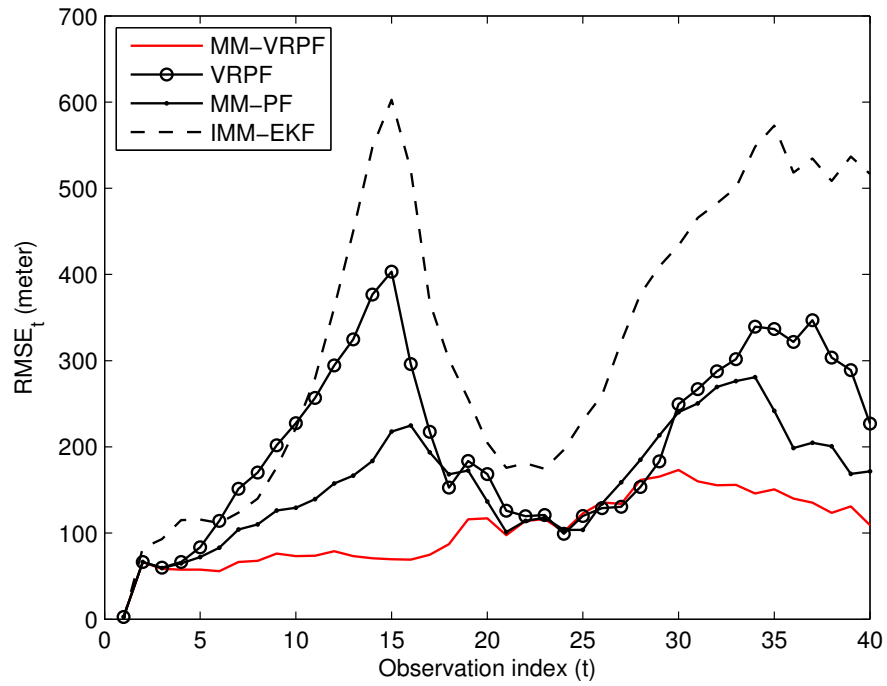


Figure 3.4: RMSE_t versus time t for true initials ($N_p = 8000$).

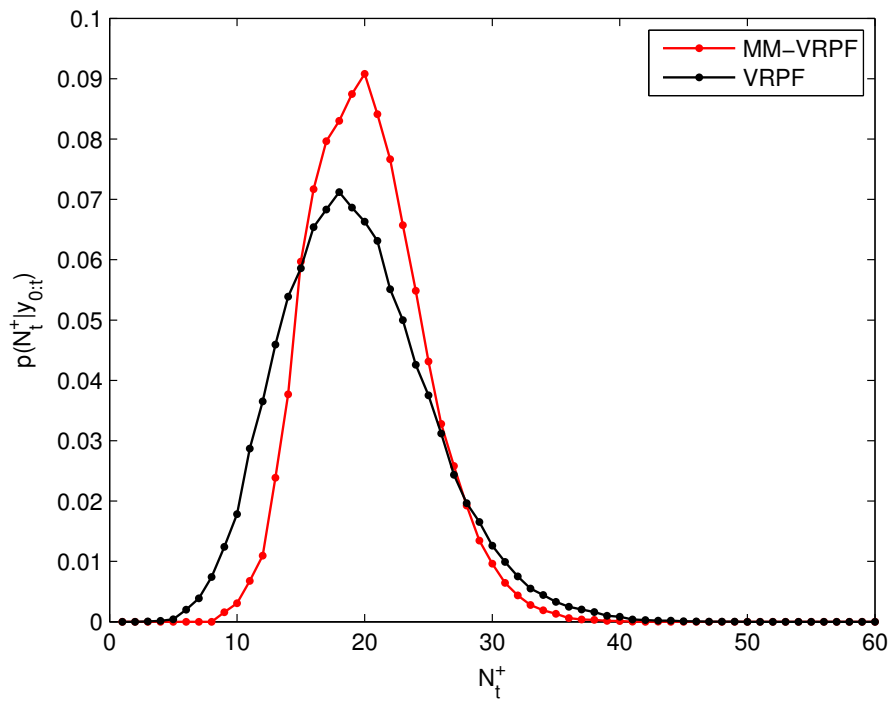


Figure 3.5: Posterior distribution of number of states $p(\mathcal{N}_t^+ | \mathbf{y}_{0:t})$ where $t = 40$.

$N_p = 2000$ and $N_p = 8000$ particles, respectively. Results are attained by using 40 observations obtained for the scenario-1 described above. As it is shown in the figures, the proposed MM-VRPF is able to track the target before and after the maneuver with lower RMSE_t values compared to the other filters for varying size of particles $N_p = 2000$ and $N_p = 8000$ while all models are superior to IMM-EKF utilizing NCT motion model. Experimental results show the capability of MM-VRPF in characterizing straight as well as maneuvering trajectories with the same set of parameters.

Fixed rate models such as MM-PF and IMM-EKF associates each state vector with an observation received at a particular time t . Hence, the total number of states utilized along the scenario presented in Figure 3.1 is fixed and equal to 40. However, the number of states that represent the same target trajectory is a random variable for the variable rate models. The posterior distribution of the number of states, $p(\mathcal{N}_t^+ | \mathbf{y}_{0:t})$, at time $t = 40$ is illustrated in Figure 3.5. It is clear that the distribution of the number of states for MM-VRPF and VRPF are comparable. This result also shows that MM-VRPF can reach to a better posterior estimate even by using less than a half of the number of states used by the fixed rate models.

In order to observe the long term behavior of the algorithms, we evaluated the tracking performance in the second scenario. Given the true initial conditions, RMSE_t error of each model versus observation index is plotted in Figure 3.6 and Figure 3.7 for varying particle sizes of $N_p = 2000$ and $N_p = 8000$, respectively. Due to low performance results obtained by IMM-EKF, we just reported RMSE values for the scenario-2. It is clear that MM-VRPF outperforms VRPF and MM-PF throughout the scenario for both particle size setting. This shows that compared to VRPF and MM-PF, MM-VRPF can better track a target maneuvering at different turn rates. Similarly, the RMSE values reported in Table 3.3 show that MM-VRPF achieves lower RMSE values when compared to other filters for each particle size.

Table 3.2: RMSE for varying particle size obtained by using true initials for the Scenario-1.

	MM-VRPF	VRPF	MM-PF	IMM-EKF
$N_p = 2000$	148.9	236.56	203.5	367.44
$N_p = 8000$	113.82	234.34	180.29	367.44

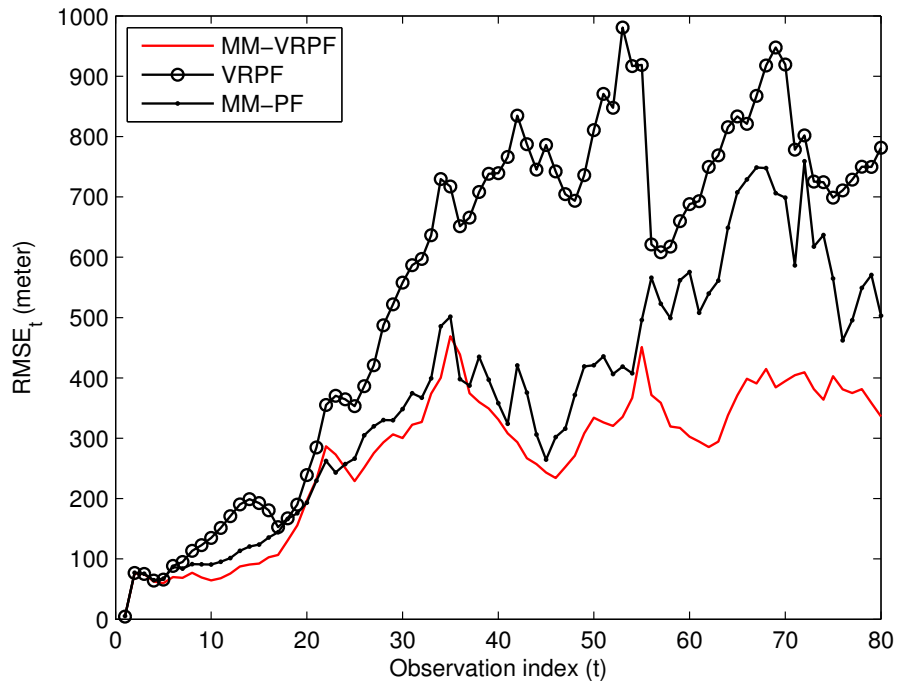


Figure 3.6: RMSE_t versus time t for true initials ($N_p = 2000$).

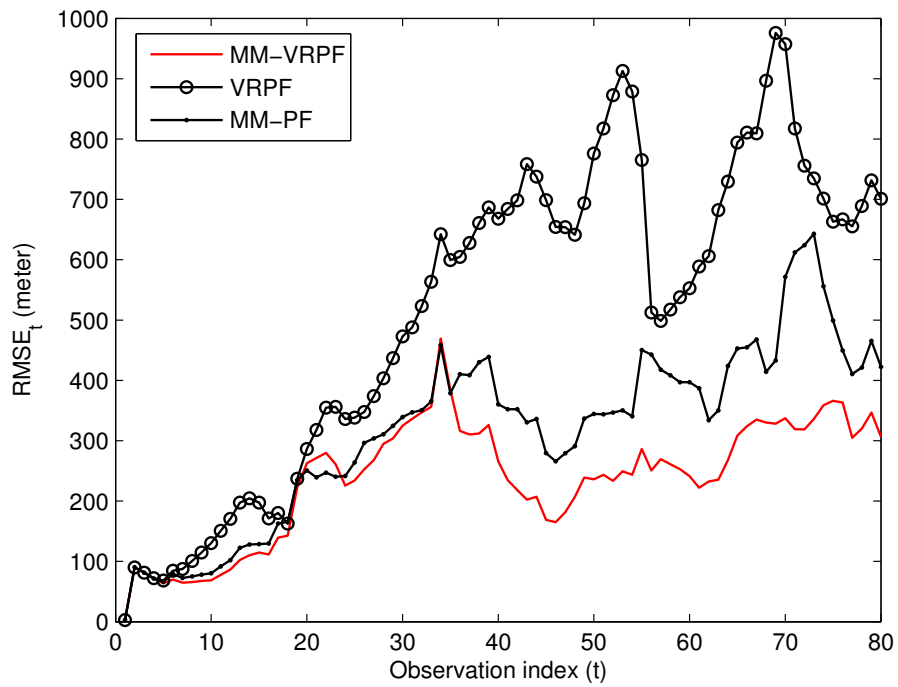


Figure 3.7: RMSE_t versus time t for true initials ($N_p = 8000$).

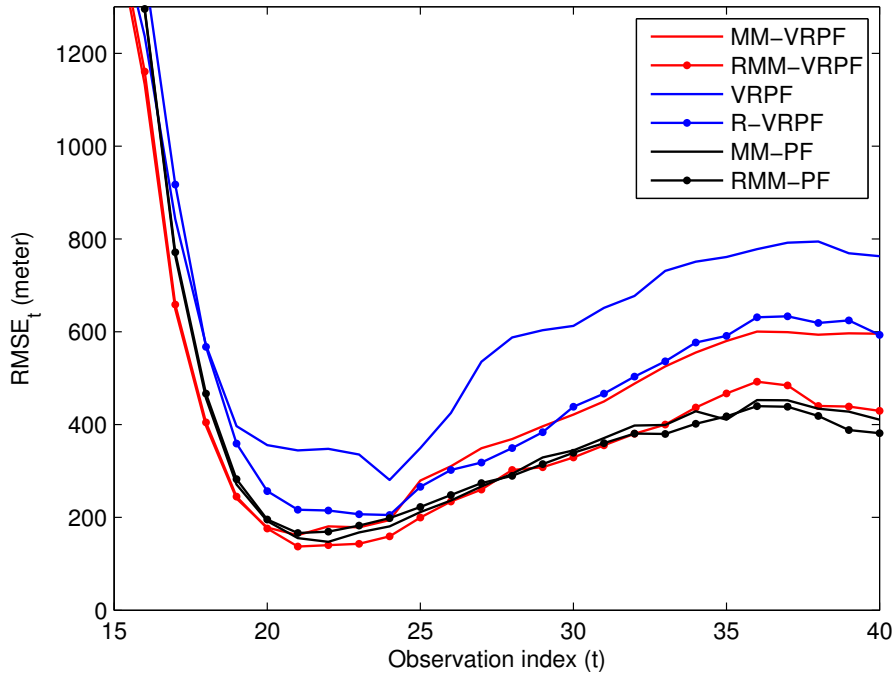


Figure 3.8: $RMSE_t$ versus time t for erroneous initial conditions ($N_p = 2000$).

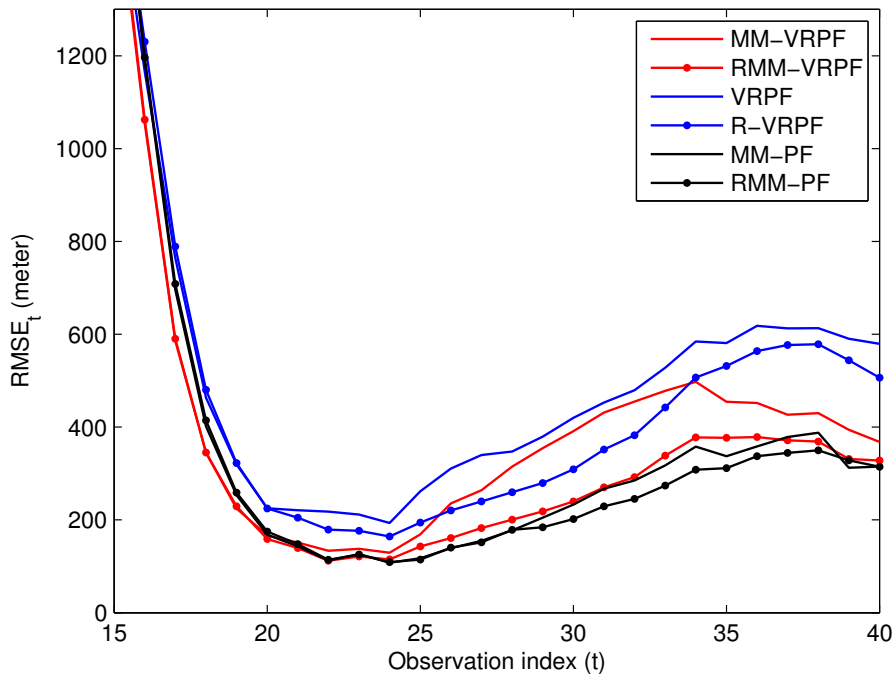


Figure 3.9: $RMSE_t$ versus time t for erroneous initial conditions ($N_p = 8000$).

We also compared the regularized algorithms RMM-VRPF, R-VRPF and RMM-PF with MM-VRPF, VRPF and MM-PF. The results are reported in means of $RMSE_t$ and RMSE values for both of the scenarios for varying particle sizes of $N_p = 2000$ and $N_p = 8000$. We evaluated this test under erroneous initial conditions to show the effects of the initial value selection to the tracking performance in a bearing only scenario. Erroneous initial conditions tend the algorithms to suffer from the degeneracy and regularization process often improves the performance when degeneracy is the case.

$RMSE_t$ error values achieved for erroneous initials for the scenario-1 are plotted in Figure 3.8 and Figure 3.9 for particle size of $N_p = 2000$ and $N_p = 8000$, respectively. Table 3.4 also reports overall RMSE error. Note that in the case erroneous initials are defined, we calculated the RMSE values starting at time $t = 17$ to ensure that the observability condition is valid for the presented scenarios (Song, 1996). For $N_p = 2000$ and $N_p = 8000$, RMM-VRPF and R-VRPF take the advantage of regularization and outperforms their non-regularized counterparts however no improvement could be achieved by the RMM-PF. We concluded that regularization can significantly improve the filtering performance of variable rate models for the Scenario-1 when initials are selected erroneously. In this test, MM-VRPF can not outperform MM-PF, the RMM-VRPF shows comparable performance to MM-PF and its regularized counterpart. We believe that MM-PF takes advantage of using noisy NCT dynamic model particularly when erroneous initials are defined whereas VRPF utilizing a parametric curvilinear motion relatively lacks of modeling large disturbances.

In Figure 3.10 and Figure 3.11 the $RMSE_t$ error values versus observation index are plotted for the Scenario-2 at $N_p = 2000$ and $N_p = 8000$, respectively and the RMSE values are reported in Table 3.5 using erroneous initial conditions. For $N_p = 2000$ regularization improves the performance of both of the variable rate models slightly

Table 3.3: RMSE for varying particle size obtained by using true initials for the Scenario-2.

	MM-VRPF	VRPF	MM-PF
$N_p = 2000$	297.36	625.74	422.85
$N_p = 8000$	274.74	588.60	388.93

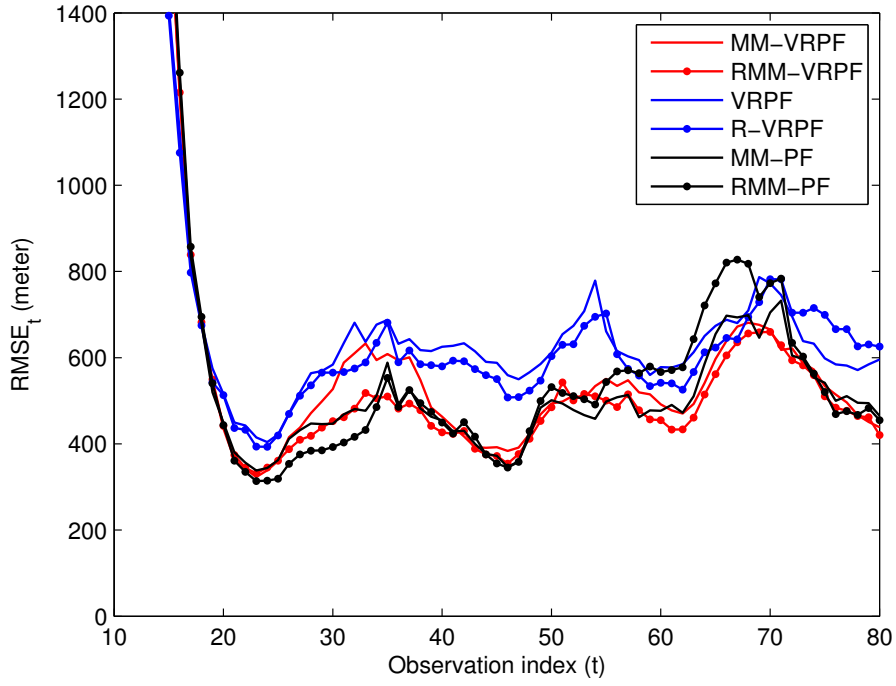


Figure 3.10: $RMSE_t$ versus time t for erroneous initial conditions ($N_p = 2000$).

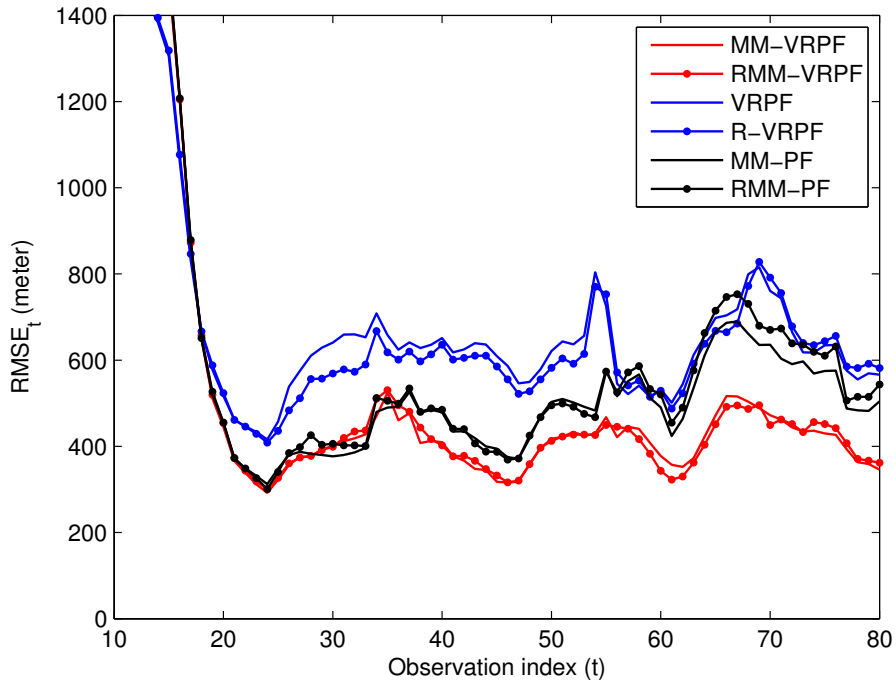


Figure 3.11: $RMSE_t$ versus time t for erroneous initial conditions ($N_p = 8000$).

Table 3.4: RMSE for varying particle size obtained by using erroneous initials for the Scenario-1.

	MM-VRPF	RMM-VRPF	VRPF	R-VRPF	MM-PF	RMM-PF
$N_p = 2000$	442.89	360.42	613.20	484.22	368.70	361.68
$N_p = 8000$	368.90	297.27	459.27	411.65	295.65	284.07

Table 3.5: RMSE for varying particle size obtained by using erroneous initials for the Scenario-2.

	MM-VRPF	RMM-VRPF	VRPF	R-VRPF	MM-PF	RMM-PF
$N_p = 2000$	523.62	493.53	618.15	602.46	526.40	549.80
$N_p = 8000$	427.52	429.15	622.60	606.34	505.02	525.16

but no improvement is achieved when the particle size is $N_p = 8000$. However performance improvement for MM-PF could not be obtained by regularization in either particle size. When we examine the overall performance at particle size $N_p = 2000$ we see the RMM-VRPF and MM-VRPF are superior to VRPF and R-VRPF whereas their performance is comparable to MM-PF and RMM-PF. Advantage of MM-VRPF outcomes when we use $N_p = 8000$ particles as seen in Figure 3.11 and Table 3.5. In Figure 3.11 MM-VRPF and RMM-VRPF outperforms VRPF, MM-PF and their regularized counterparts particularly after the execution of first target maneuver within the time interval (20-27). It can be concluded that MM-PF performs even better than MM-VRPF until the execution of first maneuver of the target, however in the remaining part of the scenario MM-VRPF is the best performing filter particularly when the particle size is $N_p = 8000$.

As a summary, we concluded that MM-VRPF and its regularized counterpart achieve lower estimation error compared to the VRPF and IMM-EKF algorithms in various test conditions. Moreover, MM-VRPF outperforms the MM-PF if the algorithm is not poorly initiated. However, under poor initial conditions, MM-VRPF can achieve comparable performance to the MM-PF only if the particle size is increased above $N_p = 8000$. We also showed that the regularization improves tracking performance of variable rate models by diversifying the particles particularly when the non regularized filters suffer from degeneracy.

4. SEQUENTIAL MONTE CARLO SAMPLERS FOR DIRICHLET PROCESS MIXTURE MODELS

Exact inference for the DPM model posterior is unfortunately intractable. Therefore approximate methods are highly desirable for solution of the high dimensional DPM posterior distribution. The best well known approximate inference techniques proposed in literature rely on variational (Blei and Jordan, 2004, 2006) and Monte Carlo Markov Chain (MCMC) based methods (Escobar and West, 1992; S. Walker and Smith, 1999; MacEachern et al., 1999; Neal, 2000). Though, majority of these methods perform batch algorithms that apply the inference on the entire dataset (Blei and Jordan, 2004; Neal, 2000), sequential methods that cluster each new observation upon its arrival have also been proposed (MacEachern et al., 1999; Quintana, 1996; Fearnhead, 2004).

Intuitively, a DPM model is a probability density over disjoint partitions of the observations. Once a partition is chosen, the parameters of each cluster can be estimated often very easily. Even better, if the prior density of the parameters is selected to be conjugate to the observation model, one can integrate out the parameters analytically and represent the target posterior as a discrete distribution on a collection of cluster indicators. In this case the Bayesian inference problem based on Monte Carlo methods deals with sampling from a high dimensional discrete distribution defined on all possible clusterings of the data.

By construction, the DPM model is exchangeable and the ordering of data does not matter but for inference it is nevertheless beneficial to process data sequentially in some natural order. Such an approach gives computational advantages especially for large datasets. In the literature a number of online inference techniques have been proposed to estimate an artificially time evolving DPM posterior (Quintana, 1996; MacEachern et al., 1999; Fearnhead, 2004).

However it is also shown that sequential filtering is not an appropriate method for especially large data sets due to the accumulated Monte Carlo error over time

(Quintana and Newton, 1998). This problem arises from the sampling procedure of the particle filtering algorithm in which the discarded particles are never reconsidered thus previous label assignments can never be updated according to the information provided by the latest observations.

In order to overcome this drawback, we propose an efficient sequential Monte Carlo sampler that estimates the sequentially evolving DPM model posterior. Unlike the existing methods proposed by Quintana (1996); Fearnhead (2004) our algorithm enables us to update past trajectories of the particles in the light of recent observations. Our method takes advantage of the SMC sampler framework to design such update schemes (Del Moral et al., 2006). The proposed algorithm is evaluated on a single dimensional Gaussian mixture density estimation problems and performance improvement over conventional models are shown.

4.1 Revisiting the DPM Model and SCM samplers

Let us denote the observation sequence at time n by $y_n = \{y_{n,1} \dots y_{n,n}\}$. Each observation $y_{n,i}$, $i = 1, \dots, n$, is assigned to a cluster where $z_{n,i} \in \{1, \dots, k_n\}$ is the cluster label and, $k_n \in \{1 \dots n\}$ represent the number of clusters at time n . The vector of cluster variables is defined as $z_n = \{z_{n,1} \dots z_{n,n}\}$ and corresponding cluster parameters are represented with the parameter vector $\theta_n = \{\theta_{n,1} \dots \theta_{n,k_n}\}$.

The DPM model assumes that the cluster parameters are independently drawn from the prior $\pi(\theta)$ and the observations are independent of each other conditional on the assignment variable $z_{n,i}$. Hence the DPM posterior density $\pi_n(x_n)$ can be expressed as,

$$\pi_n(x_n) \propto p(z_n) \prod_{j=1}^{k_n} p(\theta_{n,j}) \prod_{i=1}^n g(y_{n,i} | \theta_{n,z_{n,i}}) \quad (4.1)$$

where $x_n = \{z_n, \theta_n\}$. The prior on clustering variable vector z_n is formulated by Eq.(4.2) in a recursive way,

$$p(z_{n,i+1} = j | z_{n,\{1:i\}}) = \begin{cases} \frac{l_j}{i+\kappa}, & j = 1, \dots, k_i \\ \frac{\kappa}{i+\kappa}, & j = k_i + 1 \end{cases} \quad (4.2)$$

where k_i is the number of clusters in the assignment $z_{n,\{1:i\}}$. l_j is the number of observations that $z_{n,\{1:i\}}$ assigns to cluster j and κ is a 'novelty' parameter.

In our work, we assume that a conjugate prior is chosen such that given z_n , the parameter θ_n can be integrated out and the DPM posterior distribution can be calculated up to a normalizing constant.

4.1.1 A generic SMC sampler

The SMC sampler samples from a sequence of target densities evolving with a countable index n , $\pi_1(x_1) \dots \pi_n(x_n)$, each defined on a common measurable space (E_n, \mathcal{E}_n) where $x_n \in E_n$. The generic SMC algorithm sampling from the sequentially evolving target posterior $\tilde{\pi}_n$ is presented as follows:

Assume that a set of weighted particles $\{W_{n-1}^i, X_{1:n-1}^i\}_{i=1}^{N_p}$ approximate $\tilde{\pi}_{n-1}$ at time $n-1$. At time n the path of each particle can be extended using a Markov kernel, $K_n(x_{n-1}, x_n)$. The unnormalized importance weights associated with the extended particles are calculated according to Eq.(4.3),

$$\begin{aligned} w_n(x_{1:n}) &= w_{n-1}(x_{1:n-1})v_n(x_{n-1}, x_n) \\ &= \frac{\tilde{\gamma}_n(x_{1:n})}{\eta_n(x_{1:n})} \end{aligned} \quad (4.3)$$

where the incremental term of weight equation, $v_n(x_{n-1}, x_n)$, is equal to

$$v_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n)L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}. \quad (4.4)$$

The discrepancy between η_n and $\tilde{\gamma}_n$ tends to grow with n , consequently the variance of the unnormalized importance weights increases. A resampling step is used if the variance is above a certain level as measured by, e.g, effective sample size (ESS).

Design of efficient sampling schemata hinges on properly choosing L_n with respect to K_n . The introduction of the L_n extends the integration domain from E to E^n and eliminates the necessity of calculating $\eta_n(x_n)$. However increasing the integration domain also increases the variance of the importance weights. Del Moral et al. (2006) showed that the optimal backward Markov kernel L_{k-1}^{opt} ($k = 2, \dots, n$) minimizing the variance of the unnormalized importance weight $w_n(x_{1:n})$ is given for any k by,

$$L_{k-1}^{opt}(x_k, x_{k-1}) = \frac{\eta_{k-1}(x_{k-1})K_k(x_{k-1}, x_k)}{\eta_k(x_k)} \quad (4.5)$$

However, the kernel given by Eq.(4.5) usually does not admit a closed form solution. Therefore common strategy is to approximate the optimal kernel as close as possible

to provide asymptotically consistent estimates (Del Moral et al., 2006) . A sensible approximation at a given time step n can be obtained by substituting π_{n-1} for η_{n-1} , where the approximate kernel L_{n-1} can be expressed as in Eq.(4.6),

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}{\int \pi_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)dx_{n-1}} \quad (4.6)$$

that can yield a closed form solution to the weight update equation if it is possible to calculate the integration. An alternative to approximate backward kernel can be obtained as in Eq.(4.7) by replacing $\pi_{n-1}(x_{n-1})$ by $\pi_n(x_{n-1})$ and selecting K_n as a MCMC kernel targeting π_n in Eq.(4.6).

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1})K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \quad (4.7)$$

Note that, although Eq.(4.6) is a closer approximation to the optimal backward kernel, Eq.(4.7) can lead to simpler weight update equations.

4.2 A SMC sampler for the Dirichlet Process Mixtures

In this section we will explain the proposed SMC based algorithm that generates weighted samples from the DPM model posterior described in Section 4.1.

Now, assuming conjugacy, we devise an algorithm that approximates the posterior distribution,

$$P(z_n|y_n) \approx \sum_{i=1}^{N_p} W_n^i \delta_{Z_n^i}(z_n) \quad (4.8)$$

with a set of weighted samples $\{W_n^i, Z_n^i\}_{i=1}^{N_p}$ where each particle Z_n^i encodes an assignment vector of all datapoints upto time n , formally represented with a Dirac delta function $\delta_{Z_n^i}(z_n)$.

Let us define a forward kernel, K_n , generating samples from the sequence of distributions built according to Eq.(4.1). We first partition an assignment vector $z_n = \{z_{n,r}, z_{n,d}, z_{n,n}\}$ where r is a subset of $\{1, \dots, n-1\}$, a set of not necessarily consecutive indices, and $d = \{1, \dots, n-1\} - r$. Note that throughout the text we will call the set $z_{n,r}$ as the active block. We define $u = r \cup \{n\}$, and denote $-u \equiv d$. Exploiting the conjugacy property, we propose using the following conditional distribution for K_n as given in Eq.(4.9).

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1}, -u}(z_{n,-u}) \pi_n(z_{n,u}|z_{n,-u}) \quad (4.9)$$

This kernel allows us updating $z_{n,u}$ which includes the current and some past assignments without changing the rest $z_{n,-u}$.

By replacing K_n in Eq.(4.6) we obtain the straight derivation to the approximate kernel,

$$L_{n-1}(z_n, z_{n-1}) = \delta_{z_{n,-u}}(z_{n-1,-u}) \quad (4.10)$$

$$\times \pi_{n-1}(z_{n-1,r} | z_{n-1,-r}).$$

Given our choices of the forward and backward kernels, now we are able to write the expression for the incremental weight function given in Eq.(4.4) as follows,

$$v_n(z_{n-1}, z_n) = \frac{\gamma_n(z_{n,-u})}{\gamma_{n-1}(z_{n-1,-r})}. \quad (4.11)$$

The proposed scheme can also be seen as a generalization of a conventional particle filtering weight update scheme. The particle filter simply uses the forward kernel $K_n(z_{n-1}, z_n) = \delta_{z_{n,-n}}(z_{n,-n})\pi_n(z_{n,n} | z_{n,-n})$. In this case only the clustering variable $z_{n,n}$ is updated upon arrival of the new observation that yields the weighting function given in Eq.(4.12).

$$v_n^{\text{pf}}(z_{n-1}, z_n) = \frac{\gamma_n(z_{n,-n})}{\gamma_{n-1}(z_{n-1})} \quad (4.12)$$

The sequential imputation scheme given by Liu (1996) and many particle filtering based methods proposed by Quintana and Newton (1998); Chen and Liu (2000) use the simplified incremental weight update function given by Eq.(4.12). Note that such a kernel selection strategy is not capable of updating the active set $z_{n,r}$ according to the new observations, therefore can yield to poor estimation performance.

In order to render our sampling approach more efficient by making more global moves we wish to change a block of variables, i.e., choose the cardinality of the index set r as large as possible. However, when the cardinality of r increases, the time required for the exact computation of the incremental weight in Eq.(4.11) grows exponentially. In the sequel, we will define MCMC and approximate Gibbs type moves where the associated weight update equations can be computed efficiently. This leads to low complexity algorithms for sampling from the time evolving DPM posterior distribution.

4.2.1 MCMC kernels

We first define the forward kernel as

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1}, -u}(z_{n,-u}) \times K_n(z_{n,n}, z_{n,r} | z_{n-1}) \quad (4.13)$$

where $K_n(z_{n,n}, z_{n,r} | z_{n-1})$ is a valid MCMC kernel applying a single Gibbs step targeting the full conditional distribution $\pi_n(z_{n,n}, z_{n,r} | z_{n,-u})$. Intuitively, this kernel updates the active block using a Gibbs sampler and constructs the proposal distribution using the sequence of full conditional distributions.

A corresponding backward kernel can be obtained by substituting $K_n(z_{n-1}, z_n)$ into the Eq.(4.7). This yields in the following incremental weight update equation,

$$v_n^{gb}(z_{n-1}, z_n) = \frac{\gamma_n(z_{n-1,r}, z_{n,-u})}{\gamma_{n-1}(z_{n-1})}. \quad (4.14)$$

Note that as a consequence of the chosen backward kernel, Eq.(4.14) is independent from the initialization of the Gibbs moves. If the active block set is selected as $r = \{1 \dots n-1\}$, the update equation in Eq.(4.13), will be equal to the one introduced by MacEachern et al. (1999) as S4 algorithm.

The above schema depends exclusively on local Gibbs moves. As is the case in the application of the Gibbs sampler, we may expect to get stuck in local modes due to slow mixing especially when the posterior distribution is multi modal. In such situations, annealing is a general strategy to pass through low probability barriers. However, as one modifies the target density gradually, finding the correct schedule is crucial. On the other hand, in the SMC framework we don't have to choose a schedule explicitly. We are free to choose any forward kernel, provided we compute the corresponding incremental weight. Here, we propose a forward kernel which targets the modified full conditional distribution, $\pi_n(z_{n,n}, z_{n,r} | z_{n,-u}, \rho_n)$. Note that bridging is achieved simply by changing the novelty parameter of the underlying Dirichlet process to ρ_n . The SMC theory guarantees that we still target the original target density.

A valid backward kernel can be obtained by replacing π_n with the modified version of the target distribution $\pi_n(\cdot | \rho_n)$ in Eq.(4.7) and the resulting weight update equation

can be represented as follows,

$$v^{an}(z_{n-1}, z_n) = \frac{\gamma_n(z_n)}{\gamma_{n-1}(z_{n-1})} \times \frac{\pi_n(z_{n-1,r} | z_{n-1,-r}, \rho_n)}{\pi_n(z_{n,u} | z_{n,-u}, \rho_n)}. \quad (4.15)$$

While we are able to escape low probability barriers, the modified full conditional distribution introduces a further approximation. Thus, we advise still choosing the ρ_n converging to the true κ with the increasing time index n . Note that this is merely a choice, not a requirement in contrast to a tempered Gibbs sampler, where the final density must coincide with the true target.

4.2.2 Sequential approximation

As we rely on a blocked Gibbs sampler, we are constrained by low dimensional blocks. The key idea in this method is to approximate sequentially to the exact full conditional distributions given by Eq.(4.9) and Eq.(4.10). As in the previous section, we are free to choose any approximation to the full conditionals as these are merely used as our proposal density. Asymptotically, the SMC sampler ensures convergence to true target posterior even approximations to these full conditional distributions are defined. Note that the approximations, should be selected as close as possible to the full conditionals to obtain an efficient sampler.

We assume that there are Q clustering variables in the active block and we further enumerate them

$$z_{n,r} = \{z_{n,r_1} \cdots z_{n,r_Q}\} \quad (4.16)$$

where r_q denotes the q 'th index of the block at time n with $q = 1 \dots Q$. In the sequel, we will design an approximation that enables us to design kernels where the computational load increases linearly with Q . Hence, we can chose an active block with size Q quite large in practice.

We propose the following approximation to the forward kernel K_n ,

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1,-u}}(z_{n,-u}) \hat{\pi}_n(z_{n,u} | z_{n,-u}) \quad (4.17)$$

where

$$\begin{aligned} \hat{\pi}_n(z_{n,u}|z_{n,-u}) &= \pi_n(z_{n,n}|z_{n,r}, z_{n,-u}, \rho_n) \\ &\times \prod_{i=1}^Q \pi_{n-1,-r_{\{i+1:Q\}}}(z_{n,r_i}|z_{n,-u}, z_{n,r_{\{1:i-1\}}}, \rho_n). \end{aligned} \quad (4.18)$$

We assume $r_{\{i:j\}}$ is empty for $i > j$. The rationale beyond this choice is as follows: we drop all the observations corresponding to the active block, including the last observation and incorporate them one by one in a new (possibly random) order. Recall that in Eq.(4.18), $\pi_{n-1,-r_{\{i+1:Q\}}}(z|z', \rho_n) = p(z|z', \{y_{n-1,1:n-1}\} - \{y_{n-1,r_{i+1}} \cdots y_{n-1,r_Q}\})$ denotes the modified full conditional distribution given the all observations excluding the ones indexed by $r_{\{i+1:Q\}}$. Note that approximations of this form are quite common in approximate inference for state space models, where q corresponds to a time index; we merely omit the effect of the 'future' observations.

The formulation given by Eq.(4.18) enables us to recursively calculate and sample the overall kernel density function $\hat{\pi}_n(z_{n,u}|z_{n,-u})$ efficiently with a reasonable complexity even for large values of Q . Note that the scheme introduced in Eq.(4.18) processes the observations sequentially in the indexed order $\{r_1 \dots r_q\}$ and finally extends the space using the proposal function $\pi_n(z_{n,n}|z_{n,r}, z_{n,-u}, \rho_n)$. Clearly, due to exchangeability of the DPM model, there is no need to process the observations in a fixed order. To diminish the effects of the particular processing order, it is preferred to apply a random permutation of the indices in r at each step of the algorithm.

A similar sequential procedure is also required to approximate the backward kernel given by

$$\begin{aligned} L_{n-1}(z_n, z_{n-1}) &= \delta_{z_{n,-u}}(z_{n-1,-u}) \\ &\times \hat{\pi}_{n-1}(z_{n-1,r}|z_{n-1,-r}) \end{aligned} \quad (4.19)$$

where

$$\begin{aligned} \hat{\pi}_{n-1}(z_{n-1,r}|z_{n-1,-r}) &= \\ &\prod_{i=1}^Q \pi_{n-1,-r_{\{i+1:Q\}}}(z_{n-1,r_i}|z_{n-1,-u}, z_{n-1,r_{\{1:i-1\}}}, \rho'_n). \end{aligned} \quad (4.20)$$

According to the resampling scheme given in Section.4.2.3 it is convenient to select $\rho'_n = \rho_n$ in order to construct a good approximation to the optimal backward kernel.

Note that given the same index order $r_1 \dots r_Q$, Eq.(4.20) will have the same functional form with the right most hand side of Eq.(4.18) when $\rho'_n = \rho_n$.

Finally, using the given approximations for the forward and backward kernels the weight update equation can be arranged according to Eq.(4.4) as follows,

$$v^{sq}(z_{n-1}, z_n) = \frac{\gamma_n(z_n)}{\gamma_{n-1}(z_{n-1})} \times \frac{\hat{\pi}_{n-1}(z_{n-1,r}|z_{n-1,-r})}{\hat{\pi}_n(z_{n,u}|z_{n,-u})}. \quad (4.21)$$

In Monte Carlo computations for solving high dimensional complex problems, it is common practice to resort to a collection of kernels rather than committing to a fixed choice. One can define a mixture kernel in the context of a SMC algorithm as follows,

$$K_n(z_{n-1}, z_n) = \sum_{m=1}^M \alpha_n^m(z_{n-1}) K_n^m(z_{n-1}, z_n) \quad (4.22)$$

where $m \in \{1 \dots M\}$ is the mixture label, α_n^m denotes the selection probability of the mixture component at time n , $\sum_{m=1}^M \alpha_n^m(z_{n-1}) = 1$, and $K_n^m(z_{n-1}, z_n)$ denotes the forward kernel corresponding to the m 'th component. In order to circumvent the computational burden of Eq.(4.22), a backward kernel of the form of a mixture is proposed in (Del Moral et al., 2006).

$$L_n(z_n, z_{n-1}) = \sum_{m=1}^M \beta_n^m(z_n) L_n^m(z_n, z_{n-1}) \quad (4.23)$$

where β_n^m is the backward mixture component selection probability at time n , $\sum_{m=1}^M \beta_n^m(z_n) = 1$. Now, this definition enables us to perform importance sampling on an extended space $E \times E \times \mathcal{M}$ by the definition of a latent kernel selector variable \mathbf{M}_n , taking values $\mathcal{M} = \{1 \dots M\}$, $m \in \mathcal{M}$. Consequently the weight function for each mixture component can be expressed as given in Eq.(4.24).

$$v_n(z_{n-1}, z_n, m) = \frac{\gamma_n(z_n)}{\gamma_{n-1}(z_{n-1})} \times \frac{L_{n-1}^m(z_n, z_{n-1}) \beta_n^m(z_n)}{K_n^m(z_{n-1}, z_n) \alpha_n^m(z_{n-1})} \quad (4.24)$$

In our work we define three different algorithms labeled as SMC-1, SMC-2 and SMC-3 each utilizes a different kernel. The SMC-1 algorithm employs the forward kernel given by Eq.(4.13) and updates the weights according to Eq.(4.14).

The SMC-2 algorithm uses a mixture kernel in order to admit the Gibbs sampler to make global moves in the DPM space. When the selection probabilities α and β are chosen equal and independent of the z_n and z_{n-1} respectively, the mixture weight update function for $m = 1$ and $m = 2$ can be given by Eq.(4.14) and Eq.(4.15) respectively.

In SMC-3 we use the mixtures of the forward kernel given by Eq.(4.13) and Eq.(4.17) where the sequential construction enables us to define a backward kernel independent from the modified forward kernel parameter ρ_n . The associated weight update functions are calculated according to Eq.(4.14) and Eq.(4.21) respectively when selection probabilities α and β are equal and chosen independent of z .

4.2.3 Algorithmic details

As denoted before, in our work, we propose an active block to be updated as each new observation arrives. In order to limit the computational cost required at each time step we determine a constant block size Q and index the block with $r_1 \dots r_Q$. Similar block update strategies have also been proposed by Doucet et al. (2006) under the SMC samplers framework. In our scheme the indexes of the active block $r_1 \dots r_Q$ are incremented by Q as each new observation arrives. Whenever the last index value is $r_Q > n$ then we set $r = \{1 \dots Q\}$. Note that according to the sequential construction defined in Eq.(4.18) the approximations will be less accurate for the algorithm SMC-3 with the increasing size Q .

In order to prevent particle degeneracy, in a SMC framework it is required to perform occasional resampling steps when the effective sample size drops below a predefined threshold. Intuitively, this step selects the high weighted particles and discards the low weighted ones. However, discarding the low weighted particles prematurely may prevent an algorithm to explore promising modes of the time evolving posterior distribution. It is quite common in practice, that a mode initially less dominant becomes more pronounced when a larger fraction of the data is processed. Hence for the DPM model, we found it crucial to apply the resampling step on the modified target distribution $\pi(\cdot|\rho)$ instead of the true target posterior π in order to prevent the low weighted particles to be discarded too early.

We calculate the weights as follows: We first calculate the unnormalized weights for the modified target distribution $\pi_n(\cdot|\rho_n)$ according to $w'_n = w_n \times \gamma_n(z_n|\rho_n)/\gamma_n(z_n)$. Assuming that, $\{W_n^{(i)}\}$ represents the normalized weights, we apply systematic resampling if effective sample size, $N_{eff} = 1/\sum_{i=1}^{N_p} (W_n^{(i)})^2$ is below a predefined threshold. Following the resampling, a reweighting step, $w_n = \gamma_n(z_n)/\gamma_n(z_n|\rho_n)$, is being carried out, in order to find the weights approximating to the true target posterior.

4.3 Experimental Results

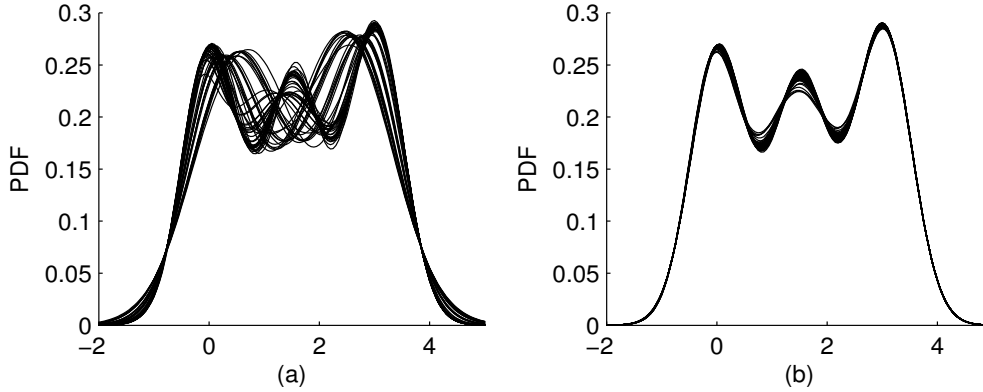
Our goal in this section is to illustrate the effectiveness of the SMC samplers framework for online inference in DPM models. For this purpose, we compare performance of SMC samplers each detailed in Section.4.2.2, namely; SMC-1, SMC-2, SMC-3, Particle filter (PF) (MacEachern et al., 1999; Fearnhead, 2004) and a batch algorithm, Gibbs sampler (GS) (MacEachern, 1994). Performance has been reported in terms of log-marginal likelihoods, mean, variance estimates and respective estimation variances. Mixture density estimates are also provided for visual comparison.

The problem is the standard Gaussian mixture density estimation problem with unknown number of components. Our model is standard and assumes that observations y are drawn from a univariate Gaussian with unknown mean μ and variance σ^2 , $\theta = \{\mu, \sigma^2\}$, where the number of mixtures are unknown. The distribution of the parameters μ and σ^2 are chosen as normal and inverse-gamma, respectively to ensure the conjugacy condition.

Apart from the resampling threshold and the number of particles, several algorithm parameters need to be set: The selection probabilities of the forward and the backward kernels (α and β), active block size (q), and the parameter sequence ρ_n . The selection probabilities determine the shape of the forward and backward kernels therefore an appropriate choice is crucial. Selection probabilities of the forward and backward kernel are set to $\alpha^m = \{0.9, 0.1\}$ and $\beta^m = \{0.9, 0.1\}$ respectively. Note that $m = 2$ corresponds to the modified kernel component and we practically observed that a small weight is often enough to obtain a good mixing property. Increasing the weighting of the modified kernel often increase the algorithms ability to explore new modes. Even a single kernel where $\alpha^m = \{0, 1\}$ can be used for certain dataset where modes are highly isolated. The parameter sequence ρ_n is updated according to a geometric update

Table 4.1: Mixture model parameters.

	Mixture weights	Mean	Std. dev.
Data-1 (D-1)	1/3,1/3,1/3	0,1.5,3	0.5,0.5,0.5
Data-2 (D-2)	1/2,1/6,1/3	0,2,4	0.5,0.5,2.5

**Figure 4.1:** Estimated mixture densities by the (a) SMC-1 and (b) SMC-3 algorithm for 50 Monte Carlo runs.

function where the common parameter is set to $1/150$ and the initial value is set to $\rho_1 = 1$ (Neal, 2001). The active block size q is set to 4. This choice seems to balance well computational burden with inference quality.

To alleviate the degeneracy, we applied systematic resampling scheme. The resampling scheme for SMC-2 and SMC-3 is applied according to Section 4.2.3. For a fair comparison the particle size is selected as $N_p = 1000$ for particle filter, $N_p = 200$ for SMC algorithms and we performed 1000 iterations by Gibbs sampler where the first 300 were used for the burn-in period. All the results are reported for 100 independent Monte Carlo runs. We selected two test sets (D-1 and D-2) generated from a Gaussian mixture model. Each data set has 1000 points, and the results are reported sequentially for 200, 500 and 1000 observations. Both datasets are generated from a model comprising of three mixture components with parameters given in Table 4.1. In order to evaluate the performance of the proposed kernels we performed two tests that aim to assess the mixing property (ability to escape local modes) as well as the consistency and quality of the estimate (bias and low variance).

Sampling based inference schemes get stuck in local modes of the posterior distribution, particularly when the novelty parameter is chosen too small for the given problem. In order to compare the mixing property of the proposed algorithms we set the novelty parameter to a very low value of $\kappa = 0.05$ and compare the

mixture densities estimated by SMC-1 and SMC-3 algorithms, respectively. We performed the test by generating a total of 1000 observations from the model D-1 which comprise three overlapping mixture components. As a gold standard reference we performed a very long Gibbs sampler run and observed that the estimated number of components is 2.16, 3.09 and 3.11 for 200, 500 and 1000 observations consecutively. Figure 4.1 (a) and (b), respectively illustrate the estimated mixture densities obtained by SMC-1 and SMC-3 for 50 Monte Carlo runs. It is clear that SMC-3 can represent all tree components of the mixture density for all runs. However, in nearly half of the runs, the SMC-1 estimates 2 mixture components because it gets stuck to a local mode. The mean estimates of log-marginal likelihood and the number of components are given in Table 4.2 for SMC-1, SMC-2, SMC-3, PF and GS. The results show that the mean estimate of the SMC-2 and SMC-3 are very close to the long run estimate of the Gibbs sampler, however SMC-1, PF and GS underestimate the mean value even when the observation size is 1000 and GS requires a longer burn-in period in order to converge to the true posterior distribution. SMC-2 and SMC-3 are also superior in means of marginal log-likelihoods.

In order to measure the consistency of the proposed algorithms, the performance of the algorithms for different parameter settings are reported in Table 4.3 for D-1 and D-2, respectively. The novelty parameter is set to $\kappa = 0.5$ which avoids the algorithms to stuck at a local solution. The mean estimate for the long Gibbs sampler run is 3.73 for D-1 and 4.63 for D-2 at $n = 1000$. As it is shown in Table 4.3, PF, GS and SMC algorithms provide very close mean estimates to the long run Gibbs sampler for D-1. Estimation variance of the mean estimate for particle filter gradually increases with the

Table 4.2: Estimated average Log-marginal likelihoods, mean values and respective estimation variances (in parenthesis) for SMC-1, SMC-2, SMC-3, PF and GS.

Dataset-1 (D-1), $\kappa = 0.05$				
Algo.	Log-marg.	Estimated Mean		
		200	500	1000
SMC-1	-723.4 (102.8)	2.11 (0.014)	2.51 (0.233)	2.67 (0.243)
SMC-2	-711.2 (4.41)	2.15 (0.006)	3.10 (0.025)	3.10 (0.020)
SMC-3	-711.1 (3.22)	2.15 (0.007)	3.09 (0.011)	3.09 (0.010)
PF	-727.6 (52.9)	2.10 (0.015)	2.35 (0.181)	2.49 (0.249)
GS				2.69 (0.239)

Table 4.3: Estimated average Log-marginal likelihoods, mean values and respective estimation variances (in parenthesis) for SMC-1, SMC-2, SMC-3, PF and GS.

Dataset-1 (D-1), $\kappa = 0.5$				
Algo.	Log-marg.	Estimated mean		
		200	500	1000
SMC-1	-708.9 (1.54)	2.99 (0.038)	3.61 (0.061)	3.71 (0.056)
SMC-2	-708.6 (0.96)	3.00 (0.025)	3.61 (0.044)	3.69 (0.036)
SMC-3	-708.9 (1.20)	2.96 (0.025)	3.60 (0.042)	3.69 (0.038)
PF	-712.4 (9.86)	2.98 (0.041)	3.70 (0.272)	3.79 (0.293)
GS				3.68 (0.055)
Dataset-2 (D-2), $\kappa = 0.5$				
Algo.	Log-marg.	Estimated mean		
		200	500	1000
SMC-1	-1117.3 (0.35)	4.14 (0.035)	4.54 (0.068)	4.65 (0.091)
SMC-2	-1117.3 (0.31)	4.14 (0.021)	4.53 (0.064)	4.63 (0.129)
SMC-3	-1117.2 (0.29)	4.13 (0.019)	4.50 (0.054)	4.58 (0.086)
PF	-1117.7 (0.98)	4.14 (0.030)	4.56 (0.119)	4.73 (0.281)
GS				4.68 (0.095)

observation size and reaches to a value of 0.293 at $n = 1000$ whereas all three SMC samplers achieve approximately 8 times lower estimation variance. It can be concluded that all SMC algorithms provide a significant performance improvement over PF with the same computational cost and they are also more reliable. When we compare the SMC-1, SMC-2 and SMC-3, non of the algorithms outperform the others in means of estimation variance. The results also show that performance of the SMC algorithms and GS are comparable for the dataset D-1 when $\kappa = 0.5$.

A similar performance has also been reported for the dataset D-2. The mean estimate of PF, GS, SMC and the long run Gibbs sampler are very close. All three SMC algorithms and GS provide comparable estimation variance for mean estimates while they are lower for PF. The results obtained for D-1 and D-2 also indicate that, SMC-2 and SMC-3 achieve reliable estimates at different parameter sets.

5. ANNEALED SMC SAMPLERS FOR NONPARAMETRIC BAYESIAN MIXTURE MODELS

In this chapter we aim to further improve the efficiency of the proposed sequential Monte Carlo sampler in Chapter 4 by utilizing annealing strategies under the SMC samplers framework (Ulker et al., 2010a, 2011). The key idea of the method is maintaining an intermediate (annealed) distribution as a surrogate target for the SMC algorithm where resampling is carried out according to this annealed distribution. Consequently, we use this surrogate density as a proposal to the true target where we can calculate the correct weights without any extra computational cost. Intuitively, we are using the SMC machinery to compute a good proposal density. This strategy enables us to maintain a diverse particle set that seems to be crucial in obtaining an efficient sampler.

Due to importance of modeling the multidimensional dependencies in high dimensional datasets we extended the proposed algorithm to the multidimensional Bayesian density estimation problem with unknown number of components where the prior on parameters are conjugate. Conjugate prior on the parameter set is chosen as Normal-scaled inverse Gamma distribution for the univariate model and Normal-inverse Wishart prior is utilized for the multivariate case which is the multidimensional extension of the Normal-scaled inverse Gamma distribution. We applied the proposed algorithm to the emotion recognition from speech where the DPM model allows to simultaneously estimate the number of mixture components as well as the parameters required to represent the emotional class densities.

The proposed algorithm in this section is applied to the emotion recognition from speech problem that has taken an increasing attention in order to build autonomous systems particularly for commercial human-machine applications (Ulker and Günsel, 2011). The model proposed by Ulker and Günsel (2011) eliminated the need for specification of the number of mixture components that represent the data. The real world speech emotion data is obtained from the EMODB public database and the

ability of the algorithm to select the model structure that best suits the training data is illustrated with comparisons.

5.1 SMC Samplers for DPM Models

Let K_n denote the forward kernel that will be used to generate samples from the posterior distribution formulated in Eq.(4.1). We first partition an assignment vector $z_n = \{z_{n,r}, z_{n,d}, z_{n,n}\}$ where r is a subset of $\{1, \dots, n-1\}$, a set of not necessarily consecutive indices, and $d = \{1, \dots, n-1\} - r$. Throughout the text we will call the set $z_{n,r}$ as the active block. We define $u = r \cup \{n\}$, and denote $-u \equiv d$.

Let us define the forward kernel as follows,

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1}, -u}(z_{n,-u}) K_n(z_{n,n}, z_{n,r} | z_{n-1}) \quad (5.1)$$

where $K_n(z_{n,n}, z_{n,r} | z_{n-1})$ is a valid MCMC kernel applying a single Gibbs iteration targeting the full conditional distribution $\pi_n(z_{n,n}, z_{n,r} | z_{n,-u})$.

The corresponding backward kernel can be obtained by substituting Eq.(5.1) into Eq.(2.60) that yields the incremental weight update equation,

$$v_n(z_{n-1}, z_n) = \gamma_n(z_{n-1,r}, z_{n,-u}) / \gamma_{n-1}(z_{n-1}). \quad (5.2)$$

Note that as a consequence of using the MCMC kernel K_n , Eq.(5.2) is independent from the kernel initialization. When the active block set is selected as $r = \{1 \dots n-1\}$, we obtain the update rule Eq.(5.2) introduced by MacEachern et al. (1999) as S4 algorithm. Intuitively, the MCMC kernel updates the active block using a Gibbs sampler and constructs the proposal distribution using the sequence of full conditional distributions.

In a sequential problem the posterior distribution changes over time and new modes of the posterior distribution may emerge as new observations are received. The algorithm must have a good mixing property to explore the modes of the time evolving posterior distribution and to achieve a good approximation to the true target posterior. However, conventional sequential and batch algorithms based on the Gibbs sampler may fail to represent the modes of the true target posterior due to the slow convergence property of the Gibbs samplers. This is particularly when the posterior distribution has a multimodal form where the modes are isolated (Neal, 2001). To deal with this problem, in

the next subsection we introduce an algorithm that converges to the true DPM posterior as the new observations are received sequentially.

5.2 Annealed SMC Samplers for DPM Models

The conventional approach presented in Section 5.1 applies Gibbs moves to each particle in order to obtain weighted samples from a sequence of target distributions denoted as $\pi_1(z_1), \dots, \pi_n(z_n)$. We propose an annealing scheme to improve the efficiency of posterior estimation. In the literature annealing schemes have been widely used to handle isolated modes in batch processing. It is adopted to importance sampling to construct the proposal distribution suitable to sampling of the true target distribution (Neal, 2001).

To achieve our goal let us construct an annealed time evolving target posterior as, $\pi'_1(z_1), \dots, \pi'_n(z_n)$, $k = \{1 \dots n\}$, where π'_k is the annealed target posterior defined as,

$$\pi'_k(z_k) = \pi_k(z_k | \kappa = \alpha_k). \quad (5.3)$$

Annealing is achieved by changing the novelty parameter of the underlying Dirichlet process which is set to α_k in Eq.(5.3). Note that α_k is a parameter of the prior distribution of number of components where a higher value yields higher number of mixtures. The idea behind constructing a sequence of annealed target posterior distribution is to obtain a class of intermediate distributions by selecting a α_k value which is higher than the true model novelty parameter κ and provide a well defined support to the time evolving target posterior. In other words, the annealed distributions can be interpreted as an underlying DPM model of which the parameters are relaxed in order to obtain an annealed posterior which is easy to sample.

In order to sample the sequence of annealed target distributions, let us define a forward kernel as follows,

$$K_n(z_{n-1}, z_n) = \delta_{z_{n-1}, -u}(z_{n,-u}) K_n(z_{n,n}, z_{n,r} | z_{n-1}) \quad (5.4)$$

where $K_n(z_{n,n}, z_{n,r} | z_{n-1})$ is an MCMC kernel which targets the conditional distribution $\pi'_n(z_{n,n}, z_{n,r} | z_{n,-u})$. Using Eq.(2.60), the backward kernel can be written as in Eq.(5.5),

$$L_{n-1}(z_n, z_{n-1}) = \pi'_n(z_{n-1}) K_n(z_{n-1}, z_n) / \pi'_n(z_n) \quad (5.5)$$

and the incremental weights for the annealed target posterior can be obtained as follows,

$$v'_n(z_{n-1}, z_n) = \frac{\gamma'_n(z_n)\pi'_n(z_{n-1,r}|z_{n-1,-u})}{\gamma'_{n-1}(z_{n-1})\pi'_n(z_n, z_{n,r}|z_{n,-u})}. \quad (5.6)$$

where $\pi'_n(z_n) = \gamma'_n(z_n)/Z_n$ and the weights associated with the particles can be calculated according to $w'_n(z_{1:n}) = w'_{n-1}(z_{1:n-1}) \times v'_n(z_{n-1}, z_n)$. Assuming $\{W_n'^{(i)}\}$ represents the normalized weights approximating to $\pi'_n(z_n)$, we perform a resampling step if effective sample size, $N_{eff} = 1/\sum_{i=1}^{N_p} (W_n'^{(i)})^2$, is below a predefined threshold. Finally, in order to approximate the target distribution $\pi_n(z_n)$, we reweight the particles according to $w_n(z_{1:n}) = w'_n(z_{1:n}) \times v_n(z_n)$ where $v_n(z_n) = \gamma_n(z_n)/\gamma'_n(z_n)$.

Specification of the active block size r shown in Eq.(5.6) is an important issue in the design of the proposed sampler. In order to limit the computational cost required at each time step we initially determine a constant block size Q and index the block with $r_1 \dots r_Q$. The indexes of the active block is incremented by Q as each new observation is received. The blocks do not overlap to each other and update scheme is cycled whenever all the clustering labels up to time n are updated. Note that similar block update strategies are also used by Doucet et al. (2006) under the SMC samplers framework.

5.2.1 The annealing parameter

As denoted above the sequence of annealed posterior distributions, $\pi'_1(z_1), \dots, \pi'_n(z_n)$ is constructed by updating the annealing parameter α_n of the underlying DPM model shown in Eq.(5.3). At each time step of the algorithm α_n is updated according to a geometric spacing function

$$\alpha_n = \alpha_{n-1} + c_\alpha(\kappa - \alpha_{n-1}) \quad (5.7)$$

where $\alpha_1 > 0$, $\alpha_{n-1} > \alpha_n$ and c_α is the common parameter that determines the amount of spacing at each time step. Note that, Neal (2001) reported that to change the annealing parameter according to geometric spacing of α_k is suitable when the π'_n varies smoothly with time.

In our framework, we construct the sequence of annealed distributions by setting an initial value α_1 and updating α_n as each new observation arrives. Intuitively the initial

value for α_1 and the common parameter c_α are set empirically in order to form the intermediate distributions that are not too far apart from the true target density π_n . We note that, in conventional annealing approaches, where one modifies the target density gradually, finding the correct schedule is a hard but crucial task. In contrast, in the SMC framework we don't have to choose a schedule very strictly. We are free to choose any forward kernel, provided we compute the corresponding incremental weight – we will be sampling from the correct target at any given time.

5.2.2 Multivariate conjugate prior selection for the DPM model

In this section we derive the posterior distribution of the mixture parameters, $p(\theta_n | z_n, y_n)$, given the labeling vector z_n and the marginal posterior of the clustering labels, $p(z_n | y_n)$, up to a normalizing constant under conjugate settings. We assume that observations are drawn from a multivariate Gaussian distribution with unknown mean vector μ and covariance matrix Σ , $\theta = \{\mu, \Sigma\}$, where the number of mixtures are unknown.

In order to ensure the conjugacy property we utilize a Normal-inverse Wishart prior for the parameter vector $\theta = \{\mu, \Sigma\}$ where,

$$NIW(\tau_0, \omega_0, \Lambda_0, \nu_0) \equiv p(\mu, \Sigma) = p(\Sigma)p(\mu|\Sigma). \quad (5.8)$$

The covariance matrix Σ is inverse Wishart distributed as,

$$\Sigma \sim IW(\Lambda_0^{-1}, \nu_0). \quad (5.9)$$

where Λ_0, ν_0 is the inverse scale matrix and the degrees of freedom respectively. Given the covariance matrix Σ , the mean vector μ is normal distributed as shown in Eq.(5.10)

$$\mu \sim N(\tau_0, \Sigma/\omega_0). \quad (5.10)$$

where τ_0 and Σ/ω_0 are the mean and covariance parameters. According to Eq.(5.10) the covariance defined over the mean value is proportional to the covariance of the Gaussian component and creates a flexible prior structure for the DPM model.

Accordingly, the joint prior distribution, $p(\mu|\Sigma)$, defined over the parameters can be expressed as,

$$p(\mu, \Sigma) \propto |\Sigma|^{((\nu_0+d)/2+1)} \exp\left(-\frac{1}{2}tr(\Lambda_0\Sigma^{-1}) - \frac{\omega_0}{2}(\mu - \tau_0)^T\Sigma^{-1}(\mu - \tau_0)\right) \quad (5.11)$$

Note that, Normal inverse Wishart distribution is a generalization of the Normal-inverse Gamma distribution with the parameters $\theta = \{\mu, \sigma\}$, where μ and σ are scalar quantities.

Let us denote the j 'th mixture component parameters at time index n with, $\theta_j = \{\mu_j, \Sigma_j\}$ where θ_j is distributed according to,

$$\theta_j \sim NIW(\tau_0, \omega_0, \Lambda_0, v_0), \quad (5.12)$$

and the overall joint parameter vector is $\theta = \{\theta_1 \dots \theta_{k_n}\}$.

The posterior distribution of the j 'th cluster parameters $\{\mu_j, \Sigma_j\}$ is also Normal inverse Wishart distributed, $NIW(\mu_j, \Sigma_j | \tau_j, \omega_j, \Lambda_j, v_j)$, where the parameters are calculated as,

$$\tau_j = \frac{\omega_0 \tau_0 + n_j \bar{y}_j}{\omega_0 + n_j} \quad (5.13)$$

$$\omega_j = \omega_0 + n_j \quad (5.14)$$

$$\Lambda_j = \Lambda_0 + \sum_{i=1}^{n_j} (y_{j,i} - \bar{y}_j)(y_{j,i} - \bar{y}_j)^T + \frac{\omega_0 n_j}{\omega_0 + n_j} (\bar{y}_j - \tau_0)(\bar{y}_j - \tau_0)^T \quad (5.15)$$

$$v_j = v_0 + n_j. \quad (5.16)$$

In the above equations n_j is the number of observations, $y_{j,i}$ is index to each observation and \bar{y}_j is the mean vector of the observations in the ' j 'th cluster. The marginal of the posterior distributions on the mean vector μ_j and the covariance matrix Σ_j can be computed analytically by the inverse Wishart and student-t distributions respectively, as shown below,

$$p(\Sigma_j | z_n, y_n) = IW(\Lambda_j^{-1}, v_j) \quad (5.17)$$

$$p(\mu_j | z_n, y_n) = t_{v_j-d+1} \left(\tau_j, \frac{\Lambda_j}{\omega_j(v_j - d + 1)} \right). \quad (5.18)$$

Accordingly if the parameters $\theta_j \in \{\Sigma_j, \mu_j\}$, $j = \{1 \dots k\}$, in Eq.(2.69) are integrated out, the posterior probability $p(z_n | y_n)$ of the assignment z_n can be expressed up to a proportionality as follows,

$$p(z_n | y_n) \propto p(z_n) \times \prod_{j=1}^{k_n} \frac{\Gamma_d(v_j/2) \Lambda_0^{v_0/2} \omega_0^{d/2}}{\pi^{n_j d/2} \Gamma_d(v_0/2) \Lambda_j^{v_j/2} \omega_j^{d/2}} \quad (5.19)$$

where $p(z_n)$ is the prior on clustering assignment vector z_n , Γ_d is the multidimensional Gamma function and d is the dimension of the observation space.

Table 5.1: Mixture model parameters.

	p_1, p_2, p_3	μ_1, μ_2, μ_3	$\sigma_1, \sigma_2, \sigma_3$
Data-1 (D-1)	1/3,1/3,1/3	0,1.5,3	0.5,0.5,0.5
Data-2 (D-2)	1/2,1/6,1/3	0,2,4	0.5,0.5,2.5

5.3 Test Results

Our goal in this section is to illustrate the effectiveness of the annealed SMC samplers for online inference in DPM models for univariate and multivariate datasets.

5.3.1 Univariate density estimation problem

For this purpose, we compare performance of three samplers namely; the SMC-G which utilizes conventional Gibbs moves on the DPM space (MacEachern et al., 1999), the proposed SMC sampler (SMC-A), the SMC-M algorithm that utilize a mixture of Gibbs moves and approximate Gibbs moves based on sequential approximation (Ulker et al., 2010b) and the Particle filter (PF) (Fearnhead, 2004). Performance has been reported in terms of log-marginal likelihoods, mean estimates and respective standard errors. Mixture density estimates are also provided for visual comparison.

Algorithms are evaluated on a univariate infinite dimensional Gaussian mixture density estimation problem. Observations are drawn from a univariate Gaussian with $\theta = \{\mu, \sigma^2\}$ where μ is the mean and σ^2 is the variance. The conjugate prior distributions are chosen as normal and inverse-gamma respectively.

To alleviate the degeneracy, a systematic resampling scheme is applied for sequential algorithms when $N_{eff} < 3/4N_p$. For a fair comparison the number of particles is selected as $N_p = 1000$ for PF, $N_p = 100$ for SMC-A algorithm and $N_p = 200$ for the SMC-M algorithm where the active block size Q is set to 9 and 4 for the SMC-A and SMC-M algorithms respectively. Note that block size determines the approximation introduced by the kernels for the SMC-M algorithm (Ulker et al., 2010b). The results are reported for 100 independent Monte Carlo runs for each model. The initial annealing parameter for annealed target distribution is set to $\alpha_1 = 1$ and it is geometrically updated according to Eq.(5.7) at each time step where the common parameter, c_α , is set to $1/100$.

Two test sets (D-1 and D-2) are generated from a Gaussian mixture model comprising three mixture components with parameters given in Table 5.1 where μ_i, σ_i , and p_i , for

$i \in \{1 \dots 3\}$, denote the mean, standard deviation and the mixture weight for each component, respectively. In order to evaluate the performance on real data, we also performed the tests on the speech data set (D-3) publicly available. Reported results are obtained for the emotional state “sad” where the actual number of mixture components is priorly unknown. Each test set has a total of 1000 points and the results are reported sequentially for 200, 500, and 1000 samples.

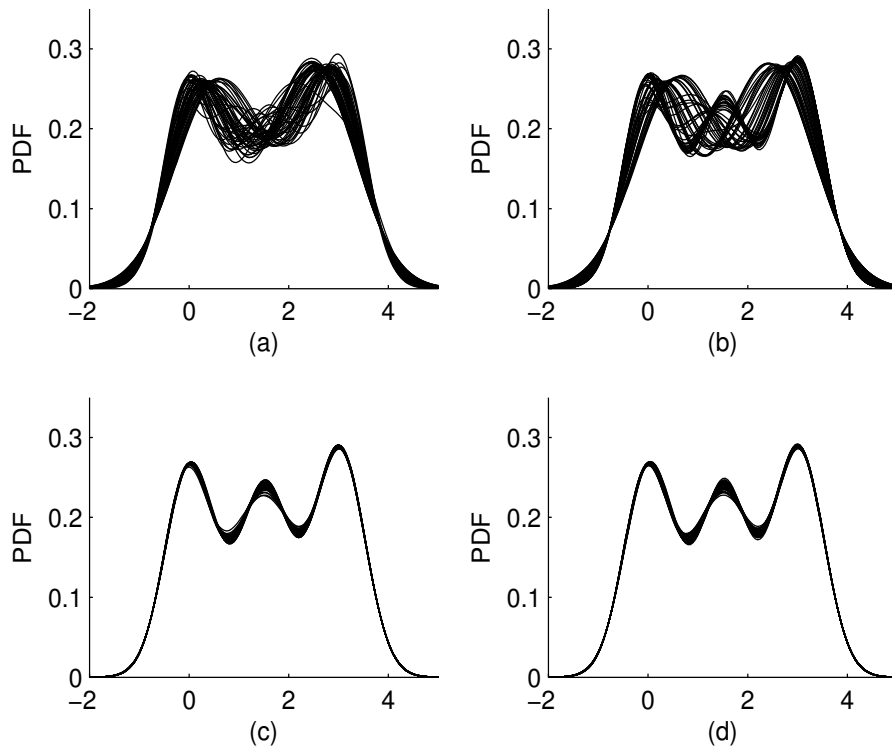


Figure 5.1: Estimated mixture densities by the (a) PF, (b) SMC-G, (c) SMC-A (d) SMC-M algorithm for 50 Monte Carlo runs. SMC-A and SMC-M represent all tree components of the mixture density in all runs.

In order to illustrate the mixing capability of the proposed algorithm we set the novelty parameter to a very low value of $\kappa = 0.05$. Note that a low κ will probably cause the posterior to have isolated modes hence this test aims to assess the mixing property (ability to escape local modes) of the algorithms. We performed the test by generating a total of 1000 observations from the model D-1 which comprise three overlapping mixture components. As a gold standard reference we performed a very long Gibbs sampler run and observed that the estimated number of components is 2.16, 3.09 and 3.11 for 200, 500 and 1000 observations consecutively. In Figure 5.1, the mixture

Table 5.2: Estimated average Log-marginal likelihoods, mean values and respective Monte Carlo standard errors (in parenthesis).

D-1					
Algo.	κ	Log-marg.	Mean Estimate		
			200	500	1000
SMC-G	0.05	-723.4 (10.1)	2.11 (0.118)	2.51 (0.152)	2.67 (0.493)
SMC-A	0.05	-710.8 (1.61)	2.15 (0.070)	3.07 (0.077)	3.07 (0.071)
SMC-M	0.05	-711.1 (1.78)	2.15 (0.083)	3.09 (0.104)	3.09 (0.115)
PF	0.05	-727.6 (7.27)	2.10 (0.122)	2.35 (0.425)	2.49 (0.491)
D-2					
SMC-G	0.5	-1117.3 (0.59)	4.14 (0.187)	4.54 (0.260)	4.65 (0.266)
SMC-A	0.5	-1117.3 (0.52)	4.14 (0.158)	4.53 (0.244)	4.63 (0.330)
SMC-M	0.5	-1117.2 (0.53)	4.13 (0.137)	4.50 (0.232)	4.58 (0.293)
PF	0.5	-1117.7 (0.99)	4.14 (0.173)	4.56 (0.345)	4.73 (0.530)
D-3					
SMC-G	0.05	-2052.1 (2.26)	2.50 (0.197)	3.09 (0.447)	3.58 (0.452)
SMC-A	0.05	-2050.6 (0.30)	2.61 (0.424)	3.40 (0.378)	4.04 (0.320)
SMC-M	0.05	-2051.1 (1.39)	2.60 (0.303)	3.35 (0.360)	4.01 (0.401)
PF	0.05	-2052.8 (2.54)	2.48 (0.251)	3.06 (0.500)	3.39 (0.573)

densities are plotted for each run of the PF, SMC-G, SMC-A and SMC-M algorithms, respectively. It is clear that SMC-A and SMC-M can represent all 3 components of the mixture density in all runs of the algorithms whereas SMC-G and PF commonly gets stuck at a local mode and fits 2 mixture components to the data for several runs (more than the half) of the algorithm. We also reported the log-marginal likelihood, mean estimate of the number of components and respective standard errors (in parenthesis) in Table 5.2 for SMC-G, SMC-A, SMC-M and PF. The results illustrate that SMC-A and SMC-M are able to converge to the 3 components for a small number of observations, however the SMC-G and PF algorithms do not converge to the true posterior even when the observation size is 1000. It is also clear that SMC-A has much lower standard error compared to SMC-G and PF in means of log-marginal likelihoods and the mean estimates whereas a slight improvement is achieved over SMC-M.

In order to examine dependency of the performance of the algorithms on different datasets and parameter settings, we set the novelty parameter to $\kappa = 0.5$ and report the results in Table 5.2 for dataset D-2. It is clear that PF and SMC algorithms provide very close mean estimates. However, SMC-G, SMC-A and SMC-M can achieve significantly lower standard error compared to PF at $n = 1000$. This result shows that SMC algorithms are more reliable with the same computational cost.

Moreover SMC-A achieves comparable performance to SMC-G and SMC-M in means of standard error when $\kappa = 0.5$ while it provides similar mean estimates.

Finally we compared the performance for dataset D-3, where the novelty, initial annealing and the common parameter are set to $\kappa = 0.05$, $\alpha_1 = 0.25$ and $c_\alpha = 1/2000$ respectively. As a gold standard reference the results of a very long Gibbs sampler run are found as 2.53, 3.35, 4.10 for 200, 500 and 1000 observations consecutively. The results given in Table 5.2 shows that the SMC-A and SMC-M provides closer estimates to the long Gibbs sampler run particularly when $n = 1000$ whereas SMC-G and PF underestimates the mean value. Similarly SMC-A outperforms SMC-G and PF in means of log marginal likelihood and achieves lower standard error compared to the SMC-M algorithm.

5.3.2 Multivariate density estimation problem

In order to evaluate the performance of the algorithms on the multivariate density estimation problem we reported the test results of the SMC-G which utilizes conventional Gibbs moves on the DPM space (MacEachern et al., 1999) and the proposed multivariate annealed SMC sampler (SMC-MA) for two different multivariate datasets. These two test sets (M-1 and M-2) are respectively generated from two and five dimensional multivariate Gaussian mixture that comprise of three equally weighted, $\{p_1, p_2, p_3\} = 1/3$, mixture components. The mean vectors of each component are $\mu_1 = \{0, 0\}$, $\mu_2 = \{-2, -2\}$, $\mu_3 = \{2, 2\}$ for the two dimensional case and $\mu_1 = \{0, 0, 0, 0, 0\}$, $\mu_2 = \{-2, -2, -2, -2, -2\}$, $\mu_3 = \{2, 2, 2, 2, 2\}$ for the 5 dimensional data. The covariance matrices Σ_1 and Σ_3 are chosen as diagonal matrices where the diagonal elements equals to one, $\sigma^2 = 1$, and the third matrix Σ_2 is selected as a full covariance matrix.

The initial annealing parameter is set to $\alpha_1 = 1$ and it is geometrically updated according to Eq.(5.7) at each time step where the common parameter, c_α , is set to $1/1000$, for both of the multidimensional test cases.

In order to illustrate the mixing capability of the proposed algorithm for the two dimensional data, we set the novelty parameter to the value of $\kappa = 0.05$ and plot the mixture densities estimated by a single particle generated by the SMC-G and SMC-MA algorithms in Figure 5.2(a) and Figure 5.2(b) respectively. It is clear that

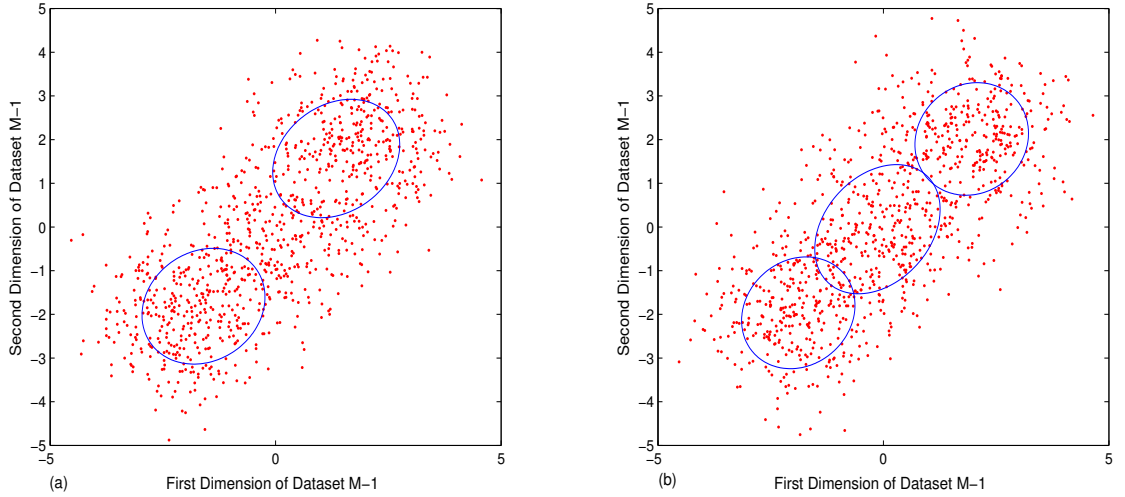


Figure 5.2: Observations (red dots) on 2-D space and 50% confidence intervals of the Estimated mixture densities by the (a) SMC-G, (b) SMC-MA algorithms for a single Monte Carlo run. SMC-MA represent all tree components of the mixture density.

Table 5.3: Estimated average Log-marginal likelihoods, mean values and respective estimation variances (in parenthesis).

M-1					
Algo.	κ	Log-marg.	Mean Estimate		
			200	500	1000
SMC-G	0.05	-1421.0 (740.1)	2.15 (0.007)	2.24 (0.105)	2.49 (0.174)
SMC-MA	0.05	-1410.9 (420.6)	2.27 (0.051)	2.71 (0.130)	2.99 (0.103)

SMC-MA can represent all 3 components of the mixture density whereas SMC-G gets trapped at a local mode and fits 2 mixture components to the data. We also reported the log-marginal likelihood, mean estimate of the number of components and respective estimation variances (in parenthesis) in Table 5.3 for SMC-G and SMC-MA . The results illustrate that SMC-MA is able to converge to the 3 mixture components, whereas the SMC-G algorithm is not able to converge to the true posterior even when the observation size is 1000. It is also clear that SMC-MA has lower estimation variance compared to SMC-G in means of log-marginal likelihoods and mean estimates.

In order to evaluate the performance of the proposed algorithm for higher dimensional data we reported log-marginal likelihood and mean estimate of the number of components as well as the estimation variances (in parenthesis) for the dataset M-2 in Table 5.4. The results indicate that, similar to the results given for two

Table 5.4: Estimated average Log-marginal likelihoods, mean values and estimation variance (in parenthesis).

M-2					
Algo.	κ	Log-marg.	Mean Estimate		
			200	500	1000
SMC-G	0.05	-2535.8 (3101.2)	2.18 (0.147)	2.18 (0.140)	2.22 (0.218)
SMC-MA	0.05	-2434.3 (2432.1)	2.43 (0.204)	2.96 (0.051)	2.97 (0.054)

dimensional data, the proposed algorithm, SMC-MA is able to converge to the 3 mixture components. However the conventional algorithm, SMC-G gets trapped to 2 components even when the observation size is 1000. We also observed that SMC-MA has higher log marginal likelihood and lower estimation variance compared to SMC-G.

6. DIRICHLET PROCESS MIXTURES FOR TIME SERIES CLUSTERING

In this section we construct an infinite dimensional model based on Dirichlet process mixtures in order to cluster a time series where the number of clusters k and the parameters of each cluster are priorly unknown. Let us consider a toy, time series clustering problem illustrated in Figure 6.1(a), where discrete samples are generated by a two state Markov Chain switching its parameters at certain change points. Given

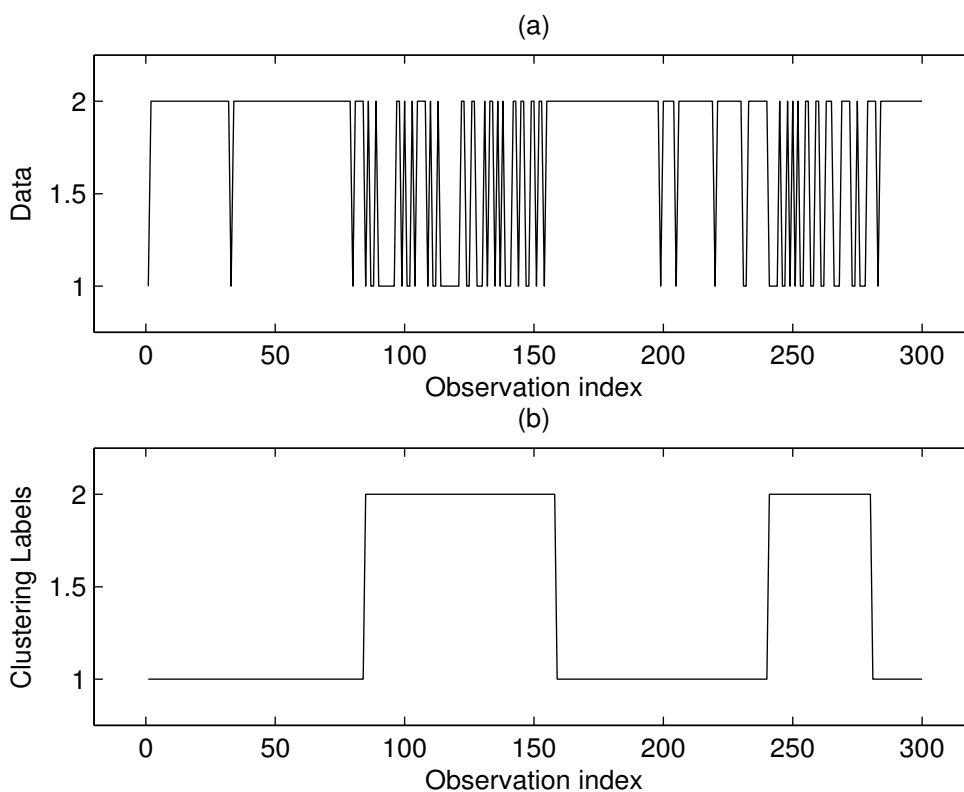


Figure 6.1: (a) Discrete time series generated from a mixture of Markov Chains (b) Switching labels that select the active Markov chain.

the data in Figure 6.1(a), it is easy to identify that there is a total of two Markov chains interchanging between each other at the change points 80, 160, 240 and 280 consecutively, and the clustering labels associated with each observation are given in Figure 6.1(b). Assuming that the clustering labellings are known, it is straightforward

to estimate the model parameters, such as transition matrix and the initial probability of each chain.

However in real world problems we encounter with situations where the model parameters, change point times (arrival times) and the model order is priorly unknown. In order to alleviate such cases, the given problem can be defined with a model that switches its parameters within an infinite set of $\theta = \{\theta_1, \dots, \theta_k\}$, $k \in \{1 \dots \infty\}$, selected according to the clustering labellings $z_i \in \{1 \dots k\}$ at time intervals τ_i , $i = \{1, \dots, n\}$ which are determined according to an arbitrarily selected distribution. Solution to the given problem can be quite challenging and online algorithms are highly desired due to real time and large scale data processing requirements.

In order to solve the problem defined above, in our work we propose a Bayesian model based on Dirichlet process mixtures, that estimates the number of clusters, model parameters and the arrival times in a time series data under the assumption that there is no temporal correlation between consecutive clustering labellings and the order of the labels is not important. The model is not restricted to a certain type of data and can be applied for online clustering of discrete or continuous time stochastic processes. Since there is no closed form solution to the proposed model, we solved the proposed model by an online inference scheme based on the sequential Monte Carlo sampling methodology.

In the sequel we will explain the proposed model structure and the sampling algorithm designed for sequential inference of the model posterior distribution. We applied the algorithm to a time series clustering problem where the data stream is generated from a mixture of Markov Chains which is an important time series clustering problem particularly in the area of network traffic analyzing and bio-informatics. We reported the clustering performance of the algorithm for various synthetic datasets and evaluated the performance on the audio data.

6.1 Model Construction

Let us denote the observation sequence received until time n by $y_n = \{y_{n,1}, \dots, y_{n,n}\}$. Assuming that m_n represent the total number of changepoints and $\tau_{n,i}$, $i \in \{1, \dots, m_n\}$ is the index to i 'th changepoint, the vector of discrete arrival times until time n can be represented as $\tau_n = \{\tau_{n,1} \dots \tau_{n,m_n}\}$. Note that m_n is a random

variable by construction and let us characterize the time series clustering problem as a semi-Markovian structured change point model.

The model associates each clustering variable $z_{n,i}$ to the change point $\tau_{n,i}$ where $z_{n,i} \in \{1, \dots, k_n\}$, and $k_n \in \{1 \dots m_n\}$ represent the number of clusters at time n . The vector of cluster variables is defined as $z_n = \{z_{n,1}, \dots, z_{n,m_n}\}$ and the clustering variable $z_{n,i}$ selects the model parameter, $\theta_{n,z_{n,i}}$, that is active between the time intervals $\tau_{n,i}$ and $\tau_{n,i+1}$, from the parameter set $\theta_n = \{\theta_{n,1}, \dots, \theta_{n,k_n}\}$.

The proposed model assumes that the arrival times are distributed according to an arbitrarily selected prior distribution $p(\tau_n)$, and the cluster parameters are independently drawn from the prior $p(\theta_n)$. The observation groups, $y_{n,\{\tau_i:\tau_{i+1}\}}$, $i \in \{1, \dots, m_n - 1\}$ are independent of each other conditional on the assignment variable $z_{n,i}$. Under the given assumptions, DPM posterior density $\pi_n(x_n)$ can be expressed as,

$$\pi_n(x_n) \propto p(z_n)p(\tau_n) \prod_{j=1}^{k_n} p(\theta_{n,j}) \prod_{i=1}^{m_n-1} g(y_{n,\{\tau_i:\tau_{i+1}\}}|\theta_{n,z_{n,i}})g(y_{n,\{\tau_{m_n}:n\}}|\theta_{n,z_{n,m}}) \quad (6.1)$$

where $x_n = \{z_n, \theta_n, \tau_n\}$.

Since the order of the cluster labellings is not important as in a conventional DPM model, the prior on clustering variable vector z_n is formulated by Eq.(6.2) in a recursive way,

$$p(z_{n,i+1} = j | z_{n,\{1:i\}}) = \begin{cases} \frac{l_j}{i+\kappa}, & j = 1, \dots, k_i \\ \frac{\kappa}{i+\kappa}, & j = k_i + 1 \end{cases} \quad (6.2)$$

where k_i is the number of clusters in the assignment $z_{n,\{1:i\}}$. l_j is the number of observations that $z_{n,\{1:i\}}$ assigns to cluster j and κ is a 'novelty' parameter.

The prior distribution on the arrival times, $p(\tau_n)$, has the Markov property and can be explicitly expressed in the form,

$$p(\tau_n, m_n) = p(\tau_{n,m_n^+} > n) \prod_{i=1}^{m_n-1} p(\tau_{n,i+1} | \tau_{n,i}). \quad (6.3)$$

where τ_{n,m_n^+} denotes the arrival time of the $m + 1$ 'th change point. The probability, $p(\tau_{n,m_n^+} > n)$ ensures that no change points occur between time τ_{n,m_n} and n .

In order to obtain an efficient inference scheme we selected a conjugate prior model such that given $\{z_n, \tau_n\}$, the parameter θ_n can be integrated out and the model posterior, $p(z_n, \tau_n | y_n)$, can be calculated up to a normalizing constant. In the following section

we will design an algorithm that enable us to sample the distribution, $p(z_n, \tau_n | y_n)$, and estimate the parameters θ_n in closed form using the set of samples and associated weights. The designed inference scheme also allow us to select an arbitrary distribution on the sojourn times, $\tau_{i+1} - \tau_i \sim p(\tau_{n,i+1} | \tau_{n,i})$.

6.2 Sequential Monte Carlo Sampler for the Proposed Model

Under the sequential Monte Carlo samplers framework we designed an algorithm that is able to represent the proposed model posterior distribution with a set of weighted samples whenever a new observation arrives. For the notation simplicity and clearance, we use the time evolving representation given in (Del Moral et al., 2006) and in order to achieve a simpler inference scheme we utilized conventional particle filtering kernels for sampling the target posterior density.

Our aim is to sample from a sequence of target densities evolving with a countable index n , $\pi_1(x_1) \dots \pi_n(x_n)$, each defined on a common measurable space (E_n, \mathcal{E}_n) where $x_n \in E_n$. Let us define the sequence of target densities $\pi_1(x_1) \dots \pi_n(x_n)$ and the corresponding proposal distributions as $\eta_1(x_1) \dots \eta_n(x_n)$. According to importance sampling theory, the unnormalized importance weight w_n at time n can be defined as,

$$W_n = \frac{\gamma_n(x_n)}{\eta_n(x_n)} \quad (6.4)$$

where γ_n is the unnormalized target distribution, $\pi_n = \gamma_n/Z$, and Z is the normalizing constant.

Under the conventional particle filtering framework, it is possible to obtain a incremental weight update equation as

$$W_n = W_{n-1} \times w_n \quad (6.5)$$

where it is possible to derive the incremental weights as given in Eq.(6.6)

$$w_n = \frac{\gamma_n(x_n)\eta_{n-1}(x_{n-1})}{\eta_n(x_n)\gamma_{n-1}(x_{n-1})}. \quad (6.6)$$

The proposal distribution, $\eta_n(x_n)$, can be explicitly written of the form

$$\eta_n(x_n) = \int \eta_{n-1}(x_{n-1})K(x_{n-1}, x_n)dx_{n-1}. \quad (6.7)$$

Computation of the importance distribution $\eta_n(x_n)$ for $n > 1$ requires an integration with respect to x_{n-1} thus a closed form solution to weight update equation given by

Eq.(6.6) is not available except for specifically designed kernels. To appreciate this limitation, we will consider the form of the kernel explicitly.

6.2.1 Kernel selection

In this section we design a proposal kernel in order to obtain weighted samples from the posterior distribution of the defined model. As discussed above under the particle filtering framework it is not possible to calculate the incremental weight update function for proposal kernels which update the particle history. Though, it is possible to calculate the weight update function for an arbitrarily selected kernel under the SMC samplers framework (Del Moral et al., 2006), for simplicity we just utilized the common particle filtering approach for sequential inference in the proposed model.

In order to obtain an efficient sampler the prior distribution on the model parameters θ and the likelihood function are selected as a conjugate pairs. The conjugate model enables us to factorize the model posterior as, $p(\theta_n|z_n, \tau_n, y_n)p(z_n, \tau_n|y_n)$ where the first factor is calculated in closed form and we just need to sample the clustering variables z_n and τ_n instead of the whole parameter space. Hence we define the proposal kernel on the space $x_n = \{z_n, \tau_n\}$.

The proposal kernel $K(x_{n-1}, x_n)$ given in Eq.(6.8) aims to sample the change point $\tau_{n,m}$ as well as the clustering label, $z_{n,m}$, associated with the observations $y_{n,\tau_{n,m}:n}$ according to a mixture kernel that comprise two type of moves. The left hand side of the Eq.(6.8) denotes the proposal of a new change point, τ_m , and its associated clustering variable z_m at time n whereas the right hand side depicts no new change point to the state variable x_{n-1} .

$$\begin{aligned}
K(x_{n-1}, x_n) = & \alpha_1(x_{n-1})\pi(z_{n,m}|z_{n,-m}, \tau_{n,m} = n, \tau_{n,-m})\delta_{x_{n-1}}(x_{n,-m}) \\
& + \alpha_2(x_{n-1})\delta_{x_{n-1}}(x_n)
\end{aligned} \tag{6.8}$$

In Eq.(6.8), the notation $x_{n,-m}$ denotes the components of x_n excluding the m 'th element, $x_{n,m}$, and $\{\alpha_1, \alpha_2\}$ denotes the kernel mixture weights. The kernel mixture weights, α_1 and α_2 , that ensures a full conditional proposal kernel, $\pi_n(\cdot|x_{n-1})$, can be

calculated according to the equation given by Eq.(6.9),

$$\alpha_1 = \frac{\gamma_n(\tau_{n,m} = n, z_{n,-m}, \tau_{n-m})}{Z_\alpha} \quad (6.9)$$

$$\alpha_2 = \frac{\gamma_n(z_n, \tau_n)}{Z_\alpha} \quad (6.10)$$

and it is straightforward to calculate the normalizing constant Z_α with the help of the property $\alpha_1 + \alpha_2 = 1$.

6.2.2 Weight update function

In order to calculate the unnormalized particle weights W_n at each time step we calculate the incremental weight w_n by replacing Eq.(6.8) in Eq.(6.6) as shown in the following,

$$w_n = \frac{\gamma_n(x_{n-1})}{\gamma_{n-1}(x_{n-1})}. \quad (6.11)$$

Note that, if no new change point is introduced by the proposal kernel at time step n , the dimensionality of the model do not change, $x_n \equiv x_{n-1}$, hence we just need to reweigh the particles according to the Eq.(6.11) in a single algorithm iteration.

The designed SMC algorithm for time series clustering can be summarized as follows.

SMC Algorithm for Clustering Time Series

- step 1 : Initialize $n = 1, \tau_1 = 0, z_1 = 1$ where $W_n^i = 1/Np$.
- step 2 : $n = n + 1$. Resample the weights W_{n-1} if $N_{eff} < Thr$.
- step 3 : For $i = 1$ to N_p draw $x_n^i \sim K(\cdot|x_{n-1})$ given by Eq(6.8)
- step 4 : For $i = 1$ to N_p calculate the incremental weight w^i according to

$$w_n^i = \frac{\gamma_n(x_{n-1})}{\gamma_{n-1}(x_{n-1})}.$$

- step 5 : Update the particle weight according to

$$W_n^i = W_{n-1}^i \times w_n^i$$

Normalize the particle weights $\sum_{i=1}^{N_p} W_n^i = 1$.

(Iterate through step 2 to step 5 at each time step.)

6.3 An application: Clustering Mixture of Markov Chains

In this section we apply the proposed model to a time series clustering problem where the observations are generated from a mixture of finite discrete Markov chains. We assume that the number of chains and the arrival times that denote the transitions from one chain to the other are priorly unknown and the rank of the transition matrix is R and do not change over time. Our aim is to sequentially cluster the data stream and estimate the arrival times as well as the number of discrete Markov Chains and the parameters (Transition matrix, T , and the initial distribution P).

In order to define the model we just need to construct the prior structure on the cluster parameters, θ_n , and the arrival times τ_n . Let $\theta_{n,j} = \{P_j, T_j\}$, $j = \{1, \dots, \infty\}$ denote the parameters of the j 'th cluster where $T_j(p, r)$, $(p, r) \in \{1, \dots, R\}$ denotes the random variable that denotes the p 'th row and r 'th column of the transition matrix T_j . Similarly, $P_j(r)$ represents the r 'th element of the j 'th initialization array P_j . Since Dirichlet distribution is conjugate over the multinomial distribution, the initialization array P_j and each row of the transition matrix, $T_j(p, \{1, \dots, R\})$, $j = \{1, \dots, \infty\}$ are selected independently distributed according to,

$$P_j \sim Dir(\kappa_P), \quad T_j(p, \{1, \dots, R\}) \sim Dir(\kappa_T), \quad j = \{1, \dots, \infty\} \quad (6.12)$$

where κ_P and κ_T are the concentration parameter defined for the initialization array and transition matrix respectively.

The likelihood function corresponding to the observation model is defined as given in Eq.(6.13)

$$g(y_{n, \{\tau_{n,i} : \tau_{n,i+1}\}} | \theta_{n, z_{n,i}}) = \prod_{p=1}^R \prod_{r=1}^R (T_{z_{n,i}}(p, r))^{N_{z_{n,i}}(p, r)} (P_{z_{n,i}}(r))^{N_{z_{n,i}}(r)} \quad (6.13)$$

where $N_{z_{n,i}}(p, r)$ denotes the total number of times a transition from p to r occurs when the Markov Chain labeled $z_{n,i}$ is active and $N_{z_{n,i}}(r)$ denotes the total number of times the chain starts with r .

Further, we define negative binomial prior distribution on the sojourn times that practically determine the change point times. The negative binomial, $NB(a, b)$, can be defined as the distribution of random number of successes in a sequence of Bernoulli trials with a probability of success until a total of b failures occur. In our work prior on the sojourn times, $\tau_{n,i} - \tau_{n,i-1} \sim NB(a, b)$, $i = \{1, \dots, m_n - 1\}$, are chosen as

negative binomial distribution that is a well suited discrete distribution to model the arrival times $\tau_{n,i}$, $i = \{1, \dots, m_n - 1\}$.

According to the defined model the posterior distribution of $\pi_n(z_n, \tau_n)$ can be calculated up to the normalizing constant as follows,

$$\begin{aligned} \pi_n(z_n, \tau_n) &\propto p(z_n)p(\tau_n) & (6.14) \\ &\times \prod_{j=1}^{k_n} \prod_{p=1}^R \frac{\Gamma(\sum_{r=1}^R \alpha_r)}{\Gamma(\sum_{r=1}^R N_j(p, r) \sum_{r=1}^R \alpha_r)} \frac{\prod_{r=1}^R \Gamma(\alpha_j + N_j(p, r))}{\prod_{r=1}^R \Gamma(\alpha_r)} \\ &\times \prod_{j=1}^{k_n} \frac{\Gamma(\sum_{r=1}^R \alpha_r)}{\Gamma(\sum_{r=1}^R P_j(r) \sum_{r=1}^R \alpha_r)} \frac{\prod_{r=1}^R \Gamma(\alpha_r + P_j(r))}{\prod_{r=1}^R \Gamma(\alpha_r)} \end{aligned}$$

where $N_j(p, r)$ is the total number of times a transition from p to r occurs in the Markov Chains indexed with k , and $P_j(r)$ is the number of times the chain is initialized with r .

6.4 Experimental Results

In this section we evaluate the performance of the proposed algorithm on time series clustering problem where the data is generated from mixture of Markov Chains where the number of components and corresponding parameters are unknown.

The observations are processed sequentially and the parameter estimates are updated whenever a new observation is received. The clustering labels associated with each observation, estimated model parameters and the change point estimation performance are reported for various synthetic datasets. In order to evaluate the performance of the proposed algorithm on the real world datasets we also reported the estimation results for the audio signal clustering.

The first dataset 'DS-1' is a synthetic data generated from mixture of two Markov chains where the initial state probability of the chains are set to $\mathbf{P}_1 = \mathbf{P}_2 = [1/3 \ 1/3 \ 1/3]$ and the transition matrices are selected as,

$$\mathbf{T}_1 = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.25 & 0.25 & 0.5 \\ 0.5 & 0.25 & 0.25 \end{bmatrix} \quad \mathbf{T}_2 = \begin{bmatrix} 0.25 & 0.25 & 0.5 \\ 0.5 & 0.25 & 0.25 \\ 0.25 & 0.5 & 0.25 \end{bmatrix}$$

The sojourn time between consecutive change points are generated according to a negative binomial distribution, $\tau_{i+1} - \tau_i \sim NB(r, p)$ where the distribution parameters are determined as $r = 3$ and $p = 0.001$. Whenever a new changepoint occurs, the

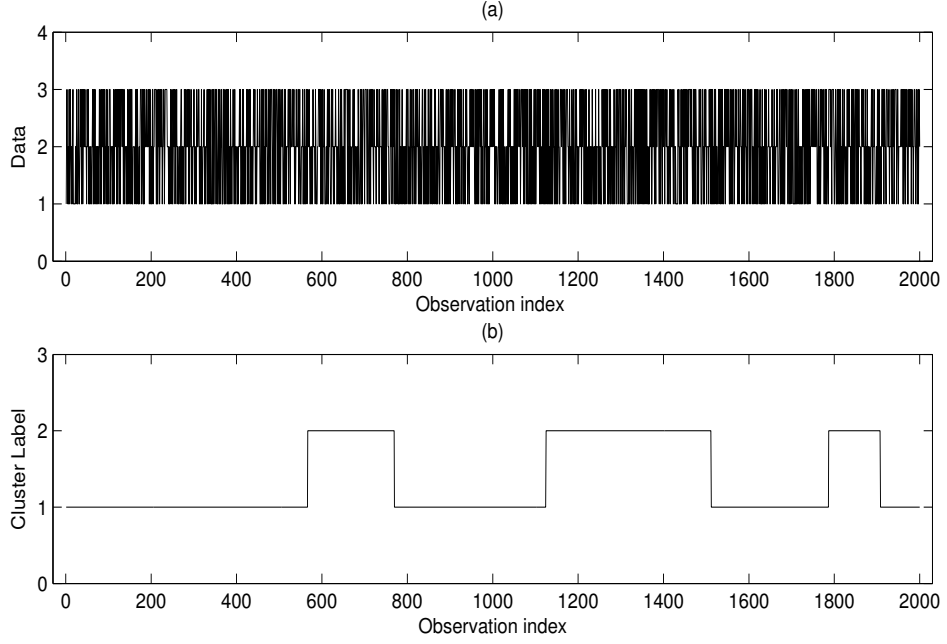


Figure 6.2: (a) Synthetically generated data and (b) associated cluster labels.

clustering label is switched between the labels $\{1, 2\}$ in order to select the active transition matrix, \mathbf{T}_1 or \mathbf{T}_2 .

In order to cluster the data stream, the proposed algorithm sequentially estimate the number of chains, k_n , the parameters of each chain $\theta_{n,j} = \{T_j, P_j\}$, $j \in \{1 \dots k_n\}$ and the labellings, z_n , of the observation received until time n . Non-informative Dirichlet priors are defined over the parameters, $\theta_j = \{P_j, T_j\}$, $j \in \{1, \dots, \infty\}$ as given in Eq.(6.15) where all the components of the Dirichlet distribution equal to each other.

$$\kappa_P = 50, \quad \kappa_T = 1 \tag{6.15}$$

Note that non-informative prior do not favor any component of the Dirichlet distribution over the other and when $\kappa = 1$ the prior distribution is uniform over all points in its support. The values greater than one, $\kappa > 1$, create an evenly distributed prior and conversely values lower than one, $\kappa < 1$, the distribution is sparse that allow the algorithm to overweight some components over the others. In our simulations we selected the parameter on the transition matrix as, $\kappa_T = 1$, to design an algorithm that is able to handle different parameter settings whereas a tight distribution function is defined over the initialization array by selecting $\kappa_P = 50$.

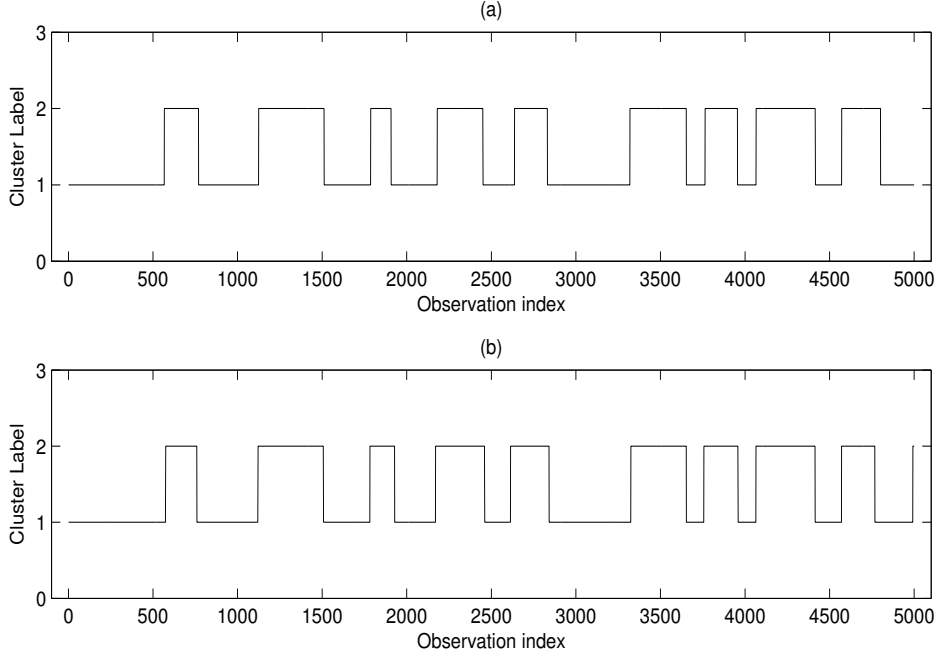


Figure 6.3: (a) True cluster labels versus observation index. (b) Estimated cluster labels versus observation index.

The dataset 'DS-1' comprises a total of 5000 observations generated from two Markov chains that switches between at certain time intervals. In order to achieve a clear illustration, we draw the first 2000 observations and corresponding labellings of each observation in Figure 6.2(a) and Figure 6.2(b) respectively.

We processed the synthetic dataset sequentially for $n = 1$ to $n = 5000$ and report the estimated cluster labellings associated with each observation in Figure 6.3(a). For comparison purposes we also draw the estimated clustering labellings in Figure 6.3(b). The proposed algorithm achieves a high cluster labeling accuracy of 96.68% and the estimated mean transition matrix for each Markov chain is,

$$\hat{\mathbf{T}}_1 = \begin{bmatrix} 0.21 & 0.52 & 0.27 \\ 0.26 & 0.25 & 0.49 \\ 0.52 & 0.23 & 0.25 \end{bmatrix} \quad \hat{\mathbf{T}}_2 = \begin{bmatrix} 0.25 & 0.24 & 0.51 \\ 0.52 & 0.24 & 0.24 \\ 0.23 & 0.50 & 0.27 \end{bmatrix}.$$

The mean error of the estimates of the matrices, $\hat{\mathbf{T}}_1$, $\hat{\mathbf{T}}_2$ are $e_{\hat{\mathbf{T}}_1} = 0.0154$ and $e_{\hat{\mathbf{T}}_2} = 0.0117$ respectively. The test results indicate that the algorithm is able to estimate the number of transition matrix, its parameters, as well as the clustering labellings accurately for the dataset DS-1.

In order to observe online performance of the algorithm, in Figure 6.4 we draw the normalized expected changepoint detection latency calculated for predefined

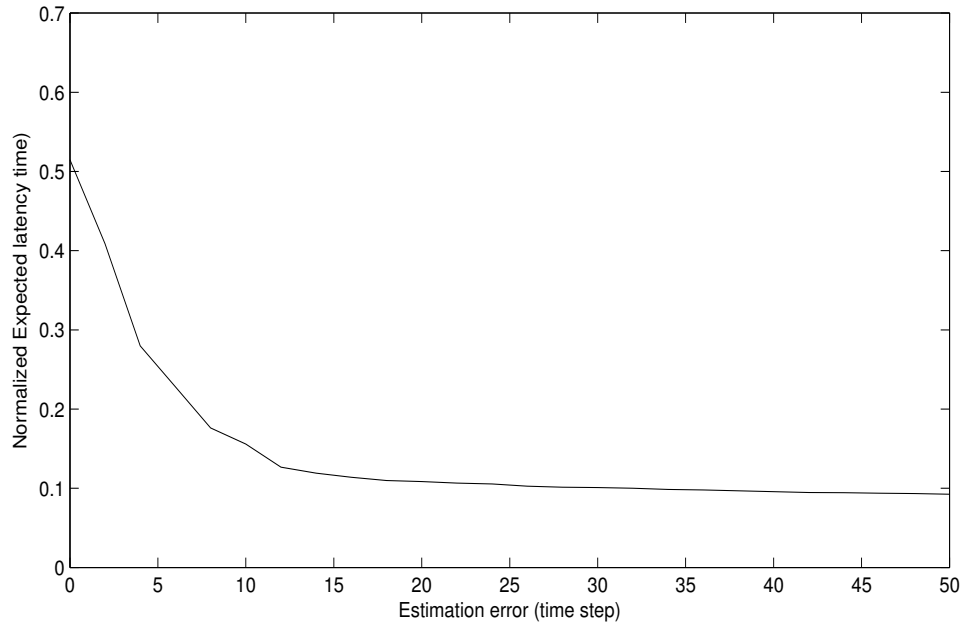


Figure 6.4: Normalized expected latency versus changepoint location estimation error.

estimation error. In our work we define the changepoint detection latency as the time required to estimate a new changepoint within a predefined estimation error. The calculated latency is normalized according to the mean sojourn time of the corresponding dataset for easy interpretation of the results. The proposed algorithm estimates the model posterior distribution sequentially and the clustering performance gradually increases till an adequate amount of data that represents the actual model is received. Therefore, for the dataset DS-1, the first 1000 observations are assumed to be the learning phase of the algorithm and neglected in the calculation of the expected changepoint detection latency.

In Figure 6.4, we observe that the algorithm approximately detects a new changepoint with an expected latency of 0.13 when an accuracy of 12 time steps is required. In other words this means that the algorithm latency in detecting a changepoint with an error lower than 12 time steps is expected to be 13% of the mean sojourn time.

Next, we evaluated the performance of the algorithm for clustering Markov chains that consists of 9 states. The synthetic data 'DS-2' is generated by three Markov chains of which the parameters are switched at certain time intervals. The data and corresponding labelings are illustrated in Figure 6.5(a) and (b) respectively for the first 2000 observations.

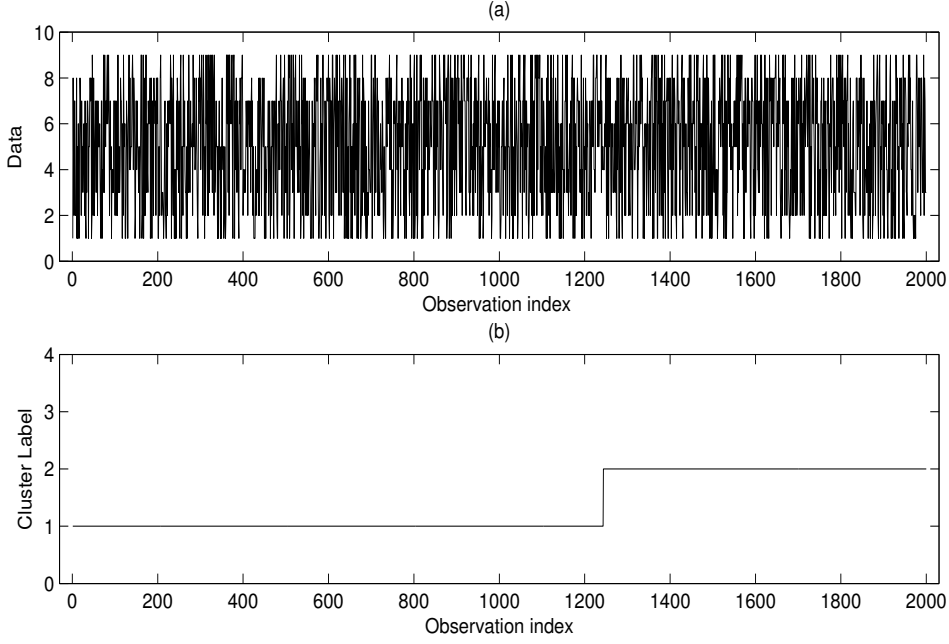


Figure 6.5: (a) Synthetically generated data and (b) associated cluster labels.

The sojourn distribution of the synthetic data is determined by a negative binomial distribution, $NB(r, p)$, parameterized as, $r = 4$ and $p = 0.0003$. The initial values are defined to be equal probable and the transition matrix of each Markov chain is randomly defined as follows,

$$\mathbf{T}_1 = \begin{bmatrix} 0.103 & 0.155 & 0.069 & 0.086 & 0.155 & 0.172 & 0.103 & 0.138 & 0.017 \\ 0.114 & 0.068 & 0.023 & 0.068 & 0.114 & 0.023 & 0.227 & 0.136 & 0.227 \\ 0.088 & 0.140 & 0.053 & 0.123 & 0.088 & 0.070 & 0.158 & 0.105 & 0.175 \\ 0.170 & 0.189 & 0.132 & 0.094 & 0.151 & 0.019 & 0.113 & 0.113 & 0.019 \\ 0.170 & 0.106 & 0.043 & 0.021 & 0.213 & 0.149 & 0.191 & 0.064 & 0.043 \\ 0.093 & 0.185 & 0.167 & 0.093 & 0.111 & 0.019 & 0.148 & 0.167 & 0.019 \\ 0.021 & 0.170 & 0.021 & 0.170 & 0.170 & 0.191 & 0.106 & 0.043 & 0.106 \\ 0.160 & 0.100 & 0.020 & 0.120 & 0.040 & 0.080 & 0.200 & 0.100 & 0.180 \\ 0.115 & 0.033 & 0.148 & 0.164 & 0.049 & 0.148 & 0.131 & 0.066 & 0.148 \end{bmatrix}$$

$$\mathbf{T}_2 = \begin{bmatrix} 0.063 & 0.083 & 0.063 & 0.083 & 0.083 & 0.188 & 0.188 & 0.167 & 0.083 \\ 0.172 & 0.069 & 0.034 & 0.138 & 0.034 & 0.241 & 0.103 & 0.138 & 0.069 \\ 0.043 & 0.170 & 0.191 & 0.021 & 0.064 & 0.085 & 0.191 & 0.021 & 0.213 \\ 0.125 & 0.125 & 0.050 & 0.150 & 0.075 & 0.150 & 0.200 & 0.100 & 0.025 \\ 0.182 & 0.109 & 0.018 & 0.164 & 0.018 & 0.073 & 0.182 & 0.091 & 0.164 \\ 0.109 & 0.109 & 0.130 & 0.043 & 0.174 & 0.022 & 0.130 & 0.152 & 0.130 \\ 0.096 & 0.154 & 0.115 & 0.019 & 0.173 & 0.115 & 0.115 & 0.192 & 0.019 \\ 0.018 & 0.073 & 0.109 & 0.145 & 0.127 & 0.055 & 0.182 & 0.182 & 0.109 \\ 0.020 & 0.180 & 0.020 & 0.180 & 0.080 & 0.180 & 0.180 & 0.020 & 0.140 \end{bmatrix}$$

$$\mathbf{T}_3 = 0.11 \times \mathbf{U}$$

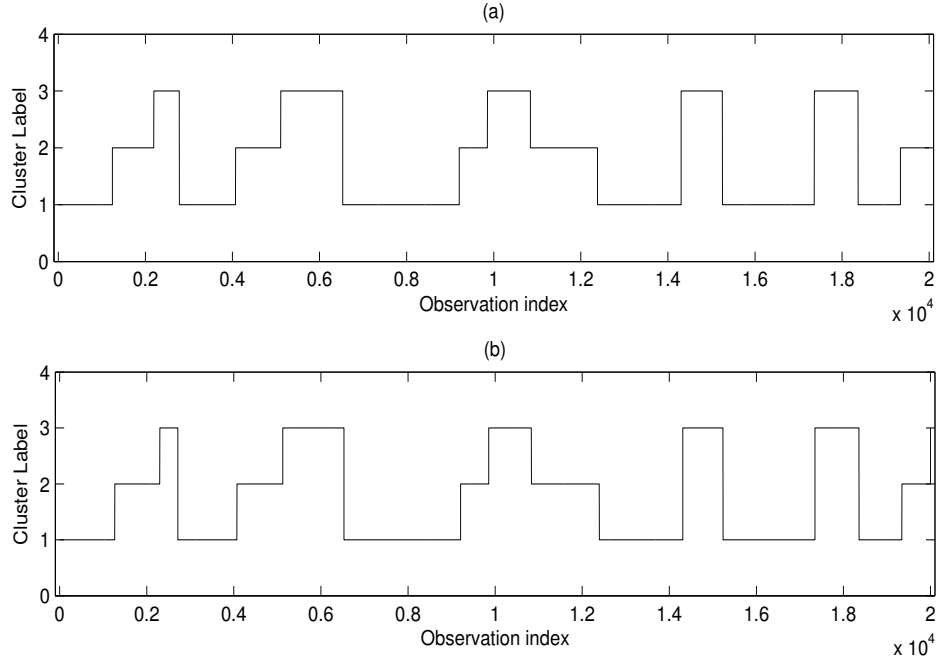


Figure 6.6: (a) True cluster labels versus observation index. (b) Estimated cluster labels versus observation index.

where \mathbf{U} is an all one matrix and \mathbf{T}_3 corresponds to a transition matrix that moves to any state with equal probability.

A total of 20000 observations are processed sequentially and true as well as the estimated labellings associated with each observation are given in Figure 6.6. The proposed algorithm is able to estimate the clustering labellings for the dataset 'DS-2' with a very high accuracy ratio of 0.986%. Note that, a 9 state Markov chain consists of 81 parameters hence the algorithm estimates a total of $k_n \times 81$ parameters in each step of the process. In order to avoid a mess of matrices we just reported the mean estimation error of each transition matrix calculated as $e_{\hat{\mathbf{T}}_1} = 0.0066$, $e_{\hat{\mathbf{T}}_2} = 0.0102$ and $e_{\hat{\mathbf{T}}_3} = 0.0109$. This result shows that the algorithm is able to estimate the transition matrices very close to the actual value.

We also showed the expected latency for the detection of the last changepoint in Figure 6.7. We observed that the algorithm achieves normalized latency values lower than 0.1 when the changepoint estimation error is set to 10 steps. We can conclude that the algorithm is able to detect the change points within a reasonable latency time even for complex dataset where the number of parameters representing the data is above 250.

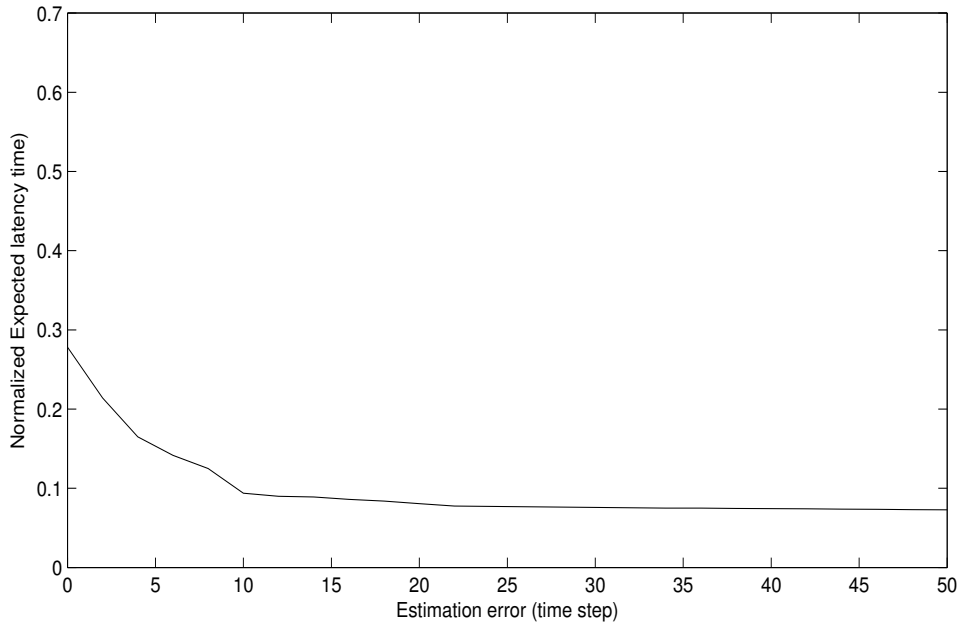


Figure 6.7: Normalized expected latency versus changepoint location estimation error.

6.4.1 Clustering audio signal

In this section we evaluated the performance of the proposed algorithm in a the real world audio signal clustering problem where the audio file comprises of several different instruments are played at overlapping time intervals.

The selected track given in Figure 6.8 (a) is a single channel professional record, sampled at the rate of 44100 kHz and stored in MP3 format. As a preprocessing step, we calculated the Mel-frequency cepstral coefficients (MFCCs) illustrated in Figure 6.8(b) in order to approximate the human auditory system. The window size for MFCC calculation is determined as 0.02 seconds with 0.0045 seconds of overlap. Next, we perform a k-means clustering algorithm to discretize the MFCC coefficients where we excluded the most significant MFCC coefficients and obtain a time series that correspond to a $R = 5$ state Markov chain shown in Figure 6.8(c).

The Dirichlet prior parameters are set to $\kappa_P = 50$, $\kappa_T = 1$ and the parameters of the sojourn distribution, $NB(r, p)$, are determined as $r = 3$ and $p = 0.001$. We sequentially processed a total of 7630 observations that corresponds to a 34.6 seconds of audio signal and report the clustering results obtained at time step $n = 7630$ in Figure 6.9.

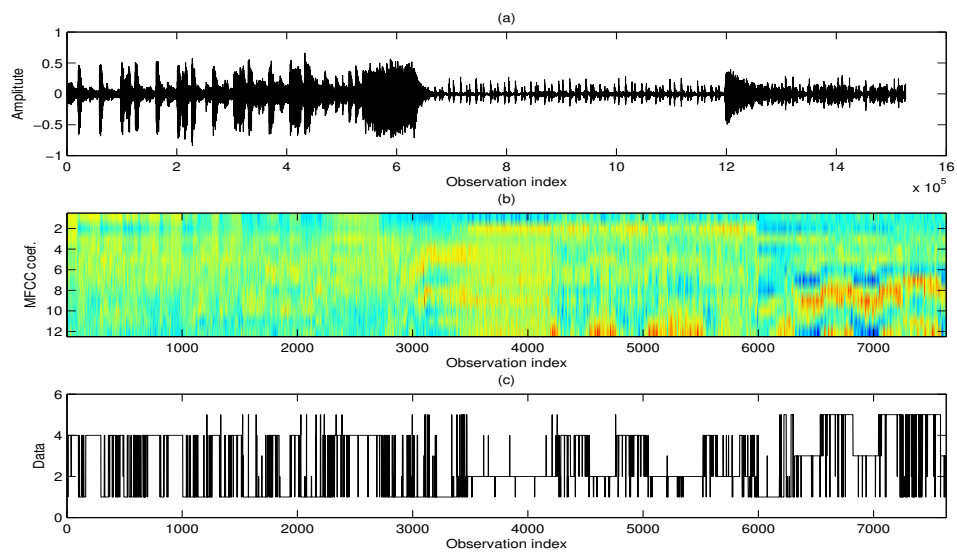


Figure 6.8: (a) 44100 kHz audio signal. (b) Mel-frequency cepstral coefficients (MFCCs). (c) MFCC coefficients digitized by using the k-means algorithm (Input data).

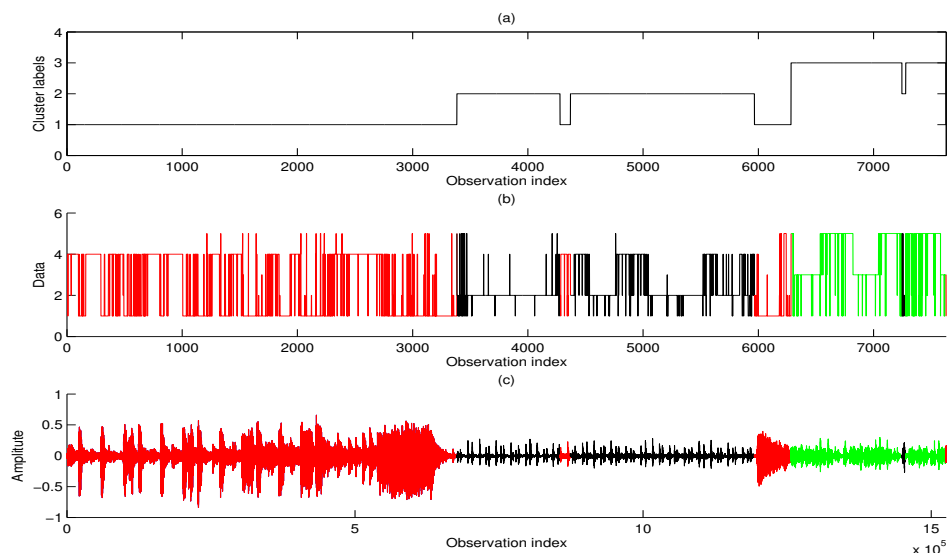


Figure 6.9: (a) Estimated clustering labels versus observation index. (b) Clustered digitized MFCC coefficients (c) Clustered audio signal.

The estimated clustering labels and the arrival times given in Figure 6.9 (a) shows that the audio signal is clustered into three components. We also draw the clustered Markov chains and the corresponding audio signal in Figure 6.9 (b) and (c) by indicating each cluster with different colors labeled as 'red=1', 'black=2' and 'green=3'. By examining the Figure 6.9 (b) and (c) we can conclude that the algorithm is able to cluster the data into three main different regimes and can detect the change points where the regimes switch.

7. CONCLUSION

In this thesis efficient sequential Monte Carlo Samplers for posterior inference in Dirichlet process mixture models are proposed and a new model structure for maneuvering target tracking problem is introduced. The contributions of the thesis can be summarized as follows.

7.1 Summary and Contributions

Following a brief introduction to the sequential Bayesian models and inference in Chapter 2, we focused on variable rate particle filtering particularly for highly maneuvering targets. In Chapter 3, we adapted the multiple model approach to the variable rate particle filtering structure in order to obtain an adaptive algorithm which efficiently tracks the motion mode and thus accurately estimates the target state vector (Ulker et al., 2008; Ulker and Günsel, 2008). The proposed algorithm, MM-VRPF, utilizes a different sojourn and model parameter set for each dynamic mode, resulting in a finer characterization of the maneuvers while preserving the parsimonious state representation. It is shown that the proposed multiple model variable rate structure utilizing a set of dynamic motion models and sojourn parameters enable efficient characterization of the maneuvers as well as the state arrival times compared to the conventional single mode variable rate structure (Ulker et al., 2008; Ulker and Günsel, 2008). In order to avoid the particle degeneracy, we also proposed a regularization scheme for variable rate models. We concluded that if degeneracy is observed, particularly due to the ill defined model parameters, regularization improves the performance of the variable rate models.

In Chapter 4 we proposed a novel sequential Monte Carlo algorithm for the DPM model under the conjugate prior settings (Ulker et al., 2010b). In contrast to the existing sequential importance sampling methods, the local moves are designed to update clustering labels that enable the introduced algorithms to obtain efficient

samples from the time evolving posterior even for large dataset sizes. We showed that previously proposed sequential schemes that apply Gibbs moves to the set of weighted particles are an instance of the SMC samplers algorithm (MacEachern et al., 1999). We evaluated the performance of the conventional particle filter, Gibbs sampler and SMC samplers with different kernel settings, on two different datasets. Test results showed that SMC sampler based methods provide more reliable estimates compared to conventional particle filter and proposed kernels can better represent the modes of the posterior distribution compared to a SMC sampler utilizing Gibbs moves (Ulker et al., 2010b). We concluded that the SMC samplers framework is a competitive alternative to the conventional Gibbs sampler for the DPM models (Ulker et al., 2010b).

In Chapter 5 we improved the SMC sampler proposed in Chapter 4 by using annealing strategies (Ulker et al., 2010a, 2011). The key idea of the method is maintaining an intermediate (annealed) distribution as a surrogate target for the SMC algorithm where resampling is carried out according to this annealed distribution. We use the surrogate density as a proposal to the true target where we can calculate the correct weights without any extra computational cost. Intuitively, we are using the SMC machinery to compute a good proposal density. This strategy enables us to maintain a diverse particle set that seems to be crucial in obtaining an efficient sampler. The test results show that proposed algorithm achieves lower estimation variance and higher log-marginal likelihoods. We also observed that our algorithm is much more efficient compared to conventional methods particularly when DPM target posterior distribution has isolated modes (Ulker et al., 2010a, 2011).

In Chapter 6 we proposed a novel DPM based model for time series clustering under a semi Markovian model structure where the number of clusters and the parameters are unknown. The semi Markovian structure reduces the dimensionality of the model, hence results in a simpler representation. We applied classical particle filtering framework for inference in the proposed model and applied the problem to the Markov Chain clustering problem. We tested the algorithm for synthetic datasets and for clustering a recorded audio data. We observed that the algorithm is able to cluster and estimate the parameters of the synthetic datasets and the audio data successfully. Finally, we envision various applications in hierarchical Bayesian models with a DPM prior. In this thesis, we have concentrated exclusively on the conjugate setting, however

we believe that the actual added benefit of the SMC framework can be realized in the non-conjugate setting where the model parameters need also be sampled.

7.2 Future Work

As future work our objective is to design multi-modal latent variable models for dynamic systems where state arrival times, model order and the parameters are unknown. Our aim is to estimate the unknown model parameters and the model order given the noisy observations. We believe that such models can be easily applied to several important problems in machine learning, signal processing, bioinformatics, pattern recognition and econometrics. The models will be covered under two main sub-topics.

7.2.1 Infinite dimensional hidden semi-Markov models

Conventionally, hidden Markov model assumes that sojourn time between transitions are exponentially distributed. In contrast, a semi-Markov process models the state arrival times as a Markovian process and enable us to model the temporal correlation between the states precisely. However these models usually assume that the model order and the model parameters are priorly known which is not the case in many real world applications.

We aim to construct an infinite dimensional hidden semi-Markov model (inf-HSMM) where a Dirichlet process or a Hierarchical Dirichlet process will be considered as prior for the model proposed (Teh et al., 2006). We believe that such a non-parametric approach will lead to a parsimonious representation under the semi-Markov formalism that will increase the efficiency of the filtering algorithm. Moreover, when transitions between modes are independent from each other, the model will further simplify and Dirichlet process will be adequate as a prior.

7.2.2 Dirichlet process mixtures for non-linear dynamic systems

The Dirichlet Process Mixtures (DPM) have been the key building block particularly in modeling linear dynamic systems with unknown model structure. In order to estimate the noise density with an unknown functional in a linear dynamic system, the DPM model is constructed as a prior over the model Caron et al. (2008). In another work, a

hierarchical DPM (HDPM) model has been introduced by Fox et al. (2007) that accounts for the correlation between the input modes of a switching linear dynamic model however such models can not impose the desired prior over the state arrival times (Fox et al., 2008). The major drawback of the model proposed by Fox et al. (2007) is the tendency of the defined HDPM prior to invent unnecessary modes and sticky HDPM models are proposed in order to solve this problem in (Fox et al., 2008, 2009). The models that rely on fixed rate model structure have serious limitations and are unable to model the nature of the data in real world data. Moreover they cause complicated models and require complex inference schemes. In literature it is shown that a semi-Markovian structure can lead to a parsimonious representation in which simple models and efficient inference mechanisms can be considered (Godsill et al., 2007; Whiteley et al., 2007). Therefore, we aim to propose a semi-Markovian structure for non-linear dynamic models where the number of dynamic motion modes and the parameters will be estimated from the observed data. The main difficulty in such a model is that observations are no longer available as an input to the DPM model, therefore marginalization over the dynamic model is required to estimate the model parameters.

Since exact inference is unavailable for both of the topics addressed above, efficient sequential Monte Carlo schemes are required to approximate the true target posterior. Recently, efficient sampling schemes based on the SMC sampler framework proposed by Del Moral et al. (2006) has been successfully applied for sequential inference in DPM models (Ulker et al., 2010b,a, 2011) and variable rate target tracking (Whiteley et al., 2007). Due to significant improvements achieved in both works, we will consider such methods for statistical inference in the models proposed. We will also investigate particle Markov Chain Monte Carlo (PMCMC) methods which is a recent advance on MCMC based sampling techniques in order to design more efficient inference schemes (Andrieu et al., 2009).

REFERENCES

- Andrieu, C., Doucet, A. and Holenstein, R.** (2009). Particle markov chain monte carlo, *Technical report*, University of British Columbia, Department of Statistics.
- Antoniak, C.** (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems, *Annals of Statistics*, **2**, 1152–1174.
- Arulampalam, M., Mansell, T., Ristic, B. and Gordon, N.** (2004). Bearings-only tracking of maneuvering targets using particle filters, *Eurasip Journal on Applied Signal Processing*, **15**, 2351–2365.
- Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T.** (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, *IEEE Trans. Signal Processing*, **50**, 174–188.
- Bar-Shalom, Li, X. R. and Kirubarajan, T.** (2001). *Estimation with applications to tracking and navigation: Theory, algorithms and software*, New York: Wiley.
- Blair, W. D. and Watson, G. A.** (1992). Interacting multiple model algorithm with aperiodic data, *SPIE Symp. Acquisition, Tracking Pointing*, Orlando, pp. 83–91.
- Blei, D. and Jordan, M.** (2006). Variational inference for dirichlet process mixtures, *Journal of Bayesian Analysis*, **1**, 121–144.
- Blei, D. M. and Jordan, M. I.** (2004). Variational methods for the Dirichlet process,, *Proceedings of the 21st International Conference on Machine Learning*.
- Blom, H. A. P., Bar-Shalom, Y., Blom, H. A. P. and Bar-Shalom, Y.** (1998). The interacting multiple model algorithm for systems with markovian switching coefficients, *IEEE Transactions on Automatic Control*, **33**, 780–783.
- Bloomer, L. and Gray, J. E.** (2002). Are more models better?: The effect of the model transition matrix on the imm filter, *34th Southeastern Symp. System Theory*, Huntsville, pp. 20–25.
- Caron, F., Davy, M., Doucet, A., Duflos, E. and Vanheeghe, P.** (2008). Bayesian inference for linear dynamic models with Dirichlet process mixtures, *IEEE. Trans. Signal Process.*, **56**, 71–84.
- Chen, G.** (1993). *Approximate Kalman filtering*, World Scientific, Singapore.

- Chen, R. and Liu, J. S.** (2000). Mixture Kalman filters, *J. Roy. Stat. Soc. B Stat. Meth.*, **62**, 493–508.
- Chen, Z.** (2003). Bayesian filtering: From kalman filters to particle filters, and beyond, *Technical report*, Adaptive Systems Lab, McMaster University.
- Chopin, N.** (2002). A sequential particle filter for static models, *Biometrika*, **89**, 539–551.
- Do, K.-A., Müller, P. and Tang, F.** (2005). A bayesian mixture model for differential gene expression, *Journal of the Royal Statistical Society Series C*, **54**(3), 627.
- Doucet, A.** (1998). On sequential monte carlo methods for bayesian filtering, *Technical report*, Dept. Eng., Univ. Cambridge, U.K.
- Doucet, A., Briers, M. and Senecal, S.** (2006). Efficient block sampling strategies for sequential Monte Carlo methods, *J. Comput. Graph. Stat.*, **15**, 693–711.
- Doucet, A., de Freitas, . F. G. and Gordon, N. J.** (2001)a. *Sequential Monte Carlo methods in practice*, chapter An introduction to sequential Monte Carlo methods, New York: Springer-Verlag, New York.
- Doucet, A., Gordon, N. J. and Krishnamurthy, V.** (2001)b. Particle filters for state estimation of jump markov linear systems, *IEEE Transactions on Signal Processing*, **49**, 613624.
- Escobar, M. and West, M.** (1992). Computing Bayesian nonparametric hierarchical models, *Technical report*, Duke University, Durham, USA.
- Fearnhead, P.** (2004). Particle filters for mixture models with an unknown number of components, *J. Stat. Comput.*, **14**, 11–21.
- Fox, E., Sudderth, E., Jordan, M. and Willsky, A.** (2008). An hdp-hmm for systems with state persistence, *International Conference on Machine learning*.
- Fox, E., Sudderth, E., Jordan, M. and Willsky, A.** (2009). Nonparametric bayesian identification of jump systems with sparse dependencies, *IFAC Symposium on System Identification*.
- Fox, E., Sudderth, E. and Willsky, A.** (2007). Hierarchical dirichlet processes for tracking maneuvering targets, *Proceedings of the International Conference on Information Fusion*, Quebec, Canada.
- Godsill, S. J. and Vermaak, J.** (2005). Variable rate particle filters for tracking applications, *IEEE Statistical Signal Processing*, 1280–1285.
- Godsill, S. J., Vermaak, J., NG, K. and Li, J.** (2007). Models and algorithms for tracking of manoeuvring objects using variable rate particle filters, *Proceedings of the IEEE*, **95**, 925–952.
- Gordon, N. J., Salmond, D. J. and Smith, A. F. M.** (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation, *EE Proceedings Part F: Radar and Signal Processing*, **140**, 107113.

- Green, P.** (1995). Reversible jump markov chain monte carlo computation and bayesian model determination, *Biometrika*, **82**, 711-732.
- Hanscomb, D. C. and Hammersley, J. M.** (1964). *Monte Carlo methods*, Chapman&Hall, London.
- Jain, S. and Neal, R.** (2000). A split-merge Markov Chain Monte Carlo procedure for the Dirichlet process mixture model, *J. Comput. Graph. Stat.*, **13**, 158–182.
- Jarzynski, C.** (1997). Nonequilibrium equality for free energy differences, *Phys. Rev. Lett.*, **78**, 2690–2693.
- Jilkov, V. P. and Li, X. R.** (2004). Online bayesian estimation of transition probabilities for markovian jump systems, *IEEE Trans. on Signal Processing*, **52**, 1620–1630.
- Kalman, R. E.** (1960). A new approach to linear filtering and prediction problem, *Trans. ASME, Ser. D, J. Basic Eng.*, **82**, 34–45.
- Kirubarajan, T., Bar-Shalom, Y. and Lerro, D.** (2001). Bearings-only tracking of maneuvering targets using a batch-recursive estimator, *IEEE Transactions on Aerospace and Electronic Systems*, **37**, 770780.
- Kolmogorov, A. N.** (1941). Stationary sequences in hilbert spaces, *Bull. Math. Univ. Moscow (in Russian)*, **2**, 40.
- Li, X. R. and Jilkov, V. P.** (2003). Survey of maneuvering target tracking part 1: Dynamic models, *IEEE Trans. Aerosp. Electron. Syst.*, **39**, 1333–1364.
- Li, X. R. and Jilkov, V. P.** (2005). A survey of maneuvering target tracking, part v: Multiple-model methods, *IEEE Transactions on Aerospace and Electronic Systems*, **41**, 1255–1321.
- Liu, J. S.** (1996). Nonparametric hierarchal Bayes via sequential imputations, *Ann. Statist.*, **24**, 910–930.
- Liu, J. S. and Chen, R.** (1998). Sequential monte carlo methods for dynamical systems, *J. Amer. Statist. Assoc.*, **93**, 10321044.
- MacEachern, S. N.** (1994). Estimating normal means with a conjugate style Dirichlet process prior, *Comm. Statist. Simulation Comput.*, **23**, 727–741.
- MacEachern, S. N., Clyde, M. and Liu, J.** (1999). Sequential importance sampling for nonparametric Bayes models, *Can. J. Stat.*, **27**, 251–267.
- Marshall, A.** (1956). The use of multi-stage sampling schemes in monte carlo computations, **Meyer, M.**, ed., *Symposium on Monte Carlo Methods*, New York: Wiley, pp. 123–140,.
- Maskell, S.** (2004). Tracking maneuvering targets and classification of their maneuverability, *EURASIP J. Appl. Signal Process*, **15**, 2339–2350.

- McGinnity, S. and Irwin, G. W.** (2000). Multiple model bootstrap filter for maneuvering target tracking, *IEEE Transactions on Aerospace and Electronic Systems*, **36**, 1006–1012.
- Musso, C., Oudjane, N. and LeGland, F.** (2001). *Sequential Monte Carlo methods in practice*, chapter Improving regularized particle filters, New York, Springer-Verlag.
- Neal, R.** (2000). Markov chain sampling methods for dirichlet process mixture models, *J. Comput. Graph. Statist*, **9**, 249–265.
- Neal, R.** (2001). Annealed importance sampling, *Statist. Comput*, **11**, 125–139.
- Pitt, M. and Shephard, N.** (1999). Filtering via simulation: Auxiliary particle filters, *J. Amer. Statist. Assoc.*, **94**, 590–599.
- Quintana, F. A.** (1996). Nonparametric Bayesian analysis for assessing homogeneity in $k \times l$ contingency tables with fixed right margin totals, *J. Am. Stat. Assoc.*, **93**, 1140–1149.
- Quintana, F. A. and Newton, M. A.** (1998). Computational aspects of nonparametric Bayesian analysis with applications to the modeling of multiple binary sequences, *J. Comput. Graph. Stat.*, **9**, 711–737.
- S. Walker, P. L., P. Damien and Smith, A.** (1999). Bayesian nonparametric inference for random distributions and related functions, *J. R. Statist. Soc. B*, **61**(3), 485–527.
- Sethuraman, J.** (1994). A constructive definition of Dirichlet priors, *Statist. Sin.*, **4**, 639–650.
- Silverman, B. W.** (1986). *Density estimation for statistics and data analysis*, Chapman&Hall, London, U.K.
- Singer, R. A.** (1970). Estimating optimal tracking filter performance for manned maneuvering targets, *Transactions on Aerospace and Electronic Systems*, **6**.
- Song, T. L.** (1996). Observability of target tracking with bearings only measurements, *IEEE Transactions on Aerospace and Electronic Systems*, **32**, 14681472.
- Teh, W., Jordan, M. I., Beal, M. J. and Blei, D. M.** (2006). Hierarchical dirichlet processes, *J. Amer. Stat. Assoc.*, **101**, 15661581.
- Teh, Y. W., Jordan, M. I., Beal, M. J. and Blei, D. M.** (2004). Hierarchical Dirichlet processes, *J. Am. Stat. Assoc.*, **101**, 1566–1581.
- Del Moral, Doucet, A. and Jasra, A.** (2006). Sequential Monte Carlo samplers, *J. Roy. Stat. Soc. B Stat. Meth.*, **63**, 11–436.
- Ulker, Y. and Günsel, B.** (2008). Target tracking with regularized variable rate particle filters, *Signal Processing, Communication and Applications Conference (SIU)*, Mugla, Turkey.

- Ulker, Y. and Gonsel, B.** (2011). Learning emotional speech by using dirichlet process mixtures, *Signal Processing, Communication and Applications Conference (SIU)*.
- Ulker, Y. and Gonsel, B.** (2012). Mutiple model target tracking with variable rate particle filters, *Accepted for publication in Digital Signal Processing, Elsevier*.
- Ulker, Y., Gonsel, B. and Cemgil, A. T.** (2010)a. Annealed SMC samplers for Dirichlet process mixture models, *20th International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey.
- Ulker, Y., Gonsel, B. and Cemgil, A. T.** (2010)b. SMC samplers for Dirichlet process mixtures, *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy.
- Ulker, Y., Gonsel, B. and Cemgil, A. T.** (2011). Annealed smc samplers for nonparametric mixture models, *IEEE Signal Processing Letters*, **18**, 3–6.
- Ulker, Y., Gonsel, B. and Kirbiz, S.** (2008). A multiple model structure for tracking by variable rate particle filters, *19th International Conference on Pattern Recognition*, Tampa, USA.
- Whiteley, N., Johansen, A. M. and Godsill, S.** (2007). Efficient monte carlo filtering for discretely observed jumping processes, *IEEE Statistical Signal Processing Workshop*, Madison, Wisconsin., pp. 89–93.
- Wiener, N.** (1949). *Extrapolation, interpolation and smoothing of time series, with engineering applications*, Wiley, New York.

CIRRICULUM VITAE

Name Surname: Yener Ulker

Place and Date of Birth: 01.02.1978

E-mail: yeneru@yahoo.com

BS.C: Yıldız Technical University

MS.C: Istanbul Technical University

Professional Experience: National Research Institute of Electronics and Cryptology (UEKAE), Gebze, Kocaeli, Turkey, February 2001 to April 2006.

PUBLICATIONS ON THE THESIS

Ulker, Y., Gonsel, B., (2012), Multiple model target tracking with variable rate particle filters, *Accepted for publication in Digital Signal Processing, Elsevier.*

Ulker, Y., Gonsel, B., Cemgil, A. T., (2011), Annealed SMC samplers for nonparametric mixture models, *IEEE Signal Processing Letters*, vol.18, pp.3-6.

Ulker, Y., Gonsel, B., (2011), Learning emotional speech bu using Dirichlet process mixtures, *Signal Processing, Communication and Applications Conference*, Turkey.

Ulker, Y., Gonsel, B., Cemgil, A. T., (2010), Annealed SMC samplers for Dirichlet process mixture models, *20th International Conference on Pattern Recognition*, Istanbul, Turkey.

Ulker, Y., Gonsel, B., Cemgil, A. T., (2010), SMC Samplers for Dirichlet process mixtures, *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy.

Ulker, Y., Gonsel, B., Kirbiz, S., (2008), A multiple model structure for tracking by variable rate particle filters, *19th International Conference on Pattern Recognition (ICPR)*, Tampa, USA.

Ulker, Y., Gonsel, B., (2008), Target tracking with regularized variable rate particle filters, *Signal Processing Communication and Applications Conference (SIU)*, Turkey.

