

BAYESIAN MULTIPLE TARGET TRACKING

by

Praveen Babu Choppala

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Doctor of Philosophy
in Engineering.

Victoria University of Wellington
2014

Abstract

This thesis addresses several challenges in Bayesian target tracking, particularly for *array* signal processing applications, and for *multiple* targets.

The optimal method for multiple target tracking is the Bayes' joint filter that operates by hypothesising all the targets collectively using a joint state. As a consequence, the computational complexity of the filter increases rapidly with the number of targets. The probability hypothesis density and the multi-Bernoulli filters that overcome this complexity do not possess a suitable framework to operate directly on phased sensor array data. Instead, such data is converted into beamformer images in which close targets may not be effectively resolved and much information is lost. This thesis develops a *multiple signal classification (MUSIC) based multi-target particle filter* that improves upon the filters mentioned above. A *MUSIC based multi-Bernoulli particle filter* is also developed, that operates more directly on array data.

The above mentioned particle filters require a resampling step which impedes information accumulation over successive observations, and affects the detection of very covert targets. This thesis develops *soft resampling* and *soft systematic resampling* to overcome this problem without affecting the accuracy of approximation. Additionally, *modified Kolmogorov-Smirnov testing* is proposed, to numerically evaluate the accuracy of the particle filter approximation.

Acknowledgements

Several people have contributed in great and small ways for the successful progress of this thesis. I would like to express my gratitude to them.

It is with immense gratitude and pride that I acknowledge the support and encouragement of my supervisors Dr Paul Teal and Dr Marcus Frean. They have constantly motivated me and contributed a lot in rich technical discussions and proposing new ideas. Their knowledge, enthusiasm and patience has helped me advance further in the field. It is from them I learnt the joy of philosophical learning. It has been a pleasure to partner with them in this quality research and I gladly share the credit of my work with them.

I also thank Dr Ronald Mahler, Lockheed Martin, Dr Mark Morelande, University of Melbourne, Dr Branko Ristic, DSTO Australia and A/Prof Tadesse Ghirmai, Bothell University of Washington who have on more than one occasion answered my questions through email correspondence.

I am grateful to Callaghan Innovation (formerly Industrial Research Limited), Lower Hutt, Wellington, New Zealand, for supporting my study through the scholarship. I am also grateful to the Faculty of Graduate Research (FGR) and the School of Engineering and Computer Science (ECS), Victoria University of Wellington (VUW) and to Dr Paul Teal, for funding the conferences attended during the course of this work.

I thank the FGR, the Faculty of Engineering, the School of ECS, the Information Technology Services and the many other departments at VUW that

smoothed my work on campus. My special thanks to the Library Services and to the subject librarian Rohini Biradavolu for the literary support. I acknowledge the technical fellowship of the faculty members and co-doctoral students at the Communications and Signal Processing (CaSP) group. The group meetings have been very productive and enriching. I also thank my former and current colleagues at VUW — Dr Mohammad Ayat, Dr Siva Dorairaj, Dr Adella Campbell, Satya Aggarwal, Ahmed Sheik Deeb, Ramoni Adeogun, Tanmay Maity and Harsh Tataria — for their friendship and support. I thank Dr Krupa Rao Nakka, Prof. B.S.N. Murthy and Prof. B.V.R Murthy who inspired me in many ways to pursue doctoral studies.

I express my deep gratitude to my beloved parents Mrs & Mr Mohana Rao Choppala for believing in me enough to think this embarkment is worth while. They have been patient, supportive and encouraging in all academic and personal aspects. They were a rock of support especially during the earthquakes in Wellington. Thank you mum and dad - but for your support and forbearance, this thesis could not have been possible.

I am grateful to Lord Jesus Christ for the opportunities and resources He graciously provided to unravel the many scientific mysteries hidden by Him in nature and to more profoundly behold His awesomeness and majesty.

List of Publications

The propositions and results of this dissertation have, or will, appear as published material as follows;

1. P.B. Choppala, P.D. Teal, and M.R. Frean, “Soft resampling for improved information retention in particle filtering,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 13, 2013, pp. 4036–4040.
2. P.B. Choppala, M.R. Frean, and P.D. Teal, “Soft systematic resampling for accurate posterior approximation and increased information retention in particle filtering,” in *Proc. IEEE Workshop Statist. Signal, Process.*, Australia, 2014.
3. P.B. Choppala, P.D. Teal, and M.R. Frean, “Adapting the multi-Bernoulli filter to phased array observations using MUSIC as pseudo-likelihood,” in *Proc., 17th Int. Conf. Inf. Fusion*, Spain, 2014.
4. P.B. Choppala, P.D. Teal, and M.R. Frean, “Multi-target particle filtering using MUSIC, clustering and soft resampling,” in *IEEE Trans. Signal Process.*, 2014 (manuscript under review).
5. P.B. Choppala, P.D. Teal, and M.R. Frean, “On the performance of particle filters in accurately representing the posterior.” (Manuscript in preparation for submission to *Proc., 18th Int. Conf. Inf. Fusion*, Washington, DC, USA.)

Contents

1	Introduction	1
1.1	Organisation of this thesis	2
1.2	Scope of this thesis	2
1.2.1	Sequential Bayesian estimation	3
1.2.2	Genealogy of Bayesian filters	4
1.3	Problem identification	6
1.3.1	MTT for array signal processing	7
1.3.2	Resampling in PF	9
1.4	Contributions of this thesis	10
1.4.1	MTT for array signal processing	11
1.4.2	Resampling in PF	12
1.5	Summary	13
2	Bayesian filtering	15
2.1	Single target state space modelling	16
2.2	Single target Bayesian filtering	17
2.3	Multi-target Bayesian filtering	20
2.3.1	Joint state space modelling	20
2.3.2	The MTT Bayes' filter recursion	21
2.3.3	Implications of the sensor model for the likelihood	23
2.3.4	Curse of dimensionality	25
2.4	The Kalman filter	26
2.4.1	The Kalman filter recursion	27

2.4.2	Historical progress of the Kalman filter	28
2.5	The sequential Monte Carlo	30
2.5.1	Monte Carlo sampling	31
2.5.2	Importance sampling	33
2.5.3	Sequential importance sampling	35
2.5.4	Sequential importance resampling: The PF	37
2.5.5	The MTT joint PF	44
2.5.6	Choice of importance distribution	46
2.5.7	Track-before-detect PF	47
2.5.8	Historical progress of the PF	49
2.6	The probability hypothesis density filter	52
2.6.1	The RFS model	53
2.6.2	The PHD	55
2.6.3	The PHD filter recursion	58
2.6.4	Historical progress of the PHD filter	60
2.7	The multi-Bernoulli filter	64
2.7.1	The multi-Bernoulli approximation	65
2.7.2	The image observation model	66
2.7.3	The MeMBer filter recursion	69
2.7.4	Historical progress of the MeMBer filter	73
2.8	Summary	75
3	The MUSIC based MTT PF	77
3.1	Related work	79
3.2	Motivation	81
3.3	Phased sensor array model	81
3.4	The single target PF revisited	83
3.5	Multiple signal classification	84
3.5.1	Classical MUSIC	84
3.5.2	MUSIC as a pseudo-likelihood in the PF	88
3.6	Regulated clustering	91
3.6.1	k-means clustering	91

3.6.2	Soft resampling	93
3.6.3	Regulation of the clusters	94
3.7	Simulation results	97
3.8	Conclusion and summary	109
4	The MUSIC based MeMBer filter	111
4.1	Related work	112
4.2	Motivation	113
4.3	Phased sensor array model	114
4.4	The particle-MeMBer filter for image data	115
4.5	The MUSIC based MeMBer filter	118
4.6	Simulation results	121
4.6.1	Evaluation of MUSIC based MTT filters	132
4.7	Conclusion and summary	144
5	Soft resampling	147
5.1	Related work	149
5.2	Motivation	151
5.3	Soft resampling	152
5.4	Soft systematic resampling	155
5.4.1	Motivation for soft systematic resampling	155
5.4.2	Soft systematic resampling algorithm	156
5.4.3	Analysis of properties of the soft resampler	159
5.5	The numerical measure of PF performance	161
5.5.1	The modified KS testing approach	163
5.6	Evaluation	166
5.6.1	The proposal analysed using a TBD example	167
5.6.2	The proposal compared with a Kalman filter	175
5.6.3	Offline implementation on real data	180
5.7	Conclusion and summary	186

6 Conclusion and future extensions	187
6.1 Contributions of this thesis	188
6.1.1 The MUSIC based MTT PF	188
6.1.2 The MUSIC based MeMBer filter	189
6.1.3 Soft resampling	190
6.1.4 The KS statistic	191
6.2 Future extensions of this thesis	192
6.2.1 Bayesian filtering for array processing	192
6.2.2 ESPRIT based measurement feed for PHD filter	194
6.3 Conclusion	196

1

Introduction

Bayesian multiple target tracking (MTT) is an important tool for solving many problems in science and engineering. Examples include the tracking of aircraft from RADAR [1], fish from SONAR [2], space debris from telescopic images [3], heart-beat from ECG [4], etc. The field is attracting much attention from the research community, but there remain significant limitations. This thesis adds to the body of knowledge in Bayesian target tracking by proposing several novel techniques that overcome many of the problems of state-of-the-art tracking systems.

In this chapter, section 1.1 outlines the organisation of this thesis. Section 1.2 presents the scope of this thesis: the sequential Bayesian estimation is briefly presented and the emergence of various Bayes' filters is discussed. The problems with state-of-the-art Bayesian tracking systems that are of interest to this thesis are identified in section 1.3 and the thesis contributions are outlined in section 1.4.

1.1 Organisation of this thesis

This dissertation first gives a prologue on sequential Bayesian estimation and outlines state-of-the-art methods in modern Bayesian tracking systems. With emphasis on array signal processing and particle filter methods, it then discusses how the contemporary tracking systems are limited in many ways, and finally presents the contributions made to overcome these limitations.

This chapter presents the scope of this thesis, the focus areas, the challenges of interest and the contributions (made by this thesis) to overcome these challenges. Chapter 2 sets the notation and presents the mathematical framework of state-of-the-art Bayesian tracking systems and their developments. Thereafter, this thesis focusses on its contributions; Chapters 3 and 4 relate to the primary research focus of this thesis — MTT for array signal processing applications. Each of these chapters presents the drawbacks of contemporary Bayesian MTT systems and proposes methods to overcome them. Chapter 5 relates to a prominent Bayesian tracking scheme — the particle filter. This chapter presents a resampling method to overcome the drawbacks of the particle filter in terms of its information retention ability over time. The chapter also proposes the use of Kolmogorov-Smirnov statistic as a measure to test the particle filter performance. The conclusions and future scope of the material are then discussed in Chapter 6.

1.2 Scope of this thesis

This section draws attention to the basic principles of MTT and Bayesian estimation [5]. The various Bayesian tracking techniques which this thesis investigates are then laid out. Much scientific research depends on the estimation of the *state* of a system of interest. This state captures all the information that characterises the system. For target tracking applications, which is the primary interest of this dissertation, the information contained

in the state(s) of the target(s) relates to the dynamics of the target(s), for example, location, velocity, acceleration, etc. This state information is numerically collected in a vector. The aim of target tracking then, is to *estimate* (or *infer*) this state(s) sequentially with time. Example tracking scenarios include guidance of a missile, surveillance of the ground from air, estimating vehicle trajectory [6], handset tracking [7], tracking respiratory phenomena [8], estimating blood pressure [9], etc, as these applications involve tracking (or time-based estimation) of the state of some parameter(s) (i.e., target(s)) under observation, that evolves over time. The time-varying state of the moving target is estimated from its effects on a data stream, usually the sensor *observations* (or *measurements* or *data*). These observations reported by sensors, such as RADAR, SONAR, cameras, telescopes, etc., provide online information (or evidence) relating to the target(s) of interest, the background noise, and the internal sensor noise (also called thermal noise). The estimation of target(s) from sensor observations require some form of *tracking system* [10, 11], also referred to as a *filter*, the role of which is to estimate on an ongoing basis the state of the target(s). The fact that successive sensor observations are available means that *Bayesian estimation tools* [12] can be used to interpret the current state using the previous information as the *prior*. Bayesian estimation [13] utilises Bayes' rule [14] to recursively build a probability distribution function (pdf) [15] of the state(s) of the target(s). In this thesis, a pdf $p_{\mathbf{x}}(\mathbf{x})$ of a random variable \mathbf{x} is abbreviated as $p(\mathbf{x})$.

1.2.1 Sequential Bayesian estimation

Bayes' rule [14, 16, 17], named after Thomas Bayes¹, gives the degree of belief in a quantity A conditioned on the knowledge of another quantity B . Consider A to be the hypothesis of the target state and let $P(A)$ be its probability. $P(A)$ is also called the *a priori* (or *prior*) probability since it is the initial belief in the hypothesis A without further evidence. Once a target

¹Although proposed in 1763 by the British researcher Thomas Bayes, Bayesian estimation theory was popularised by the French mathematician Pierre-Simon de Laplace.

appears, the sensor(s) starts providing sequential evidence B of the presence and motion of the target. B may either confirm or negate the hypothesis A . $P(B|A)$ is the *conditional probability* (or *likelihood*) that specifies how probable the observed evidence B is supposing that the chosen hypothesis A is true. Then the final belief in the chosen hypothesis A can be updated via Bayes' rule. This belief is also called the *a posteriori* (or *posterior*) probability that the hypothesis A is true conditioned on the observation B . This is expressed as

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)} \quad (1.1)$$

where

$$P(B) = P(B|A) P(A) + P(B|A') P(A') \quad (1.2)$$

and A' is the proposition that A is false. $P(B)$ is the probability of obtaining B averaged over the prior. As new evidence is received at each time sample, Bayes' rule can be used to recursively update the belief in the hypothesis of the target state A . This is called sequential Bayesian estimation. Bayes' rule applies for probability densities also [18]. The recursive operation of Bayes' rule can be seen in [19].

1.2.2 Genealogy of Bayesian filters

Over the years, several Bayesian filters have been invented to deal with a wide range of tracking problems. These Bayesian filters are listed below (numerous variants of these filters are not listed here);

1. The Kalman filter,
2. The particle filter,
3. The probability hypothesis density filter,
4. The multi-Bernoulli filter.

The Kalman filter

The Kalman filter [20], proposed by R.E. Kalman in 1960, is one of the first and well known tracking filters used as a Bayes' recursive solution. It provides a analytical solution to estimate the target state in a way that minimises the error associated with the choice of the target state hypothesis. Therefore, the filter is theoretically optimal. However, the Kalman filter is applicable only for scenarios in which; a) the motion of the target is modelled to be linear, i.e., the mathematical model that describes the temporal change of values in the state vector is linear, b) the sensors capture the target state information using a linear model, i.e., the mathematical model that translates the target state vector to the observation vector is linear, and c) the posterior pdf is Gaussian in nature. These assumptions are unrealistic for many real-world scenarios. Nonetheless, the Kalman filter continues to dominate Bayesian estimation for applications that approximate linearity and Gaussianity.

The particle filter

For non-linear and non-Gaussian filtering, the *particle filter* (PF) [21], also called the sequential Monte Carlo (SMC) filter, formally developed in 1993² by Gordon et al., provides a rigorous Bayesian framework for target state estimation. The PF operates on the principle of approximating the posterior pdf by a set of samples, which in this context are known as *particles*.

The probability hypothesis density filter

The probability hypothesis density (PHD) filter [26], developed by R.P.S. Mahler in 2003³ for multiple targets, operates by approximating the posterior pdf by its first moment. The filter was primarily developed to overcome

²Initial developments leading to Monte Carlo methods appeared in the 1940s [22, 23] and 1970s [24, 25].

³Initial attempts were made as early as 1986 by Mori et al. in [27] and the theory was formally introduced in 1997 jointly by I.R. Goodman and R.P.S. Mahler [28, 29].

drawbacks of the conventionally optimal Bayesian MTT approach, namely the *Bayesian joint filter* [30].

The multi-Bernoulli filter

The multi-target multi-Bernoulli (MeMBer) filter [31], developed by Vo et al. in 2009⁴, is built on the same theory that drives the PHD filter — random finite set (RFS) theory [34] — and operates by approximating the posterior by a set of multi-Bernoulli parameters. The filters other than the Kalman filter are *sub-optimal* and hence do not provide an exact solution.

The contemporary filters mentioned above have been researched in detail and used in many applications (the MeMBer filter is relatively new and is still at the research stage). However, the filters are not without impediments and the field of MTT still presents many challenges that limit the generic use of these filters. This thesis builds on these state-of-the-art methods and provides original and innovative solutions to the impediments suffered by particle, PHD and MeMBer filters.

1.3 Problem identification

The primary interest of this dissertation is in Bayesian MTT [35, 36] for array signal processing in a PF framework. This research provided further motivation to address several challenges in the resampling step of the PF. Therefore, this dissertation can be regarded as dealing with two fields of Bayesian target tracking; the first — MTT for array signal processing, the problems of which are outlined in section 1.3.1, and the second — PF resampling, the problems of which are outlined in section 1.3.2.

⁴The MeMBer filter which was initially developed in 2007 by R.P.S Mahler [32] had an incorrect Taylor linearisation that created a bias in the target number estimate. This was corrected by Vo et al. The preliminary results for the corrected MeMBer filter appeared in 2007 in [33] and the complete solution was more formally published in 2009 in [31]. Hence hereafter, the term “MeMBer filter” refers to the corrected approach in [31].

1.3.1 MTT for array signal processing

This thesis firstly focusses on Bayesian filtering for array signal processing. In array processing [37], the information about multiple moving targets is collected by sampling a wavefield using a *phased array of sensors* that are arranged according to a known geometry. These sensors detect signals generated by or reflected from the targets [38] at discrete time intervals. A phased sensor array means that each sensor of the array is sensitive to the phase of the signals impinging on it. The goal of Bayesian filtering then is the sequential estimation of position (in the near-field case) [39] or direction (in the far-field case) [40] of the target(s) from this sensor array. Some Bayesian MTT applications for array processing [41] include SONAR for tracking a school of fish, or mapping objects on the sea-bed, or detecting a sunken ship, and RADAR for tracking a guided missile, aircraft, or a flock of birds, and microphones for acoustic tracking of talkers for speech enhancement or source separation.

Problem 1 – Joint filter is computationally expensive

If the number of targets is known, the Bayesian joint filter [42] approach is strictly the correct way of casting the MTT problem. The filter operates on joint target state hypotheses [43], each of which is a concatenation of single target hypotheses, i.e., each target state under consideration is a joint hypothesis concerning the states of *all* the actual targets present [44]. Hence the posterior is a *joint pdf*. Although theoretically optimal, the computational complexity of the joint filter *worsens exponentially* with the number of targets [45–47] thereby impeding its use in tracking large numbers of targets. A second difficulty is that frequently, the number of targets is unknown and must itself be modelled as a random variable. Furthermore, the approach suffers from the *data association problem* — the difficulty in relating the sensor point target detections to individual targets [43, 44, 48] in the joint target state.

Problem 2 – RFS filters cannot operate directly on array data

The advent of the RFS filters — the PHD and the MeMBer filters — started to change the Bayesian approach towards MTT problems. Unlike the joint filter, these filters *operate in the dimensionality of a single target*, i.e., they approximate the joint posterior by forming a *multi-modal function* with peaks at locations of the targets. The result is that complexity does not increase exponentially with the number of targets (it increases only polynomially) and data association is not required. The major challenge in RFS filters is that the observation feed to the filters can *only* be a finite set of point target detections (in the PHD filter [32] and the MeMBer filter [31]) or an image (in the MeMBer filter [49]), thereby *restricting their direct use of phased array data*. This problem is usually bypassed by first converting the signal impinging on the sensor array into an image [50] and then pre-processing the image to obtain the observation feed. This two-stage conversion causes substantial loss of the information contained in the array data, and could be detrimental for real time applications [51] in high noise conditions. Additionally, the use of image models makes it difficult to resolve close targets, i.e., two close targets may be merged inextricably and treated as one.

Research goals:

In view of the above identified problems in Bayesian filtering for array processing, the research goals of this thesis include;

1. To overcome problems 1 and 2 — develop a MTT PF that overcomes; a) the computational complexity of the joint filter, and b) the information and resolution loss of the RFS filters,
2. To overcome problem 2 — investigate the possibility (and develop a method) of operating the PHD or the MeMBer filters more directly on phased sensor array signals.

1.3.2 Resampling in PF

The developments made to accomplish the research goals in section 1.3.1 provided further inspiration to address several limitations of PF operation. Since its inception two decades ago, the PF [21] has gained great prominence and is of key interest for this thesis. The main principle of the PF is to recursively generate a set of weighted particles that represent the posterior pdf of the target state. This weighted particle set is obtained in two steps; the first, *sequential importance sampling (SIS)* specifies the process of drawing new particles from the previous ones, and updating their weights according to the likelihood of the state they represent. By itself, SIS results in a large variance of the weights and this inefficiency is known as *degeneracy*. This is overcome using a second step, *resampling* [52], that eliminates those particles having low weights and replaces them by copies of other particles having large weights. Consequently, the PF can be termed the *sequential importance sampling resampling (SISR)* filter [53, 54].

Problem 3 – Resampling cannot retain much information over time

Resampling inevitably results in loss of the information contained in those particles having low weights. These low weights may contain potentially useful target information; for example, the particle could be slowly gaining weight while detecting a very covert target⁵. Therefore, conventional resamplers impede the direct use of a PF to detect and track covert targets. To overcome this, the track-before-detect (TBD) PF [55] was developed by Salmond and Birch in 2001, in which the number of particles is crucial to detection of the appearance and disappearance of covert targets. The greater the number of particles, the higher the filter’s accuracy. The TBD PF implemented in chapter 11 of [56] used as many as 80,000 particles to effectively

⁵A covert target is one whose signal component is well hidden under noise, so one has to track the target for some time before declaring its presence.

track a target from a 20×20 staring camera⁶ image. However, the use of more particles demands increased use of computational resources.

Problem 4 – A measure of resampler performance is required

The true posterior is the Bayes' posterior pdf and it captures *completely* the information about the state(s) of the target(s) and the uncertainty associated with its estimate. It is known that the PF provides only an approximate solution to the Bayes' posterior. It is therefore important to assess the faithfulness of the PF in approximating the posterior pdf. It is predominantly the resampling step that generates inaccuracies in PF approximation to the posterior. This is because resampling involves rejection and replacement of particles based on their weights and causes major adjustments to the information contained in the particle set approximation. Hence, a numerical measure of the resampler performance is required. This has not been adequately explored.

Research goals:

In view of the above identified problems in PF resampling, the research goals of this thesis include;

3. To overcome problem 3 — develop a resampling scheme that retains most of the information contained in low weight particles without affecting the accuracy of approximation,
4. To overcome problem 4 — propose a method to quantify the reliability of the PF in its representation of the posterior.

1.4 Contributions of this thesis

The problems of interest identified in section 1.3 are the challenges this thesis aims to overcome. The contributions of this thesis are summarised hereunder.

⁶A staring camera is a fixed camera that images a pre-determined field of view.

der: section 1.4.1 presents the proposed solutions for the problems identified in section 1.3.1 and section 1.4.2 presents the proposed solutions for the problems identified in section 1.3.2.

1.4.1 MTT for array signal processing

Computation efficient MTT PF for array data:

Chapter 3 develops a *computation efficient multi-target PF* for array processing. The filter uses; (*i*) the well known multiple signal classification (MUSIC) [57] algorithm to evaluate a proxy for the likelihood, (*ii*) clustering [58–60] to separate multiple targets, and (*iii*) a modified form of soft resampling [61] to sustain the detection of weak targets (those that generate weak signals). The motivation for this development comes from problems 1 and 2 of section 1.3.1 that the joint filter approach suffers from high computational complexity and the RFS filters that overcome this complexity cannot operate directly on array data. The developed filter functions in the space of a single target and makes a more direct use of the array data. Hence the filter exhibits higher track accuracy and lower computational complexity than that of the RFS filters. Apart from providing easy means to detect appearing and disappearing targets, the filter also effectively resolves close targets.

MeMBer filter using MUSIC as pseudo-likelihood:

Chapter 4 develops a *MeMBer filter for phased sensor array data*. The motivation for this development comes from problem 2 of section 1.3.1 that the PHD and MeMBer filters do not possess a suitable framework to explicitly operate on phased array sensor signals. This thesis overcomes this problem in the MeMBer filter by virtue of using MUSIC as a proxy to the likelihood. The developed technique allows the MeMBer filter to operate more directly on the array data and hence limit the information loss incurred in the PHD filter and the originally proposed MeMBer filter. Moreover, close targets are effectively resolved.

1.4.2 Resampling in PF

Soft resampling:

Chapter 5 develops *soft resampling*, a scheme that retains more information (over time) contained in low weight particles. The motivation for this development comes from problem 3 of section 1.3.2 that state-of-the-art resamplers cannot accumulate information over long periods and hence cannot be used in a standard PF to detect very covert targets. The developed soft resampler preserves low weights and this aids in the easy detection of covert targets using a standard PF. The technique potentially avoids the need for the contemporary TBD approach that requires an excessive number of particles for effective results.

This thesis also develops *soft systematic resampling*, a modification and correction to the soft resampler. It is understood from problem 4 of section 1.3.2 that it is important to ensure that resampling does not adversely affect the accuracy of the posterior approximation. Soft systematic resampling improves the soft resampler such that the PF approximation to the posterior is now more accurate while at the same time preserving the low weights. This aids in locking fast when targets make sharp and abrupt manoeuvres, or when the model for the target dynamics is chosen incorrectly.

Kolmogorov-Smirnov statistic:

Chapter 5 discovers the potential of Kolmogorov-Smirnov (KS) [62] statistic in numerically testing the faithfulness of the PF in representing the true posterior. On those lines, this thesis develops a *modified KS testing approach* to evaluate the mismatch between the PF and the theoretically optimal Kalman filter for linear Gaussian models. The motivation for this development comes from problem 4 of section 1.3.2 that the issue of numerically measuring the PF resampling performance has not adequately been explored despite being important.

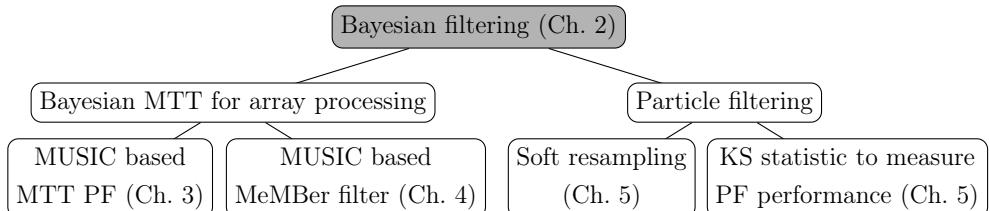
1.5 Summary

This chapter acts as a forerunner to the dissertation. Firstly, the organisation of this thesis was described. Secondly, the principles of sequential Bayesian estimation and the emergence of the various Bayes' filters were discussed. Thirdly, the problems in state-of-the-art methods, which are of interest to this thesis were identified and the research goals were put forward. Finally, the contributions of this thesis to the body of knowledge, that overcome the identified problems were summarised.

1.5. SUMMARY

2

Bayesian filtering



Filtering, in the context of stochastic signal processing [63], aims to derive a *best possible estimate* for the *true state* of a system (or *target*, in the domain of this thesis) using noisy observations about that system. In other words, it is a mathematical operation that involves *extracting* information about the target(s) at time k using the sensor observations received until time k . The target is generally *hidden*, i.e., one does not have exact knowledge about its state. Bayesian filtering [41] facilitates probabilistically modelling this *uncertainty* [64] by using the prior knowledge and sensor evidence about the system. In Bayesian filtering, one models the system states as a random variable rather than as a deterministic random variable.

In this chapter, the Bayesian filtering process and state-of-the-art Bayes' filters are introduced. Section 2.1 introduces the state space approach to modelling the time series of a dynamic system. This is followed by the sequential Bayes' filter formulation in sections 2.2 and 2.3. The Bayes' approximation techniques, namely the Kalman filter, the sequential Monte Carlo approach, the probability hypothesis density filter and the multi-Bernoulli filter are then discussed in sections 2.4, 2.5, 2.6 and 2.7 respectively.

2.1 Single target state space modelling

The state space approach to modelling a moving target involves the *state* of the target. This state encapsulates the *complete* information about the target in its space. This is a numerical quantity represented mathematically as a *vector*. The state space \mathfrak{X} of a single dynamic target [65, 66] is defined to be the set of all values the target state can take. Consider \mathbf{x}_k to be the target state vector at time index k . This state vector, treated as a target hypothesis is modelled as a random variable. For example, for a moving target in the x - y plane, \mathbf{x}_k may be expressed in a 4D space as

$$\mathbf{x}_k = [x, y, v_x, v_y]^T \quad (2.1)$$

where x and v_x are the position and velocity along the x -axis, y and v_y are the position and velocity along the y -axis, and $(\cdot)^T$ denotes matrix transpose. The time index $k \in \mathbb{N}$, i.e., even though the real system may be continuous in time, it is modelled by a discrete time system.

As the target manoeuvres, the state \mathbf{x}_k evolves with time. This dynamic nature of \mathbf{x}_k can be expressed as a first order discrete time probabilistic model as

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, q_{k-1}) \quad (2.2)$$

(2.2) is called the *process*¹ model and is regarded as a *hidden Markov process*. $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_x}$ is a known (possibly non-linear) function that converts

¹The process model in (2.2) is also called *motion* or *system* or *state evolution* model.

\mathbf{x}_{k-1} to \mathbf{x}_k . q_{k-1} is termed the process noise sequence and accommodates modelling errors, if any, in the state evolution model f_k . n_x, n_q are the dimensions of the state and process noise vectors.

The observation space is the set \mathcal{Z} of all possible sensor observations. At time k , the sensor will collect a noisy observation $z_k \in \mathcal{Z}$ about the target \mathbf{x}_k . This observation model is written as

$$z_k = h_k(\mathbf{x}_k, r_k) \quad (2.3)$$

where h_k is a (possibly non-linear) function that translates \mathbf{x}_k from the state space to the observation space. r_k is the observation noise sequence that accommodates sensor imperfections. The noise sequences q_{k-1} and r_k are assumed to be mutually independent and identically distributed (i.i.d), and also independent of \mathbf{x}_k and z_k respectively.

Target inference is achieved by sequentially estimating the state \mathbf{x}_k (or its probability distribution function (pdf)) of a target at time k using all the sensor data $z_{1:k}$ received up to the k th time step. The availability of *prior* information about the target state $\mathbf{x}_{1:k-1}$ (gained from previous sensor data $z_{1:k-1}$) and the sensor evidence z_k allows the use of sequential Bayes' filtering [12] to obtain an improved estimate of the actual target state (also called “ground truth”) at the next time step.

2.2 Single target Bayesian filtering

Bayesian filtering [10, 32] provides a rigorous mathematical framework to recursively *estimate* belief in the target state \mathbf{x}_k at time k using $z_{1:k}$. This is accomplished by forming a functional measure of the belief in \mathbf{x}_k conditioned on $z_{1:k}$ — this functional measure is formally called the Bayes' *posterior* pdf and denoted as $p(\mathbf{x}_k|z_{1:k})$. This posterior pdf captures *all* the available information in the target state and hence is regarded as the optimal solution to the problem of target state estimation.

If the posterior pdf $p(\mathbf{x}_{k-1}|z_{1:k-1})$ at time $k-1$ is known, Bayes' filtering uses a succession of two steps; a) prediction, and b) update, to obtain $p(\mathbf{x}_k|z_{1:k})$. The prediction step uses (2.2) to obtain a probable target state distribution $p(\mathbf{x}_k|z_{1:k-1})$ at the next time step k . This is derived by marginalising over all the previous states. Hence the Bayes' prediction $p(\mathbf{x}_k|z_{1:k-1})$ can be decomposed into a Chapman-Kolmogorov equation [64, 67] as

$$\begin{aligned} p(\mathbf{x}_k|z_{1:k-1}) &= \int p(\mathbf{x}_{1:k}|z_{1:k-1}) d\mathbf{x}_{1:k-1} \\ &\stackrel{(a)}{=} \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|z_{1:k-1}) d\mathbf{x}_{k-1} \\ &\stackrel{(b)}{=} \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, z_{1:k-1}) p(\mathbf{x}_{k-1}|z_{1:k-1}) d\mathbf{x}_{k-1} \\ &\stackrel{(c)}{=} \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|z_{1:k-1}) d\mathbf{x}_{k-1} \end{aligned} \quad (2.4)$$

where (a) follows because \mathbf{x}_k is independent of $\mathbf{x}_{1:k-2}|\mathbf{x}_{k-1}$ which we represent by $\mathbf{x}_k \perp \mathbf{x}_{1:k-2}|\mathbf{x}_{k-1}$. (b) is derived from the probability rule $P(A, B) = P(A|B)P(B)$ where $P(\cdot)$ represents the probability value. (c) follows because $\mathbf{x}_k \perp \mathbf{z}_{1:k-1}|\mathbf{x}_{k-1}$. The term $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ in (2.4) is the distribution that describes the Markov process in (2.2) and is hence called the *Markov transition prior* distribution.

At time k , the new sensor evidence z_k becomes available and Bayes' rule [16, 17] is used to update the prediction in (2.4) and construct the posterior

$p(\mathbf{x}_k|z_{1:k})$. This is conducted using

$$\begin{aligned}
 p(\mathbf{x}_k|z_{1:k}) &\stackrel{(a)}{=} \frac{p(z_{1:k}|\mathbf{x}_k)p(\mathbf{x}_k)}{p(z_{1:k})} \\
 &= \frac{p(z_k, z_{1:k-1}|\mathbf{x}_k)p(\mathbf{x}_k)}{p(z_k, z_{1:k-1})} \\
 &\stackrel{(b)}{=} \frac{p(z_k|z_{1:k-1}, \mathbf{x}_k)p(z_{1:k-1}|\mathbf{x}_k)p(\mathbf{x}_k)}{p(z_k|z_{1:k-1})p(z_{1:k-1})} \\
 &\stackrel{(c)}{=} \frac{p(z_k|z_{1:k-1}, \mathbf{x}_k)p(\mathbf{x}_k|z_{1:k-1})p(z_{1:k-1})p(\mathbf{x}_k)}{p(z_k|z_{1:k-1})p(z_{1:k-1})p(\mathbf{x}_k)} \\
 &= \frac{p(z_k|z_{1:k-1}, \mathbf{x}_k)p(\mathbf{x}_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
 &\stackrel{(d)}{=} \frac{p(z_k|\mathbf{x}_k)p(\mathbf{x}_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}
 \end{aligned} \tag{2.5}$$

In this derivation, (a) is written using the conditional probability rule $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$. (b) uses the rule $P(A, B) = P(A|B)P(B)$ on the term $p(z_k, z_{1:k-1}|\mathbf{x}_k)$. (c) uses the aforementioned conditional probability rule on the term $p(z_{1:k-1}|\mathbf{x}_k)$. (d) follows because $z_k \perp\!\!\!\perp z_{1:k-1}|\mathbf{x}_k$.

In (2.5), $p(z_k|\mathbf{x}_k)$ is the *likelihood* function and $p(z_k|z_{1:k-1})$ is a normalising constant. $p(z_k|z_{1:k-1})$ can be evaluated by marginalising over \mathbf{x}_k as

$$p(z_k|z_{1:k-1}) = \int p(z_k|\mathbf{x}_k, z_{1:k-1})p(\mathbf{x}_k|z_{1:k-1})d\mathbf{x}_k \tag{2.6}$$

$$\stackrel{(a)}{=} \int p(z_k|\mathbf{x}_k)p(\mathbf{x}_k|z_{1:k-1})d\mathbf{x}_k \tag{2.7}$$

where (a) follows because $z_k \perp\!\!\!\perp z_{1:k-1}|\mathbf{x}_k$.

The recurrence of (2.4) and (2.5) is the property that forms the basis for the Bayes' filter. The state estimate $\hat{\mathbf{x}}_k$ can then be extracted from $p(\mathbf{x}_k|z_{1:k})$ using familiar state estimators, for e.g., the *expected a posteriori* (EAP) as

$$\hat{\mathbf{x}}_k^{\text{EAP}} \triangleq \int \mathbf{x}_k p(\mathbf{x}_k|z_{1:k})d\mathbf{x}_k \tag{2.8}$$

or the *maximum a posteriori* (MAP) as

$$\hat{\mathbf{x}}_k^{\text{MAP}} \triangleq \arg \sup_{\mathbf{x}_k} p(\mathbf{x}_k|z_{1:k}) \tag{2.9}$$

Other information contained in $p(\mathbf{x}_k|z_{1:k})$, for example, covariance, may also be obtained.

The implementation of the Bayes' filter requires the evaluation and storage of the entire pdf in (2.4) and (2.5). Moreover, the integrals involved in obtaining (2.4) are typically high-dimensional entities [68]. This can make the filtering process *intractable*. Hence the optimal Bayes' recursive solution is only theoretical², that is to say, analytic solutions are not always possible [56].

2.3 Multi-target Bayesian filtering

In this section, the Bayes' formulation for multi-target systems [35, 36] is first presented in section 2.3.2. Then the issues that influence Bayes' MTT filtering are discussed in sections 2.3.3 and 2.3.4.

2.3.1 Joint state space modelling

In a multiple target tracking (MTT) scenario, consider a zone \mathfrak{R} that defines a bounded region of surveillance, i.e., target activity outside \mathfrak{R} is not accounted for. In MTT, it is important to model the appearance and the disappearance of targets within the zone \mathfrak{R} . The actual number of targets N is either assumed to be known or modelled as a random variable if it is unknown.

In this analysis, the number of targets is modelled as a random variable. An additional state ϕ is introduced to treat the absence of targets. Then the state space for all the targets, now termed the *joint state space*, is expressed

²The exact implementation of the Bayes' filter is possible to a restrictive class of approaches, for example, the Kalman filter [20] (refer [56, 69] for other approaches).

as

$$\mathfrak{X}^* = \begin{cases} \phi & \text{if no targets are present} \\ \mathfrak{X} & \text{if one target is present} \\ \mathfrak{X} \times \mathfrak{X} & \text{if two targets are present} \\ \vdots & \vdots \\ \vdots & \vdots \end{cases} \quad (2.10)$$

If it is assumed that there are N_k statistically independent targets at time k , then the joint state space is expressed as

$$\mathfrak{X}^* = \mathfrak{X} \times \dots \times \mathfrak{X} \quad (2.11)$$

where the product is taken $N_k - 1$ times. The target state vector captures information about *all* the targets [42] present within \mathfrak{R} at time sample k . Hence the state vector is called the *joint target state*, and is denoted by X_k . This joint state is a concatenation of the individual target states and is represented as

$$X_k = (x_k^1, x_k^2, \dots, x_k^{N_k}) \quad (2.12)$$

X_k is a hypothesis of *all* the targets in the N_k dimensional joint space \mathfrak{X}^* and is modelled as a random variable.

2.3.2 The MTT Bayes' filter recursion

The goal of MTT is to evaluate X_k and N_k using the sensor evidence z_k . The Bayesian approach to MTT is conventionally called *Bayes' joint filtering*, in that the filter constructs a joint multi-target probability distribution (JMPD) [70, 71] of the target state as well as the number of targets. This JMPD is described as

$$p(X_k, N_k | z_{1:k}) = p(x_k^1, x_k^2, \dots, x_k^{N_k}, N_k | z_{1:k}) \quad (2.13)$$

The JMPD is a probability function and hence integrates to one. Moreover, unless the targets are distinguishable in some way, it is symmetric under

permutation of the target indices.

If the posterior $p(\mathbf{X}_{k-1}, N_{k-1} | z_{1:k-1})$ at time $k-1$ is available, then in a similar way to the single target Bayes' filter, the JMPD [72] at time k is obtained in two steps; a) prediction, and b) update. The Bayes' prediction is evaluated [44] as

$$\begin{aligned}
 p(\mathbf{X}_k, N_k | z_{1:k-1}) &= \int p(\mathbf{X}_k, N_k | \mathbf{X}_{k-1} = \phi, N_{k-1} = 0) p(\mathbf{X}_{k-1} = \phi, N_{k-1} = 0 | z_{1:k-1}) d\phi + \\
 &\quad \int p(\mathbf{X}_k, N_k | \mathbf{X}_{k-1} = (\mathbf{x}_{k-1}^1), N_{k-1} = 1) \\
 &\quad p(\mathbf{X}_{k-1} = (\mathbf{x}_{k-1}^1), N_{k-1} = 1 | z_{1:k-1}) d\mathbf{x}_{k-1}^1 + \\
 &\quad \int \int p(\mathbf{X}_k, N_k | \mathbf{X}_{k-1} = (\mathbf{x}_{k-1}^1, \mathbf{x}_{k-1}^2), N_{k-1} = 2) \\
 &\quad p(\mathbf{X}_{k-1} = (\mathbf{x}_{k-1}^1, \mathbf{x}_{k-1}^2), N_{k-1} = 2 | z_{1:k-1}) d\mathbf{x}_{k-1}^1 d\mathbf{x}_{k-1}^2 + \dots \\
 &= \sum_{n=0}^{\infty} \underbrace{\int \dots \int}_{n \text{ times}} p(\mathbf{X}_k, N_k | \mathbf{x}_{k-1}^1, \dots, \mathbf{x}_{k-1}^n, N_{k-1} = n) \\
 &\quad p(\mathbf{x}_{k-1}^1, \dots, \mathbf{x}_{k-1}^n, N_{k-1} = n | z_{1:k-1}) d\mathbf{x}_{k-1}^1 \dots d\mathbf{x}_{k-1}^n \\
 &= \sum_{n=0}^{\infty} \underbrace{\int \dots \int}_{n \text{ times}} p(\mathbf{X}_k, N_k | \mathbf{X}_{k-1}, N_{k-1} = n) \\
 &\quad p(\mathbf{X}_{k-1}, N_{k-1} = n | z_{1:k-1}) d\mathbf{X}_{k-1} \tag{2.14}
 \end{aligned}$$

It is evident from (2.14) that the MTT Bayes' prediction models both the unknown target number and the appearance and disappearance of the targets. In the Markovian transition function $p(\mathbf{X}_k, N_k | \mathbf{X}_{k-1}, N_{k-1})$, the hypothesised targets in \mathbf{X}_{k-1} survive into \mathbf{X}_k with a survival probability p_S . Consequently, the dimensions of \mathbf{X}_k and \mathbf{X}_{k-1} may not be the same. Moreover, since the number of targets is always a nonnegative integer, i.e., $N_{k-1} \in \mathbb{Z}^* = \{0\} \cup \mathbb{Z}^+$, the prediction is a summation over all possible target numbers (although in practice, this is limited to N_{\max}).

Once the new sensor data z_k is obtained, the prediction distribution in (2.14)

is updated using Bayes' rule according to

$$p(\mathbf{X}_k, N_k | z_{1:k}) = \frac{p(z_k | \mathbf{X}_k, N_k) p(\mathbf{X}_k, N_k | z_{1:k-1})}{p(z_k | z_{1:k-1})} \quad (2.15)$$

where $p(z_k | \mathbf{X}_k)$ is the likelihood function and $p(z_k | z_{1:k-1})$ is a normalising constant. The recurrence of (2.14) and (2.15) is the property that forms the basis for the Bayes' MTT filter.

To estimate the joint target state and the target number from $p(\mathbf{X}_k, N_k | z_{1:k})$, the marginal multi-target (MaM) estimator may be used³: here, the marginal distribution of N_k is formed according to

$$p(N_k | z_k) \triangleq \int p(\mathbf{X}_k, N_k | z_{1:k}) d\mathbf{X}_k \quad (2.16)$$

Then the expected number of targets is the MAP estimate

$$\hat{N}_k \triangleq \arg \sup_{N_k} p(N_k | z_k) \quad (2.17)$$

and the MaM estimate of the joint target state is obtained using

$$\hat{\mathbf{X}}_k^{\text{MaM}} \triangleq \arg \sup_{\mathbf{x}_k^1, \dots, \mathbf{x}_k^{\hat{N}_k}} p(\mathbf{x}_k^1, \dots, \mathbf{x}_k^{\hat{N}_k} | z_{1:k}) \quad (2.18)$$

Under the assumptions that; a) the number of targets at each time step is known, and b) targets neither appear or disappear, the Bayes' MTT prediction and update formalism [44] is simply a generalised form of that of the single target Bayes' filter and can be written as

$$p(\mathbf{X}_k | z_{1:k}) = \frac{p(z_k | \mathbf{X}_k)}{p(z_k | z_{1:k-1})} \int p(\mathbf{X}_k | \mathbf{X}_{k-1}) p(\mathbf{X}_{k-1} | z_{1:k-1}) d\mathbf{X}_{k-1} \quad (2.19)$$

2.3.3 Implications of the sensor model for the likelihood

Here, the implications of the choice of the sensor feed z_k on evaluating the likelihood function $p(z_k | \mathbf{X}_k)$ in (2.15) are discussed.

³Other Bayes' MTT estimators can be found in Ch. 14 of [32].

Data association approach

In traditional surveillance systems, the *direct* information received at the signal processing unit is usually; (*i*) a image (e.g., SONAR image), or (*ii*) a signal (e.g., acoustic signal). In many tracking applications, this information is further processed to generate M *noisy point target detections* which then are the observation feed to the filter. This input feed contains target detections, false alarms⁴ or clutter⁵ and is expressed as

$$Z_k = \{z_k^1, \dots, z_k^M\} \quad (2.20)$$

where the m th measurement $z_k^m \in \mathfrak{Z}$ (recall from section 2.1 that \mathfrak{Z} is the single target observation space). Here, it is assumed that multiple targets *cannot* generate one point target observation. Hence, while evaluating the likelihood in (2.15), it is important to know which measurement in Z_k relates to which target hypothesis in X_k — the *data association*. The use of the observation model in (2.20) casts the MTT problem into two domains: a) filtering, and b) data association. Data association [73] aids in identifying close targets and avoiding confusing them, for example, aircraft flying close [1].

One method of achieving data association is by *multiple hypothesis tracking* (MHT) [74]. The MHT filter operates by first creating hypotheses about *all* the possible associations between the sensor detections and the target states, and then choosing one hypothesis based on its weight, interpreted as the probability that it is the correct one. A few measures that compute the probability of a hypothesis being correct are the Mahalanobis distance or global association distance [32]. This method, however, is computationally expensive as it propagates all the possible hypotheses every time step. Another approach to data association is *joint probabilistic data association*

⁴False alarms are sensor detections that did not originate from the targets, but are caused by noise.

⁵Clutter are sensor detections from objects which are not of interest, e.g., for a satellite observing sensor, stars are the clutter.

(JPDA) [43] in which all the potential target-measurement associations are treated as a single statistical unit, i.e., all measurements are assumed to be associated to every target to some extent [27]. The JPDA filter [5, 10, 11] computes the degree to which each measurement contributed to each target and then creates composite tracks that represent the correct one.

Association free approach

Much work on MTT and information fusion relates to observations which are point target detections described in (2.20). However, converting sensor data into point target measurements using pre-processing tools like thresholding introduces substantial information loss, especially at low signal-to-noise ratio (SNR). In contrast, feeding the raw information (image or signals) from the sensors directly to the filter allows merged observations — multiple targets generating the one observation datum⁶ [44]. This avoids data association and hence overcomes the combinatorial problem of target-measurement association.

2.3.4 Curse of dimensionality

Since the JMPD $p(\mathbf{X}_k | z_{1:k})$ provides *complete* information about the multi-target scenario at time k , the filtering is considered *optimal*. However the filter suffers from high computational complexity:

Example: Consider that targets lie in the discrete 1D space $\mathfrak{X} = \{1, 2, \dots, 10\}$. In a single target scenario, the state space contains 10 values and the dimensionality of \mathbf{x}_k is 1, for example, $\mathbf{x}_k = 3$. However, in a two target scenario, the joint state space $\mathfrak{X}^* = \mathfrak{X} \times \mathfrak{X}$ will now contain 100 values and the dimensionality of the joint state \mathbf{X}_k is 2, for example, $\mathbf{X}_k = (3, 7)$.

⁶Examples of merged observations: a) in a image, the intensity of a single pixel could be influenced by multiple targets, and b) in a phased sensor array, signals from multiple signals are added at the sensors.

If tracking involved computing the probability of each possible combination of states, then it is evident that the computational effort devoted to evaluate the joint pdf worsens exponentially. This complexity problem, conventionally termed “*curse of dimensionality*”⁷, [45] makes the filtering process intractable.

2.4 The Kalman filter

The computational intractability of the Bayes’ filter is overcome using approximate solutions. A popular Bayes’ filter which provides a closed-form solution to target state estimation is the Kalman filter [20]. In this section, the Kalman filter that conceptualises Bayes’ recursion in a discrete time linear filtering perspective is presented. The filter’s mathematical treatment is outlined in section 2.4.1 which is followed by a brief survey of its historical development in section 2.4.2.

Kalman filtering is regarded as a least square estimation process with provision to model the dynamic stochastic target state⁸. Consequently its solution is *optimal*. The filter assumes that the posterior pdf is Gaussian distributed at every time step [5] and hence can be parameterised by a mean and a covariance. The Gaussianity assumption in the Kalman filter will be consistent with time [78] under the following conditions:

1. The process and observation noise sequences q_{k-1} in (2.2) and r_k in (2.3) respectively are drawn from Gaussian distributions with known parameters,
2. (2.2) and (2.3) are known linear functions.

⁷The term was coined by R. E. Bellman in 1957 when considering dynamic optimisation problems [75, 76].

⁸In other words, Kalman filter is a time variant Wiener filter [77].

Hence, the process and observation models can be re-written as

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + q_{k-1} \quad (2.21)$$

$$z_k = H_k \mathbf{x}_k + r_k \quad (2.22)$$

where F_k and H_k are linear functions. The noise sequences q_{k-1} and r_k are random samples drawn from the distributions $\mathcal{N}(0, Q_{k-1})$ and $\mathcal{N}(0, R_k)$ respectively. If these conditions are not met, the Kalman filter can still be used, but is regarded in that case as a *second order moment approximation* to the single target Bayes' filter. The single target posterior $p(\mathbf{x}_k | z_{1:k})$, under the assumption that it is essentially Gaussian and not skewed, can be compressed into its *approximate sufficient statistics* — the first and second moments as

$$p(\mathbf{x}_k | z_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_k, P_k) \quad (2.23)$$

with mean $\hat{\mathbf{x}}_k$ and covariance P_k , the closed-form of which is presented in the following section.

2.4.1 The Kalman filter recursion

The derivation of the Kalman filter [20, 32, 79, 80] is not central to this thesis and is not provided here. The Kalman filter describes the posterior $p(\mathbf{x}_{k-1} | z_{1:k-1})$ at time $k - 1$ as a Gaussian. Hence the filter is parameterised by the mean $\hat{\mathbf{x}}_{k-1}$ and its error covariance P_{k-1} . At the next time index k , the filter operates in two stages; (*i*) prediction, in which the *a priori* state estimate $\hat{\mathbf{x}}'_k$ and its error covariance P'_k are computed, and (*ii*) update, in which the *a posteriori* state estimate $\hat{\mathbf{x}}_k$ and its error covariance P_k are computed.

The filter recursion from time $k - 1$ to k can be expressed as

$$p(\mathbf{x}_{k-1} | z_{1:k-1}) = \mathcal{N}(\hat{\mathbf{x}}_{k-1}, P_{k-1}) \quad (2.24)$$

$$p(\mathbf{x}_k | z_{1:k-1}) = \mathcal{N}(\hat{\mathbf{x}}'_k, P'_k) \quad (2.25)$$

$$p(\mathbf{x}_k | z_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_k, P_k) \quad (2.26)$$

where

$$\hat{x}'_k = F_k \hat{x}_{k-1} \quad (2.27)$$

$$P'_k = F_k P_{k-1} F_k^T + Q_{k-1} \quad (2.28)$$

$$\hat{x}_k = \hat{x}'_k + K_k(z_k - H_k \hat{x}'_k) \quad (2.29)$$

$$P_k = (I - K_k H_k) P'_k \quad (2.30)$$

The posterior mean in (2.29) is estimated by expressing [80] the *a posteriori* state estimate \hat{x}_k as a linear combination of the prediction \hat{x}'_k and the weighted difference between sensor observation z_k and predicted measure $H_k \hat{x}'_k$. The *Kalman gain* K_k , expressed [79] as

$$K_k = P'_k H_k^T ((H_k P'_k H_k^T + R_k))^{-1} \quad (2.31)$$

minimises the error covariance in (2.30), i.e., for small P'_k the scaling of the innovation by K_k is also small, and for small R_k the scaling of the innovation by K_k is high. The equations (2.27), (2.28), (2.31), (2.29) and (2.30) complete the Kalman recursive filtering operation.

2.4.2 Historical progress of the Kalman filter

A year after the Kalman filter was proposed, in 1961, R. E. Kalman and R. S. Bucy jointly proposed a linear continuous time counterpart to the Kalman filter, namely the Kalman-Bucy filter [81] that estimates the target state using stochastic differential equations. The filter can also be derived in a Bayesian framework [82].

Computing the Kalman gain K_k every time step is computationally expensive because of the matrix inversion in (2.31). The constant gain Kalman filter [83] overcomes this complexity by assuming the gain to be constant throughout time. Although this assumption impedes the effectiveness of the filter, it is suited for highly observable targets (small values of matrix norms $\|Q_{k-1}\|$ and $\|R_k\|$) with minimal rapid manoeuvres, e.g., aeroplanes or ships.

Alpha-Beta filters [84] are one example of the constant gain Kalman filter. Appending additional information like sensor bias, blurring of targets due to bad weather, etc., to the target state vector improves knowledge about the uncertainty of the estimate. The Schmidt-Kalman filter [85] accomplishes this information improvement without the need to increase the dimensionality of the state vector and thereby reduces complexity.

Given that the Kalman filter is limited to linear Gaussian target systems, noteworthy effort has been made to derive optimal non-linear filters, e.g., [86–89]. Additionally, several sub-optimal versions have been proposed to the Kalman filter to adapt it for non-linear systems. The extended Kalman filter (EKF) [90] extends the Kalman filter to non-linear systems by linearising about an estimate of the current mean and covariance. Here the state distribution is approximated by a Gaussian random variable and then propagated through the first order Taylor series expansion of the non-linear system [91] in order to calculate the mean and covariance. Higher order EKFs are obtained by retaining more terms of the Taylor series expansion. The unscented Kalman filter (UKF) [92, 93] approximates the Gaussian target state random process by a set of carefully chosen sample points that capture the true mean and covariance [94]. These points, when propagated in a non-linear system, capture the true mean and covariance accurately up to second order [95] (only for scalar states [96]). A summary of numerous other variants to Kalman filtering can be found in [97]. The EKF, UKF and other variants always approximate the posterior to be Gaussian, and introduce significant bias if the actual distribution is non-Gaussian; they cannot model the higher order moments of non-Gaussian posterior distributions.

The Kalman filter for MTT will generate a target state estimate (mean) and covariance estimate for each of the targets present in the scene. However, it is important to relate the sensor observations to each of the targets — the data association. The problem of data association is treated in three ways: firstly, the single hypothesis correlation (SHC) technique builds a track

table containing the state estimate and the error covariance for each target, and updates them based on the minimum association distance between each target and each sensor using the *Mahanalobis* distance, given by

$$d(\mathbf{x}_k^p | z_k) \triangleq (z_k - H_k \mathbf{x}_k^p)^T (H_k P_k^p H_k^T + R_k)^{-1} (z_k - H_k \mathbf{x}_k^p) \quad (2.32)$$

where \mathbf{x}_k^p and P_k^p are the state and the error covariance of the p th target respectively. In other words, the SHC can be visualised as a bank of parallel Kalman filters, one for each target, with filters being removed if their covariance estimate crosses a certain threshold. The SHC filter is forced to declare that a single measurement target association is correct, which is not an optimal representation when uncertainty in the association is significant. A second approach to data association is the multiple hypothesis tracking (MHT) [74] filter; however, as mentioned in section 2.3, this method involves high computational cost. Thirdly, the composite hypothesis correlation (CHC) [32] filter uses the concept of the JPDA and computes the best possible target-measurement association by using a joint formulation of all the association hypotheses.

The Kalman filter has been widely used in several array processing applications, a few examples of which include: positioning of ships [98], under-water submarine navigation [99], underwater SONAR tracking [100], rainfall estimation [101], detecting signals from buried objects [102], tracking seabed parameters [103], recording seismic activity [104], localisation using microphone arrays [105], spacecraft tracking [106], guiding satellites into their orbits [107] and estimating heart-beat [4].

2.5 The sequential Monte Carlo

The high dimensional integration in the Bayes' recursive solution makes the process computationally intractable. However, the exact closed-form solution to Bayes' filtering for linear Gaussian systems is possible using the Kalman filter described in section 2.4. In the current and subsequent sections, Bayes'

approximations for more realistic non-linear and non-Gaussian systems are presented. These approaches provide only *approximate* solutions and hence are sub-optimal.

A prominent Bayesian approximation is the sequential Monte Carlo (SMC) approach. Monte Carlo⁹ is a stochastic *sampling* approach aimed at tackling difficult numerical integration problems. Since their first use in 1949 by Metropolis and Ulam [22] in the Los Alamos laboratory, Monte Carlo methods have been explored to address many intractable problems in physics and statistics [108]. The SMC, which is a combination of Monte Carlo sampling and Bayesian statistics, is a powerful tool for online estimation of Markovian target systems. The formal introduction of Monte Carlo to Bayesian target tracking was made in 1993 by Gordon et al. and the resulting method is called the particle filter (PF) [21]. The PF belongs to the class of SMC [109] methods and approximates the probability distributions using independent *point mass* representations — also called *samples* or *particles*. These particles are sampled directly from the state space and weighted through the principle of *importance sampling*. The recursive propagation of this particle system results in an accurate Bayesian approximation [110, 111]. The primary focus of this thesis is the development of PF schemes to overcome various limitations in Bayesian MTT for array processing applications. Hence the filter is discussed in detail. Monte Carlo importance sampling is first introduced in sections 2.5.1 and 2.5.2. This is followed by the PF and its operational issues in sections 2.5.3, 2.5.4 and 2.5.6. A variant of the PF which will be further referred to in this thesis is then outlined in section 2.5.7. Finally, the historical developments of the PF are discussed in section 2.5.8.

2.5.1 Monte Carlo sampling

Here, Monte Carlo sampling is introduced using a simple derivation. Consider a statistical problem of estimating the expected value of a function $g(x)$. This

⁹Named after Monte Carlo Casino in Monaco.

can also be considered as estimating a Lebesgue-Stieltjes integral

$$E[g(x)] = \int_x g(x) dp(x) \quad (2.33)$$

where the integration of $g \rightarrow \mathbb{R}$ is defined with respect to a Lebesgue measure [112]. If $g(x)$ is intractable, then the computation of $E[g(x)]$ is difficult. Monte Carlo sampling could be employed to treat such situations. The method approximates $E[g(x)]$ using a set of independent and identically distributed (i.i.d) samples (also called particles) $\{x^1, \dots, x^I\}$, where for $i = 1, \dots, I$

$$x^i \sim p(\cdot) \quad (2.34)$$

Then the Monte Carlo estimate of $g(x)$ is

$$\hat{g} = \frac{1}{I} \sum_{i=1}^I g(x^i) \quad (2.35)$$

The expected value of the Monte Carlo estimate is given by

$$\begin{aligned} E[\hat{g}] &= E\left[\frac{1}{I} \sum_{i=1}^I g(x^i)\right] \\ &= \frac{1}{I} E\left[\sum_{i=1}^I g(x^i)\right] \\ &= \frac{1}{I} \sum_{i=1}^I E[g(x^i)] \\ &= E[g(x)] \end{aligned} \quad (2.36)$$

and the variance of the Monte Carlo estimate is given by

$$\begin{aligned}
 \text{var}[\hat{g}] &= \text{var}\left[\frac{1}{I} \sum_{i=1}^I g(\mathbf{x}^i)\right] \\
 &= \frac{1}{I^2} \text{var}\left[\sum_{i=1}^I g(\mathbf{x}^i)\right] \\
 &= \frac{1}{I^2} \sum_{i=1}^I \text{var}[g(\mathbf{x}^i)] \\
 &= \frac{I}{I^2} \text{var}[g(\mathbf{x})] \\
 &= \frac{\text{var}[g(\mathbf{x})]}{I}
 \end{aligned} \tag{2.37}$$

By the strong law of large numbers [113, 114], the Monte Carlo estimate \hat{g} converges to the true value of $E[g(\mathbf{x})]$ in (2.33) as $I \rightarrow \infty$. Monte Carlo sampling, by itself, cannot be employed in Bayesian filtering because the function $p(\mathbf{x})$ from which particles are drawn is unknown. This problem is addressed by importance sampling, which is described in the following section.

2.5.2 Importance sampling

Importance sampling aims to represent $p(\mathbf{x})$ by drawing a set of weighted particles $\{\mathbf{x}^i, w^i\}_{i=1}^I$ (I is the total number of particles) from regions of “importance” in the distribution. This is achieved by drawing from a *proposal* or *importance* distribution $q(\mathbf{x})$ that is similar to $p(\mathbf{x})$. Discussion on the choice of $q(\mathbf{x})$ is deferred until section 2.5.6. Consider the integral

$$\int g(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \int g(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \tag{2.38}$$

The Monte Carlo *representation* of $p(\mathbf{x})$ is expressed as

$$p(\mathbf{x}) \approx \sum_{i=1}^I w^i \delta(\mathbf{x} - \mathbf{x}^i) \tag{2.39}$$

where the weights (also called *importance weights*) are

$$w^i = \frac{p(\mathbf{x}^i)}{q(\mathbf{x}^i)} \quad (2.40)$$

If the normalising factor of $p(\mathbf{x})$ is not known, the importance weights can only be calculated up to a normalising constant. The weights are normalised to ensure that $\sum_{i=1}^I w^i = 1$. Then the Monte Carlo estimate to the filtered state moment $E[g(\mathbf{x})]$ is

$$\hat{g} = \frac{1}{I} \left(\frac{\sum_{i=1}^I w^i g(\mathbf{x}^i)}{\sum_{i=1}^I w^i} \right) \quad (2.41)$$

and the variance of the importance sampling estimate is

$$\begin{aligned} \text{var}(\hat{g}) &= \frac{1}{I} \text{var}[g(\mathbf{x}) w] \\ &= \frac{1}{I} \text{var}\left[g(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})}\right] \\ &= \frac{1}{I} \int \left[\frac{g(\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})} - E[g] \right]^2 q(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{I} \int \left[\frac{(g(\mathbf{x})p(\mathbf{x}))^2}{q(\mathbf{x})} - 2g(\mathbf{x})p(\mathbf{x})E[g] \right] d\mathbf{x} + \frac{(E[g])^2}{I} \\ &= \frac{1}{I} \left(\int \left[\frac{(g(\mathbf{x})p(\mathbf{x}))^2}{q(\mathbf{x})} \right] d\mathbf{x} - (E[g])^2 \right) \end{aligned} \quad (2.42)$$

It is obvious from (2.42) that the variance of the Monte Carlo estimate is minimised when $q(\mathbf{x})$ matches $p(\mathbf{x})$. Moreover, according to the weak law of large numbers [115], it is found that

$$\hat{g} \rightarrow \frac{E[g(\mathbf{x}) w]}{E[w]} \quad (2.43)$$

Returning to Bayes' filtering, the posterior $p(\mathbf{x}_k | z_{1:k})$ is the true pdf, the non-linearity and/or non-Gaussianity of which makes it difficult to draw particles from. However, $p(\mathbf{x}_k | z_{1:k})$ can be represented using a weighted particle set $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^I$, where I is the total number of particles, drawn from a known

proposal distribution $q(\mathbf{x}_k | z_{1:k})$. The importance weights of these particles, as in (2.40) is

$$w_k^i = \frac{p(\mathbf{x}_k^i | z_{1:k})}{q(\mathbf{x}_k^i | z_{1:k})} \quad (2.44)$$

Propogating these weighted particles sequentially in a Bayes' framework is performed by sequential importance sampling (SIS), which is described in the following section.

2.5.3 Sequential importance sampling

Consider that the weighted particles $\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^I$ that represent $p(\mathbf{x}_{k-1} | z_{1:k-1})$ at time $k-1$ are available. The aim of SIS is to obtain $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^I$ that represent the posterior $p(\mathbf{x}_k | z_{1:k})$ at the next time step k . This is accomplished by drawing particles from a known importance distribution $q(\mathbf{x}_k | z_{1:k})$. The process is done in two steps; a) prediction, and b) update. The prediction step propagates the particles $\{\mathbf{x}_{k-1}^i\}_{i=1}^I$ to time step k as follows: the importance distribution can be factorised as

$$q(\mathbf{x}_{1:k} | z_{1:k}) = q(\mathbf{x}_k | \mathbf{x}_{1:k-1}, z_{1:k}) q(\mathbf{x}_{1:k-1} | z_{1:k-1}) \quad (2.45)$$

This implies that particles $\mathbf{x}_{k-1}^i \sim q(\mathbf{x}_{k-1} | z_{1:k-1})$ are propagated to time k by augmenting their states by another set of particles drawn from

$$\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}, z_k) ; i = 1, \dots, I \quad (2.46)$$

The update step evaluates the weights of $\{\mathbf{x}_k^i\}_{i=1}^I$ using the new sensor evidence z_k . This is done as follows: the posterior pdf

$$\begin{aligned}
 p(\mathbf{x}_{1:k}|z_{1:k}) &= \frac{p(z_{1:k}|\mathbf{x}_{1:k})p(\mathbf{x}_{1:k})}{p(z_{1:k})} \\
 &= \frac{p(z_k, z_{1:k-1}|\mathbf{x}_{1:k})p(\mathbf{x}_{1:k})}{p(z_k, z_{1:k-1})} \\
 &= \frac{p(z_k|z_{1:k-1}, \mathbf{x}_{1:k})p(z_{1:k-1}|\mathbf{x}_{1:k})p(\mathbf{x}_{1:k})}{p(z_k|z_{1:k-1})p(z_{1:k-1})} \\
 &= \frac{p(z_k|z_{1:k-1}, \mathbf{x}_{1:k})p(\mathbf{x}_{1:k}|z_{1:k-1})p(z_{1:k-1})p(\mathbf{x}_{1:k})}{p(z_k|z_{1:k-1})p(z_{1:k-1})p(\mathbf{x}_{1:k})} \\
 &= \frac{p(z_k|\mathbf{x}_k)p(\mathbf{x}_{1:k}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
 &= \frac{p(z_k|\mathbf{x}_k)p(\mathbf{x}_k, \mathbf{x}_{1:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
 &= \frac{p(z_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, z_{1:k-1})p(\mathbf{x}_{1:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
 &= \frac{p(z_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{1:k-1})p(\mathbf{x}_{1:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
 &\propto p(z_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{1:k-1})p(\mathbf{x}_{1:k-1}|z_{1:k-1})
 \end{aligned} \tag{2.47}$$

By substituting (2.45) and (2.47) in (2.44), the unnormalised particle weight becomes

$$\begin{aligned}
 \tilde{w}_k^i &\propto \frac{p(z_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)p(\mathbf{x}_{k-1}^i|z_{1:k-1})}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, z_k)q(\mathbf{x}_{k-1}^i|z_{1:k-1})} \\
 &= w_{k-1}^i \frac{p(z_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, z_k)} ; i = 1, \dots, I
 \end{aligned} \tag{2.48}$$

The weights are normalised $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^I \tilde{w}_k^j}$ and $p(\mathbf{x}_k|z_{1:k})$ is represented by

$$p(\mathbf{x}_k|z_{1:k}) \approx \sum_{i=1}^I w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{2.49}$$

Degeneracy

As the SIS algorithm progresses, the particles near the actual target gain more weight, and those that are far away have negligible weights. This implies that high computational effort is required to update these negligible weight particles that do not contribute to the true posterior [54]. This problem is called *degeneracy*. That is, if the *genealogy* of a particle is observed by tracing its parent all the way to the initial time step, it can then be seen that as particles propagate in time, it will become more likely that a single particle will be the parent of all its descendants and this parent need not necessarily have high importance. This implies that one particle grabs all the weight and this particle may not necessarily contribute to the true posterior. This is illustrated using a simple example in Fig. 2.1.

One measure of efficiency in this context is the effective sample size [52], defined as

$$I_{\text{eff}} = \frac{1}{\sum_{i=1}^I (w_k^i)^2} \quad (2.50)$$

such that $I_{\text{eff}} \ll I$ and low I_{eff} exhibits high degeneracy. The solution to degeneracy is to *resample* the particles (in accordance to their weights) with replacement whenever I_{eff} falls below a certain threshold [109]. This resampling is presented in the following section.

2.5.4 Sequential importance resampling: The PF

The goal of resampling is to eliminate particles that have negligible weights and replace them by copies of other particles that have large weights. This is the same as replacing (2.49) by

$$p(\mathbf{x}_k | z_{1:k}) \approx \sum_{i=1}^I \frac{n_k^i}{I} \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (2.51)$$

$$\approx \sum_{i=1}^I \frac{1}{I} \delta(\mathbf{x}_k - \mathbf{x}_k^{*i}) \quad (2.52)$$

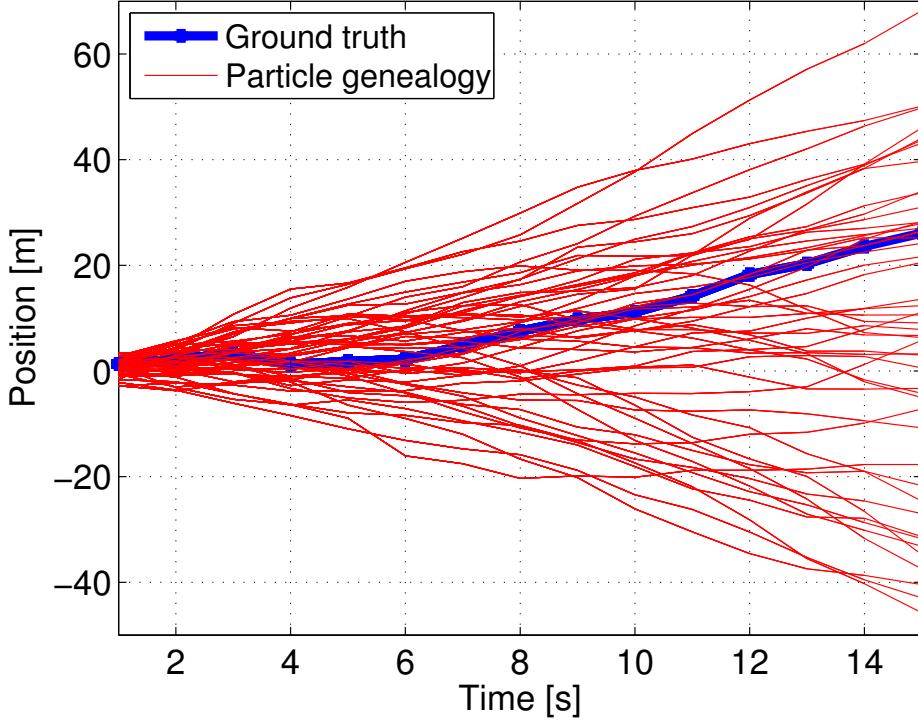


Figure 2.1: An example illustrating the effect of degeneracy. Here, the state evolution is conducted using a 2D constant velocity model. The measurements are collected using a position observing sensor model. The model is described in detail later in section 5.6.2 of chapter 5. The number of particles is $I = 50$. The genealogies of all the 50 particles are plotting as red solid lines.

where n_k^i is the number of replications/ duplications of the particle x_k^i and $\{x^{*i}\}_{i=1}^I$ is the new resampled set of particles. That is, as time progresses, the resampling process decreases the chance that one particle which initially had a low importance, is highly trusted. By replacing particles having low importance by those having high importance, the particle genealogies will go through more likely paths. This is shown in Fig. 2.2, where the example in Fig. 2.1 is now repeated with resampling.

The use of resampling with SIS is consequently called *sequential importance*

resampling (SISR) or PF¹⁰. Although resampling reduces the degeneracy problem by discarding those particles that do not contribute to the posterior, it could lead to *particle impoverishment* — a problem in which *very few* particles get replicated *many* times thereby reducing the richness (or diversity) of the particles in effectively representing the posterior: some methods that overcome this problem are discussed briefly in section 2.5.8. Being an approximation technique, the PF converges almost definitely to the true posterior as $I \rightarrow \infty$. Convergence results for the PF can be found in [116, 117].

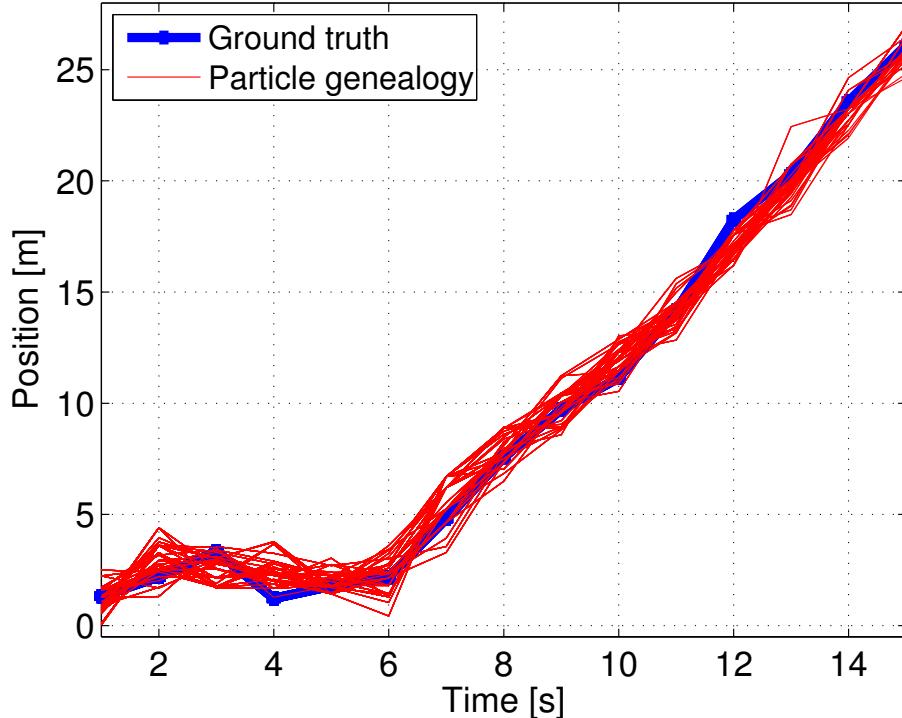


Figure 2.2: An example illustrating how resampling overcomes the effect of degeneracy. The ground truth, process and measurement model and sample size I are the same as used in Fig. 2.1.

¹⁰Various other names for the PF are *bootstrap filtering*, *condensation algorithm* and *survival of the fittest*.

Resampling techniques are majorly divided into two classes; a) stochastic, and b) deterministic. Detailed studies on resampling algorithms can be found in [118–121]. A few popular resampling techniques, that will be referred to in chapter 5 are presented here.

Multinomial selection

The aim of stochastic resampling is the *multinomial selection* of n_k^i in (2.51) which is equivalent to selecting x^{*j} for $j = 1, \dots, I$ particles, such that $P(x^{*j} = x^i) = w^i$, i.e., the particles are resampled in accordance to their weights. This multinomial selection can be done as follows

- Draw a random sample $u_k^i \sim \mathcal{U}(0, 1]$,
- Assign index $j \leftarrow i$ such that $\sum_{s=1}^{i-1} w_k^s < u_k^i \leq \sum_{s=1}^i w_k^s$.

In the resampling literature, there are four variations of this multinomial selection. These variants are briefly discussed here.

1. Multinomial resampling: This method is simply the conventional multinomial selection scheme and also termed *random resampling*. Here, draw I random samples u_k , each drawn according to $u_k^i \sim \mathcal{U}(0, 1]$. Then allocate n_k^i copies of x_k^i to the new distribution according to

$$n_k^i = \text{the number of } u_k \in \left(\sum_{s=1}^{i-1} w_k^s, \sum_{s=1}^i w_k^s \right] \quad (2.53)$$

This process is equivalent to the two-step multinomial selection process described above. (2.53) also ensures that $\sum_{i=1}^I n_k^i = I$. Finally, the weights of the new set of particles are set to $1/I$.

2. Stratified resampling: The process of creating the random samples u_k by sampling I times from the space $(0, I]$ leads to a high Monte Carlo (MC) variance. Stratified resampling is motivated by the idea that creating u_k by sampling I times, once from each of the I independent ordered subsets

of equal size $\left(\frac{i}{I}, \frac{i+1}{I}\right]$ for $i = 0, \dots, I - 1$, will reduce the MC variance. The space $(0, I]$ which is now divided into I disjoint subsets is referred to as *strata*. The selection of the random samples is conducted *independently* within each stratum, i.e., u_k is obtained by sampling I times, once from each of the I ordered subsets. Hence stratified resampling is similar to multinomial resampling in (2.53), except that now, the random samples are

$$u_k^i = \frac{(i-1) + \tilde{u}_k^i}{I} \text{ with } \tilde{u}_k^i \sim \mathcal{U}(0, 1], i = 0, \dots, I - 1 \quad (2.54)$$

The weights of the new set of particles are set to $1/I$.

3. Systematic resampling: This method aims to further reduce the MC variance. The procedure is almost similar to that of stratified resampling: the stratification process is unchanged, but the random samples drawn from the strata are *no longer independent*, i.e., all samples have the same position within each stratum. Hence systematic resampling is similar to multinomial resampling in (2.53), except that now, the random samples are

$$u_k^i = \frac{(i-1) + \tilde{u}_k}{I}, \quad i = 0, \dots, I - 1 \quad (2.55)$$

with $\tilde{u}_k \sim \mathcal{U}(0, 1]$. The weights of the new set of particles are set to $1/I$.

Example illustration: The process of selecting the random samples u_k in each of the above mentioned methods is illustrated in Fig. 2.3 using a simple example. Here, the grey solid line is the cumulative sum of the weights of $I = 5$ particles. The weight vector is chosen to be $w_k = [0.1, 0.2, 0.5, 0.15, 0.05]$. The space from which random samples are drawn is spanned by the red double arrow(s). The random sample values are pointed using the I blue arrows. In multinomial resampling, each random sample is drawn from the space $(0, I]$ (shown using a single red double arrow) according to $u_k^i \sim \mathcal{U}(0, I]$. In stratified resampling, the space $(0, I]$ is stratified into I disjoint subsets (shown using I red double arrows) and the random samples are drawn independently from each strata according to (2.54). In systematic resampling, the stratification is unchanged (shown using I red double arrows) but the

random sample value chosen from each stratum is fixed to the same value, i.e., in other words, the entire grid is offset (shown using the black double arrows) by a single random value. In all the three examples, the random sample values are pointed using I blue arrows. Once u_k is obtained, resampling is performed using the process described in (2.53).

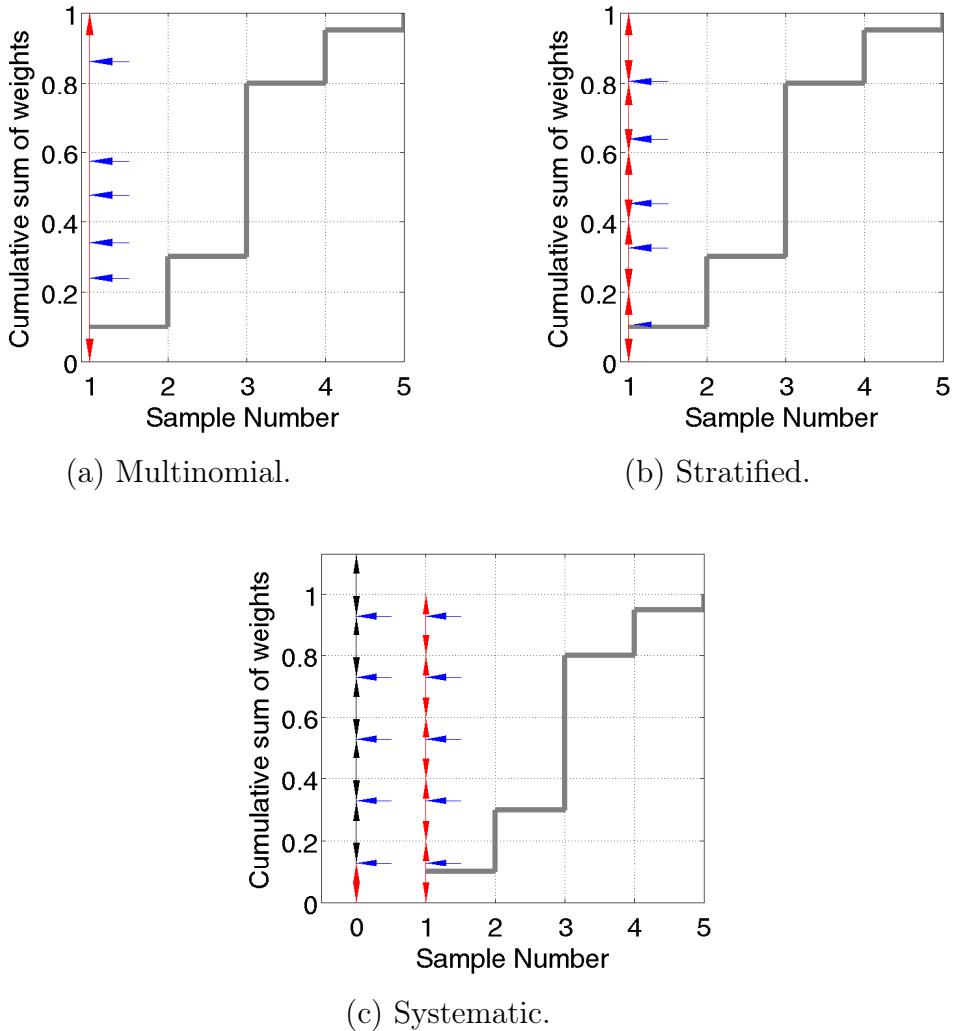


Figure 2.3: Illustration of drawing random samples u_k in the stochastic resamplers.

4. Residual resampling: This method is motivated by the idea that the

number of replications of many weights can be computed without stochasticity. Here, the number of replications n_k^i of x_k^i is given by $\lfloor Iw_k^i \rfloor$ and $K = \sum_{i=1}^I \lfloor Iw_k^i \rfloor$. In general $K < I$ and the original sample size I could be retained by stochastically resampling the K new particles to obtain the remainig $I - K$ new particles.

Partial deterministic resampling

The second class of resampling is deterministic, which is threshold based. A hybrid between deterministic and stochastic resampling is partial deterministic resampling [120, 122], which was proposed to reduce PF computational complexity and processing time. This combined with distributed PF approaches would aid in PF parallelisation. The general idea is this: the particles are categorised using a upper threshold $T_h > 1/I$ and a lower threshold $0 < T_l < T_h$. Those particles with weights above T_h are termed *dominant*, those particles with weights below T_l are termed *negligible*, and those particles with weights between the two thresholds are termed *moderate*. The dominant particles are retained, the negligible particles are eliminated and the moderate particles are resampled using any stochastic resampler.

One variant of the partial deterministic resampling is as follows: consider N_h and N_l to be the number of dominant and negligible particles respectively. The set of dominant and negligible particles are replaced by *only* the dominant particles using the following rule – the first $N_l - \lfloor \frac{N_l}{N_h} \rfloor N_h$ dominant particles are replicated $r = \lfloor \frac{N_l}{N_h} \rfloor + 2$ times and the remaining are replicated $\lfloor \frac{N_l}{N_h} \rfloor + 1$ times. Each replicated dominant particle x_k^i is reweighted as

$$w_k^i = \frac{w_k^i}{r} + \frac{W_l}{N_h + N_l} \quad (2.56)$$

where W_l is the sum of the negligible weights. The moderate particles, on the other hand, are retained and reweighted to the same number $1/I$.

The complete PF operation using 100 particles in one-dimensional scenario at a single time step is illustrated in Fig. 2.4.

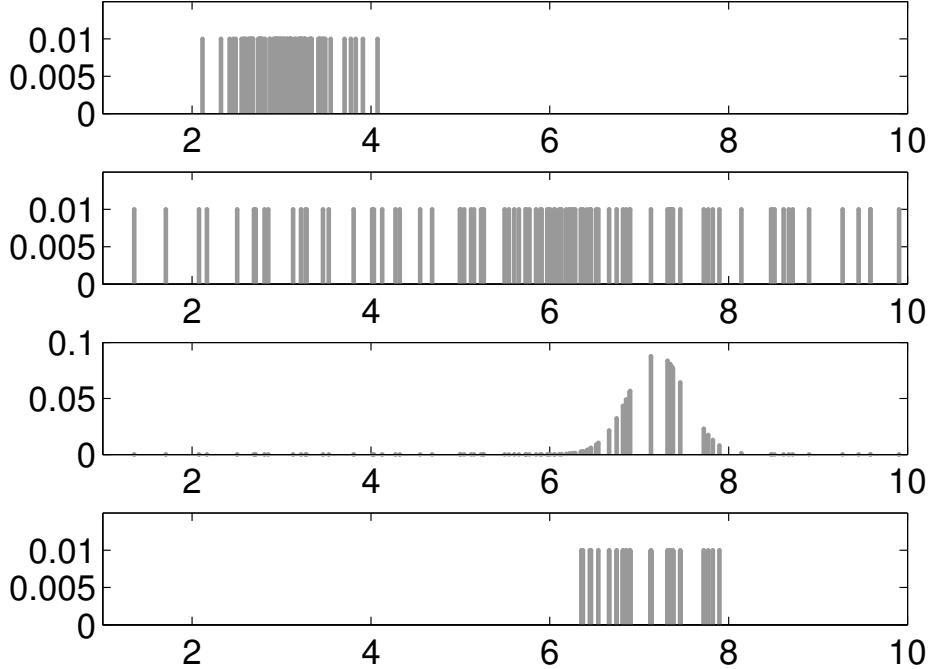


Figure 2.4: The Metropolis resampling method showing the Dirac delta function of the particles. The first figure shows $I = 100$ equally weighted samples $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^I$ that represent the posterior $p(x_{k-1}|z_{1:k-1})$. The second and third figures describe the SIS stage of the PF: (i) the second figure shows the predicted set of particles (obtained using a random walk [123] process model), and (ii) the third figure shows the weight updated set of particles (with $z_k = 7.2$) that represent the posterior $p(x_k|z_{1:k})$. It can be seen that particles located far away from the observation obtain low importance, and propagating these particles in time will lead to degeneracy. The fourth figure describes the SISR stage of the PF — particles of low importance are replaced by copies of other particles that have high importance. The weights are then reset to $1/I$. The fourth figure shows the final set of particles $\{x_k^i, w_k^i\}_{i=1}^I$ which will be propagated to the next time step.

2.5.5 The MTT joint PF

The single target PF can be directly generalised to MTT problems [42, 110] using the joint filter approach described in section 2.3. The conventional

MTT joint PF [44, 71, 124] is presented in this section.

Similar to the single target PF, the MTT joint PF approximates the joint posterior $p(\mathbf{X}_k, N_k | z_{1:k})$ using a set of I weighted joint particles $\{\mathbf{X}_k^i, w_k^i\}_{i=1}^I$. Each particle in the mix is a *hypothesis* about the states of *all* the targets, i.e., a particle is now a concatenation of individual target states and will track all the targets *simultaneously*. The (usually) unknown number of targets N_k is modelled as a random variable and hence is included in the particle state. A single particle is now expressed as

$$(\mathbf{X}_k^i, N_k^i) = (\mathbf{x}_k^{i,1}, \dots, \mathbf{x}_k^{i,N_k^i}) \quad (2.57)$$

where $N_k^i \in \mathbb{Z}^*$ is the number of single target hypotheses in the joint particle \mathbf{X}_k^i . Each particle is allowed to hypothesise any number of targets, i.e., N_k^i could be different (or the same) from that of another particle. During implementation, it is a common practice to limit the number of individual states in a particle to N_{\max} .

If the weighted joint particle set $\{\mathbf{X}_{k-1}^i, w_{k-1}^i\}_{i=1}^I$ that approximates the posterior $p(\mathbf{X}_{k-1}, N_{k-1} | z_{1:k-1})$ at time $k-1$ is available, the joint PF approximation to the posterior at time k is obtained in two steps; a) prediction, and b) update. In the prediction, a new set of particles are sampled from the previous ones according to

$$(\mathbf{X}_k^i, N_k^i) \sim q(\mathbf{X}_k, N_k | \mathbf{X}_{k-1}, N_{k-1}, z_k) \quad (2.58)$$

Target appearance and disappearance can be modelled by including a survival probability p_S for each individual target state in the joint particle — that is to say: the n th single target state vector $\mathbf{x}_{k-1}^{i,n} \in \mathbf{X}_{k-1}^i$ is either; a) retained with probability p_S and predicted into the next state, or b) eliminated with probability $1 - p_S$.

In the update step, the weight of the joint particle is updated as

$$\tilde{w}_k^i = w_{k-1}^i \frac{p(z_k | \mathbf{X}_k, N_k) p(\mathbf{X}_k, N_k | \mathbf{X}_{k-1}, N_{k-1})}{q(\mathbf{X}_k, N_k | \mathbf{X}_{k-1}, N_{k-1}, z_k)} \quad (2.59)$$

Then the weights are normalised $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^I \tilde{w}_k^j}$ and the posterior $p(\mathbf{X}_k, N_k | z_{1:k})$ is approximated as

$$p(\mathbf{X}_k, N_k | z_{1:k}) \approx \begin{cases} \sum_{i=1}^I w_k^i \delta(\mathbf{X}_k - \mathbf{X}_k^i) & \text{if } \dim(\mathbf{X}_k) = \dim(\mathbf{X}_k^i) \\ 0 & \text{otherwise} \end{cases} \quad (2.60)$$

To overcome degeneracy, the I joint particles are resampled with replacement based on their weights. This resampling can be performed using any of the techniques described in section 2.5.4. This joint MTT PF, however, suffers from data association and high computational complexity problems. Another PF approach to MTT is to use a separate filter for each target [125], but this could easily lead to filter *hijack* — a problem where more than one filter tracks one of the targets [80, 126]. A more detailed discussion on MTT PF limitations and related work is made in chapter 3.

2.5.6 Choice of importance distribution

For the importance distribution to be optimal [52], it has to be exactly the same as the actual probability distribution. An optimal importance distribution aids in drawing samples from regions of high likelihood. Hence,

$$q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, z_k)_{\text{opt}} = p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, z_k) \quad (2.61)$$

$$= \frac{p(z_k | \mathbf{x}_k^i, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{p(z_k | \mathbf{x}_{k-1}^i)} \quad (2.62)$$

Substituting (2.62) in (2.48)

$$w_k^i \propto w_{k-1}^i p(z_k | \mathbf{x}_{k-1}^i) \quad (2.63)$$

where

$$p(z_k | \mathbf{x}_{k-1}^i) = \int p(z_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i) d\mathbf{x}_k^i \quad (2.64)$$

Numerically evaluating (2.64) is non-trivial.

The analytic expression of the optimal importance distribution function

for Gaussian systems in which the motion model is non-linear and the observation model is linear can be found in [52]. For non-linear Gaussian systems, Orton and Fitzgerald [127] proposed sub-optimal approximations to the optimal importance distribution by using local linearisation techniques. The particles are sampled from a Gaussian approximation of the optimal importance distribution (2.61). If this approximation is reasonable, then the number of redundant particles are reduced so that fewer particles are required.

It is always convenient to choose the Markov transition prior as the importance distribution function. This approach however, is sub-optimal because the prediction is no longer leveraged on the new sensor evidence z_k . Hence

$$q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i) \propto p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i) \quad (2.65)$$

Substituting (2.65) in (2.48) gives

$$w_k^i \propto w_{k-1}^i p(z_k | \mathbf{x}_k^i) \quad (2.66)$$

A few other methods that propose improved importance sampling of particles can be found in [128–132].

2.5.7 Track-before-detect PF

The most common approach to target tracking is based on converting the direct data from a sensor, e.g., infrared camera image, RADAR/SONAR image, signals from phased sensor array, etc., into point measurements of the targets. This conversion is usually done by *thresholding*. This process reduces the amount of information available to the Bayesian filter. Although this information reduction is of little concern at high SNR, thresholding leads to substantial information loss in high noise conditions. This information loss is particularly detrimental while detecting *covert* targets — those targets whose signal components are well hidden under the noise components — for example, stealth aircraft, ships sunk in deep waters, etc. For the detection

of these covert targets, it is important to track them for some time using the unthresholded data (also called *raw sensor observations*) before declaring their presence. The approach to simultaneously detect and initiate tracks using raw sensor data is called *track-before-detect* (TBD). The TBD PF, developed in 2001 by Salmond and Birch [55], is introduced here and will be referred to in chapter 5.

The target state \mathbf{x}_k , for instance, could be as described in (2.1). The target manoeuvres according to the state evolution model in (2.2). Since the target could be present or absent in the observation space, its existence E_k is modelled by a two state Markov chain, such that

$$E_k = \begin{cases} 1 & \text{if target is present} \\ 0 & \text{otherwise} \end{cases} \quad (2.67)$$

Consider the transition probabilities of target birth (a.k.a appearance) (p_b) and death (a.k.a disappearance) (p_d) respectively, to be

$$p_b = \Pr\{E_k = 1 | E_{k-1} = 0\} \quad (2.68)$$

$$p_d = \Pr\{E_k = 0 | E_{k-1} = 1\} \quad (2.69)$$

Then the transitional probability matrix is given by

$$\Pi_S = \begin{bmatrix} 1 - p_b & p_b \\ p_d & 1 - p_d \end{bmatrix}$$

The sensor data z_k could be raw images [55, 56, 133] for visual tracking or phased sensor array signals [134] for acoustic tracking. As described in (2.4) and (2.5) of section 2.2, the formal Bayes' recursion consists of two steps, a) prediction, and b) update. If the posterior $p(\mathbf{x}_{k-1}|z_{1:k-1})$ at time $k-1$ is available, then the prediction (which exists only if $E_k = 1$) is expressed as

$$\begin{aligned} p(\mathbf{x}_k, E_k = 1 | z_{1:k-1}) = & \\ & \int p(\mathbf{x}_k, E_k = 1 | \mathbf{x}_{k-1}, E_{k-1} = 1, z_{1:k-1}) p(\mathbf{x}_{k-1}, E_{k-1} = 1 | z_{1:k-1}) d\mathbf{x}_{k-1} + \\ & \int p(\mathbf{x}_k, E_k = 1 | \mathbf{x}_{k-1}, E_{k-1} = 0, z_{1:k-1}) p(\mathbf{x}_{k-1}, E_{k-1} = 0 | z_{1:k-1}) d\mathbf{x}_{k-1} \end{aligned} \quad (2.70)$$

where

$$\begin{aligned}
 p(\mathbf{x}_k, E_k = 1 | \mathbf{x}_{k-1}, E_{k-1} = 1, z_{1:k-1}) \\
 &= p(\mathbf{x}_k | \mathbf{x}_{k-1}, E_k = 1, E_{k-1} = 1) p(E_k = 1 | E_{k-1} = 1) \\
 &= p(\mathbf{x}_k | \mathbf{x}_{k-1}, E_k = 1, E_{k-1} = 1)(1 - p_d)
 \end{aligned} \tag{2.71}$$

and

$$\begin{aligned}
 p(\mathbf{x}_k, E_k = 1 | \mathbf{x}_{k-1}, E_{k-1} = 0, z_{1:k-1}) \\
 &= p(\mathbf{x}_k | \mathbf{x}_{k-1}, E_k = 1, E_{k-1} = 0) p(E_k = 1 | E_{k-1} = 0) \\
 &= p_b \hat{p}_b(\mathbf{x}_k)
 \end{aligned} \tag{2.72}$$

Here, $\hat{p}_b(\mathbf{x}_k)$ is the initial pdf of the target on its appearance. Then the update step is given by

$$p(\mathbf{x}_k, E_k = 1 | z_{1:k}) = \frac{p(z_k | \mathbf{x}_k, E_k = 1) p(\mathbf{x}_k, E_k = 1 | z_{1:k-1})}{p(z_k | z_{1:k-1})} \tag{2.73}$$

The likelihood function $p(z_k | \mathbf{x}_k, E_k = 1)$, for a Gaussian image model is described in Ch. 11 of [56], and for an array processing model is described in [127].

In the TBD PF recursion of (2.70) and (2.73), each particle \mathbf{x}_k^i is parameterised by its existence E_k^i that is updated every time step using the state transition matrix Π_S . The overall probability of target existence is measured as

$$\hat{E}_k = \frac{\sum_{i=1}^I E_k^i}{I} \tag{2.74}$$

and the target is declared *present* if \hat{E}_k exceeds a certain threshold. The target state estimate is obtained by taking the mean of all the particles for which $E_k^i = 1$.

2.5.8 Historical progress of the PF

Over the years, PFs have been rigorously researched and numerous improvements have been proposed, only a fraction of which are presented here. In an

attempt to emulate the process of drawing samples from an optimal importance distribution, Pitt and Sheppard [135] proposed the Auxiliary PF (APF) that resamples the particles $\{\mathbf{x}_{k-1}^i\}_{i=1}^I$ using the evidence z_k . The importance distribution used in the APF is expressed as

$$q(\mathbf{x}_k, i|z_k) \propto p(z_k|\varrho^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)w_{k-1}^i \quad (2.75)$$

where $\varrho^i \sim p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)$. Then

$$\begin{aligned} q(\mathbf{x}_k, i|z_k) &= q(i|z_k)q(\mathbf{x}_k|i, z_k) \\ &= w_{k-1}^i p(z_k|\varrho^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i) \end{aligned} \quad (2.76)$$

This process mimics to some degree particles being sampled from the optimal importance distribution. Several improvements and modifications to the APF can be found in [110, 136, 137].

After the resampling step that reduces the variance of particles, it is possible that some particles will be exact copies of others. If most particles are identical, the *richness* of the particles in describing the posterior deteriorates. If one or more Markov chain Monte Carlo (MCMC) [138] steps are performed [127], whose invariant distribution is the posterior of interest, then the particles can be made disparate so that they better represent the posterior. Another method that improves the richness and diversity of the particles is the regularised PF (RPF) [110] that resamples particles from a continuous distribution of $p(\mathbf{x}_k|z_{1:k})$. The posterior approximation is

$$p(\mathbf{x}_k|z_{1:k}) \sim \sum_{i=1}^I w_k^i K_{\bar{h}}(\mathbf{x}_k - \mathbf{x}_k^i) \quad (2.77)$$

where

$$K_{\bar{h}}(\mathbf{x}) = \frac{1}{\bar{h}^{n_x}} K\left(\frac{\mathbf{x}}{\bar{h}}\right) \quad (2.78)$$

is a positive regularisation kernel function, n_x is the dimension of the state vector and \bar{h} is the Kernel bandwidth. The unscented PF (UPF) [139] uses an unscented transform for importance sampling; this allows the predicted

particles to move into regions of high likelihood by leveraging them on the current observation z_k . A combined version of UPF and RPF, called the unscented regularised PF (URPF) [140] uses unscented filtering for prediction and kernel density for resampling.

The main drawback of the PF is that sampling in high dimensional spaces can be inefficient. Marginalising some of the variables is an example of the technique called Rao Blackwellisation [141]. A Rao-Blackwellised (a.k.a marginalised) PF [142] exploits linear Gaussian sub-structure present in the model, and divides the components of the state accordingly. Suppose the components of \mathbf{x}_k are divided into two groups \mathbf{x}_k^l and \mathbf{x}_k^n such that the state components in \mathbf{x}_k^l follow a linear Gaussian model; then \mathbf{x}_k^l could be marginalised out in the posterior pdf as

$$p(\mathbf{x}_{1:k} | z_{1:k}) = p(\mathbf{x}_k^l, \mathbf{x}_k^n | z_{1:k}) = p(\mathbf{x}_k^l | \mathbf{x}_{1:k}^n, z_{1:k})p(\mathbf{x}_{1:k}^n | z_{1:k}) \quad (2.79)$$

The estimate of the first term in the right hand side (RHS) of (2.79) can be evaluated using a Kalman filter, and the second term can be approximated using a PF. The estimated pdf can now be represented by a weighted sum of Gaussians, where each particle has a Gaussian distribution attached to it [143]. This approach reduces the amount of computation required when filtering in a high dimensional space. The MPFs have found use in fault diagnosis in aerospace systems [144], terrain aided navigation [145, 146], estimating tyre radii and vehicle trajectory [6], and tracking of heart-beat and respiratory rate [9, 147].

The multiple model PF (MMPF) [148, 149] is a wider range of hybrid PF that allows non-linear modelling of distinctive parameters in the target state vector, e.g., a continuous-value part (target dynamics) and a discrete-valued part (target intensity or measurement association). The TBD PF described in section 2.5.7 belongs to the class of MMPFs.

The PF has been widely used in several real-time array processing applications, examples of which include: localisation of remotely operated underwa-

ter vehicles using SONAR [150], bearings-only tracking using SONAR [151], TBD filtering using RADAR [152, 153], 3D indoor tracking using RADAR [154], detecting voice activity [155], audio source separation [156], detecting seismic activity [157], catching shovelnose guitarfish [2], tracking ballistic missiles [158] and robot assisted heart diagnosis [159]. In spite of being computationally expensive [45, 46, 160], the PF is regarded as a *well established* Bayes' filter and still attracts great attention from the research community.

2.6 The probability hypothesis density filter

For MTT applications that require tracking of *many* targets, e.g., group target tracking¹¹, tracking only targets of interest in high target density, etc., the theoretically optimal approach is the Bayes' filter. However, it is known from section 2.3 that the MTT Bayes' filter is computationally challenging and hence is usually approximated. A Bayes' approximation that overcomes the predicament of high computational complexity in the presence of many targets is the probability hypothesis density (PHD) filter. The PHD filter provides a systematic approach to Bayes' MTT using finite set statistics (FISST) on a *random finite set* (RFS) theoretic framework. While the approximation of $p(\mathbf{x}_k|z_{1:k})$ by its second order moment results in the Kalman filter, the approximation of the joint multi-target posterior $p(\mathbf{X}_k|z_{1:k})$ by its first moment results in formation of the PHD [26]. The time-recursion of the PHD using sensor evidence is called PHD filtering.

In this section, RFS modelling is first introduced in section 2.6.1. After describing the PHD formation process in section 2.6.2, the PHD filter operation is outlined in section 2.6.3. Finally, the historical developments in PHD filtering are discussed in section 2.6.4.

¹¹Group tracking aims to detect and track the targets as well as their formation, e.g., naval convoys.

2.6.1 The RFS model

In a single target tracking system, the state of the target x_k is a vector with *fixed* dimension. This x_k is a *hypothesis* of a single target and is modelled as a random variable, i.e., x_k is any subset of the single target space \mathfrak{X} . In the MTT system, the multi-target joint state X_k is again a vector but with *varying* dimension; this is because the number of targets in the MTT scenario varies with time. The uncertainty in the MTT system is modelled using this joint state X_k . Hence X_k is a *hypothesis* of *all* the targets and is modelled as a random joint state, i.e., a *finite* number of individual target state vectors (chosen *randomly*) from the single target space \mathfrak{X} are collected into a single *vector* — now called the joint state vector X_k . Hence X_k is any subset of the joint space \mathfrak{X}^* . Recalling (2.12) from section 2.3.2, the joint target state is expressed as

$$X_k = (x_k^1, x_k^2, \dots, x_k^{N_k}) \quad (2.80)$$

In contrast, the PHD filter models the uncertainty in the MTT system using a multi-target state *set* Ξ_k expressed as

$$\Xi_k = \{x_k^1, x_k^2, \dots, x_k^{N_k}\} \quad (2.81)$$

The elements in Ξ_k are random instantiations from the single target state space and are finite in number — hence Ξ_k is also called a RFS. In other words, a *finite* number of individual target state vectors (chosen *randomly*) from the single target space \mathfrak{X} are collected into a single *set* — now called the RFS Ξ_k . Ξ_k is a *hypothesis* of *all* the targets and the targets are assumed to be statistically independent and indistinguishable from one another.

Additionally, the PHD filter also models the uncertainty in the observations as a RFS, that is to say the PHD filter is designed to operate *only* on observations that are noisy point target detections of the targets. Recalling (2.20) from section 2.3.3, the observation model for the PHD filter is

$$Z_k = \{z_k^1, \dots, z_k^{M_k}\} \quad (2.82)$$

The RFS prediction model

If the RFS at time $k - 1$ is

$$\Xi_{k-1} = \{x_{k-1}^1, \dots, x_{k-1}^{N_{k-1}}\} \quad (2.83)$$

then the predicted RFS at time k contains surviving targets, targets spawned¹² by surviving targets and new-born targets. This is expressed as

$$\Xi_k = \underbrace{\Xi(x_{k-1}^1), \dots, \Xi(x_{k-1}^{N_{k-1}})}_{\text{Surviving or disappearing targets}} \cup \underbrace{\Psi(x_{k-1}^1), \dots, \Psi(x_{k-1}^{N_{k-1}})}_{\text{Spawned targets}} \cup \underbrace{\Psi_0}_{\text{New-born targets}} \quad (2.84)$$

Here, the prediction $\Xi(x_{k-1}^n)$ of the n th element in Ξ_{k-1} is such that x_{k-1}^n either survives into Ξ_k with probability $p_s(x_{k-1}^n)$ or disappears with probability $1 - p_s(x_{k-1}^n)$. If x_{k-1}^n survives into Ξ_k , then it undergoes a state update as $\Xi(x_{k-1}^n) = \{x_k^n \xleftarrow{f_k} x_{k-1}^n\}$, where f_k mentioned in (2.2) in section 2.1, is the Markovian state evolution function. Hence $x_k^n \sim p(x_k | x_{k-1})$. On the other hand, if x_{k-1}^n disappears, then $\Xi(x_{k-1}^n) = \emptyset$. Hence $\Xi(x_{k-1}^n)$ is either an empty set or a singleton.

The RFS $\Psi(x_{k-1}^n)$ corresponds to the set of targets spawned by x_{k-1}^n . Hence its cardinality is $|\Psi(x_{k-1}^n)| \geq 0$ and its joint distribution is $b(X_k | x_{k-1}^n)$. The RFS Ψ_0 is the set of new-born targets at time k . The joint distribution of these targets is $b(X_k)$. The numbers of spawned and new-born targets are modelled as Poisson distributed.

The RFS observation model

As mentioned previously, the sensor observations for a PHD filter are the *noisy point measurements* of the targets.

If the predicted RFS is given by

$$\Xi_k = \{x_k^1, \dots, x_k^{N_k}\} \quad (2.85)$$

¹²Spawned targets are those that appear near existing targets, e.g., missile from an aircraft, aircraft from a warship, etc.

then the RFS observation model contains target detections, false alarms and clutter points. This can be summarised as

$$\Sigma_k = \underbrace{\{\Sigma(x_k^1), \dots, \Sigma(x_k^{N_k})\}}_{\text{Target detections or misses}} \cup \underbrace{\Theta}_{\text{False alarms and clutter}} \quad (2.86)$$

Here, the observation $\Sigma(x_k^n)$ of the n th element in Ξ_k is such that x_k^n is either detected with probability $p_D(x_k^n)$ or missed with probability $1 - p_D(x_k^n)$. If x_k^n is detected, it undergoes state transition as $\Sigma(x_k^n) = \{z_k \xleftarrow{h_k} x_k\}$, where h_k mentioned in (2.3) in section 2.1, is the observation function. On the other hand, if x_k^n is missed, then $\Sigma(x_k^n) = \phi$.

The set Θ contains false alarms and clutter detections collected by the sensor(s). The number of these non-target detections $|\Theta|$ is modelled as a Poisson process with mean λ_F .

2.6.2 The PHD

The PHD approximates the multi-target joint posterior by its first moment. The PHD function, under low noise conditions [32], is an approximate sufficient statistic to describe the joint posterior. This PHD captures the expected number of targets and their state estimates from the MTT scene.

Recall the RFS in (2.81)

$$\Xi_k = \{x_k^1, x_k^2, \dots, x_k^{N_k}\}$$

The joint distribution $p(X_k)$ (in vector form) can be derived in a RFS framework using the FISST approach [32, 44]. This is expressed as

$$p(\Xi_k) = N_k! p(X_k) \quad (2.87)$$

since the probability density assigned to the N_k element set Ξ_k must be distributed across all possible vector combinations. This also implies that the posterior pdf $p(\Xi_k)$ of the set could be divided by $N_k!$ to make it a pdf.

Here it can be understood that except for the fact that X_k is a vector and Ξ_k is a set, both imply the same meaning and the same distribution in a Bayesian MTT context.

By definition, the first moment of $p(X_k)$ is

$$\hat{X}_k = E[X_k] = \sum_{N_k=0}^{\infty} \int X_k p(X_k) dX_k \quad (2.88)$$

Since evaluation of (2.88), i.e., addition of vectors with unequal sizes, is numerically undefined, a transformation $X_k \rightarrow T_{X_k}$ is chosen such that

$$E[T_{X_k}] = \sum_{N_k=0}^{\infty} \int T_{X_k} p(X_k) dX_k \quad (2.89)$$

This transformation T_{X_k} is selected to be a set of Dirac delta functions [27, 34]. These functions lie in the space of a single target and are concentrated at the individual target states contained in X_k , i.e.,

$$T_{X_k} = \delta_{X_k}(x_k) \triangleq \begin{cases} 0 & \text{if } X = \phi \\ \sum_{b \in X_k} \delta_b(x_k) & \text{otherwise} \end{cases} \quad (2.90)$$

Then the transformed first moment of the joint pdf $p(X_k)$ is

$$D(x_k) \triangleq E[\delta_{X_k}(x_k)] = \sum_{N_k=0}^{\infty} \int \delta_{X_k}(x_k) p(X_k) dX_k \quad (2.91)$$

$D(x_k)$ is not a vector nor a number; it is a *function in the single target space* \mathfrak{X} and termed the PHD [32]. This can be understood by expanding (2.91),

$$\begin{aligned} D(x_k) = & \int \delta_\phi(x_k) d\phi + \int \delta_{x_k^1}(x_k) p(x_k^1) dx_k^1 + \\ & \int \int [\delta_{x_k^1}(x_k) + \delta_{x_k^2}(x_k)] p(x_k^1, x_k^2) dx_k^1 dx_k^2 + \dots \\ & \underbrace{\int \dots \int}_{n \text{ times}} [\delta_{x_k^1}(x_k) + \dots + \delta_{x_k^n}(x_k)] p(x_k^1, \dots, x_k^n) dx_k^1 \dots dx_k^n + \dots \end{aligned} \quad (2.92)$$

For example, if it is hypothesised that there is a single target in the scene, i.e., if $\Xi_k = \{\mathbf{x}_k^1\}$ and the cardinality $|\Xi_k| = 1$, then

$$\begin{aligned} D(\mathbf{x}_k) &= \int \delta_{\mathbf{x}_k^1}(\mathbf{x}_k) p(\mathbf{x}_k^1) d\mathbf{x}_k^1 \\ &= p(\mathbf{x}_k^1 = \mathbf{x}_k) \end{aligned} \quad (2.93)$$

where $p(\mathbf{x}_k^1 = \mathbf{x}_k)$ is the value of the pdf $p(\mathbf{x}_k^1)$ at \mathbf{x}_k .

If it is hypothesised that there are two targets in the scene, i.e., if $\Xi_k = \{\mathbf{x}_k^1, \mathbf{x}_k^2\}$ and the cardinality $|\Xi_k| = 2$, then

$$\begin{aligned} D(\mathbf{x}_k) &= \int \int [\delta_{\mathbf{x}_k^1}(\mathbf{x}_k) + \delta_{\mathbf{x}_k^2}(\mathbf{x}_k)] p(\mathbf{x}_k^1, \mathbf{x}_k^2) d\mathbf{x}_k^1 d\mathbf{x}_k^2 \\ &= \int \int \delta_{\mathbf{x}_k^1}(\mathbf{x}_k) p(\mathbf{x}_k^1, \mathbf{x}_k^2) d\mathbf{x}_k^1 d\mathbf{x}_k^2 + \int \int \delta_{\mathbf{x}_k^2}(\mathbf{x}_k) p(\mathbf{x}_k^1, \mathbf{x}_k^2) d\mathbf{x}_k^1 d\mathbf{x}_k^2 \\ &= \int p(\mathbf{x}_k^1 = \mathbf{x}_k, \mathbf{x}_k^2) d\mathbf{x}_k^2 + \int p(\mathbf{x}_k^1, \mathbf{x}_k^2 = \mathbf{x}_k) d\mathbf{x}_k^1 \\ &= p(\mathbf{x}_k^1 = \mathbf{x}_k) + p(\mathbf{x}_k^2 = \mathbf{x}_k) \end{aligned} \quad (2.94)$$

Hence in general, if it is hypothesised that there are N_k targets in the scene, $\Xi_k = \{\mathbf{x}_k^1, \dots, \mathbf{x}_k^{N_k}\}$, then

$$D(\mathbf{x}_k) = \sum_{n=1}^{N_k} p(\mathbf{x}_k^n = \mathbf{x}_k) \quad (2.95)$$

It is obvious from (2.95) that the PHD is not a pdf because its integral

$$\int D(\mathbf{x}_k) d\mathbf{x}_k = \hat{N}_k \quad (2.96)$$

is the expected number of targets in the scene. It can also be understood that the PHD is a multi-modal function in the space of a single target with peaks near the locations of the targets.

Fig. 2.5 illustrates the PHD formation in a two target case using a simple numerical example. It can be seen that the PHD casts the MTT problem into the space of a single target. Hence the dimensionality of operation does not increase with N_k .

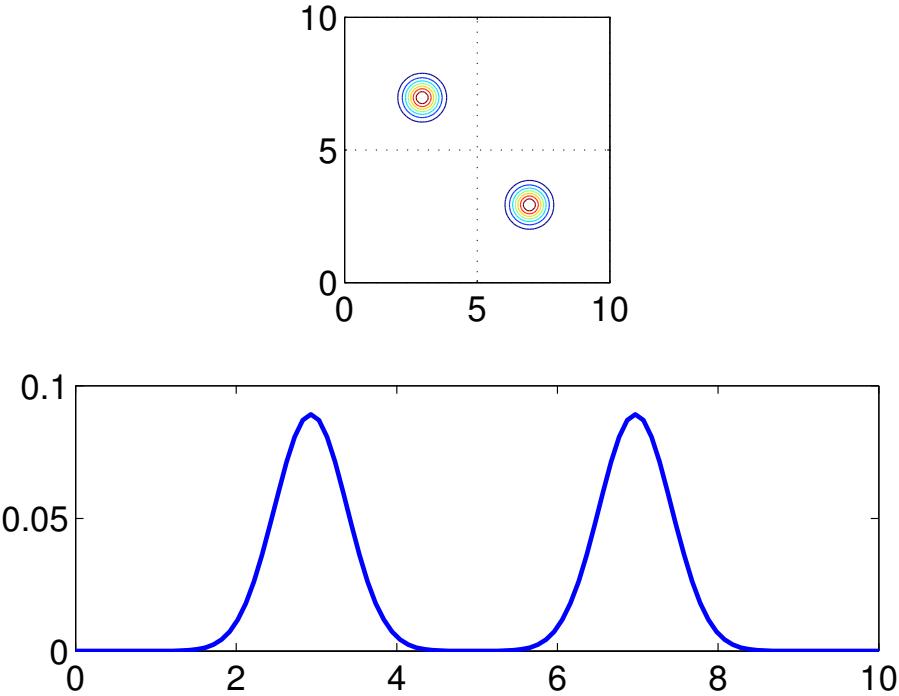


Figure 2.5: Illustration of PHD formation for two targets. Consider the single target space to be 1D $\mathfrak{X} = \mathbb{R}(0, 10]$ and there are two statistically independent targets at 3 and 7. The top figure shows the JMPD $p(\mathbf{X}_k)$ which is bimodal with a peak at $(\mathbf{x}_k^1, \mathbf{x}_k^2) = (3, 7)$ and $(7, 3)$. The bottom figure shows the corresponding PHD function $D(\mathbf{x}_k)$ obtained using (2.91).

2.6.3 The PHD filter recursion

The PHD filter operates by recursing the PHD function in time. The PHD filter recursion is presented in this section. Similar to Bayes' recursion, the PHD filter recurses the PHD function in time using two steps; a) prediction: $D(\mathbf{x}_{k-1}|Z_{1:k-1}) \rightarrow D(\mathbf{x}_k|Z_{1:k-1})$, and b) update: $D(\mathbf{x}_k|Z_{1:k-1}) \rightarrow D(\mathbf{x}_k|Z_{1:k})$. The derivation of the PHD filter [26, 32] is not central to this thesis and hence is not provided here. The PHD recursion is as follows:

The PHD prediction

If the PHD function $D(\mathbf{x}_{k-1}|Z_{1:k-1})$ that approximates the joint posterior $p(\mathbf{X}_{k-1}|Z_{1:k-1})$ at time $k-1$ is known, then the prediction step aims to form a predicted PHD $D(\mathbf{x}_k|Z_{1:k-1})$ that approximates $p(\mathbf{X}_k|Z_{1:k-1})$. This predicted PHD is given by

$$D(\mathbf{x}_k|Z_{1:k-1}) = b(\mathbf{x}_k) + \left[\int \left(p_S(\mathbf{x}_{k-1}) p(\mathbf{x}_k|\mathbf{x}_{k-1}) + b(\mathbf{x}_k|\mathbf{x}_{k-1}) \right) D(\mathbf{x}_{k-1}|Z_{1:k-1}) d\mathbf{x}_{k-1} \right] \quad (2.97)$$

Here, the term $p_S(\mathbf{x}_{k-1}) p(\mathbf{x}_k|\mathbf{x}_{k-1})$ corresponds to targets that survive from time $k-1$ to k with probability p_S . These targets are distributed according to the Markov transition prior. The term $b(\mathbf{x}_k|\mathbf{x}_{k-1})$ corresponds to targets spawned by the surviving targets. $b(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the PHD of the joint distribution of the spawned targets $b(\mathbf{X}_k|\mathbf{x}_{k-1}^n)$ as described in the RFS prediction model in section 2.6.1. The surviving and spawning targets relate to the targets manoeuvring from $k-1$ to k and hence their distributions are integrated over all possible previous states. The term $b(\mathbf{x}_k)$ corresponds to new-born targets at time k . $b(\mathbf{x}_k)$ is the PHD of the joint distribution of new-born targets $b(\mathbf{X}_k)$ as described in the RFS prediction model in section 2.6.1. The number of spawned and new-born targets are modelled as Poisson distributed.

The PHD update

Once the predicted PHD $D(\mathbf{x}_k|z_{1:k-1})$ and sensor evidence Z_k are available, the PHD can be updated according to

$$D(\mathbf{x}_k|Z_{1:k}) \propto \left\{ 1 - p_D(\mathbf{x}_k) + \sum_{m=1}^{M_k} \frac{p_D(\mathbf{x}_k)p(z_k^m|\mathbf{x}_k)}{\lambda_{FC}(z_k^m) + \int p_D(\mathbf{x}_k) p(z_k^m|\mathbf{x}_k) D(\mathbf{x}_k|Z_{1:k-1}) d\mathbf{x}_k} \right\} D(\mathbf{x}_k|Z_{1:k-1}) \quad (2.98)$$

where p_D is the probability of detecting a sensor observation and λ_F is the Poisson mean of the number of false alarm/clutter detections. $p(z_k|x_k)$ is the likelihood function of an observation conditioned on a single target hypothesis. $c(z_k)$ is the false alarm/clutter rate and is typically chosen to be uniform; suppose that \mathfrak{R} is the region that is being observed by the sensor, and let V be its size (area, volume, hypervolume), then $c(z_k) = 1/V$.

It can be seen that the updated PHD is obtained by multiplying the predicted PHD with a PHD pseudo-likelihood. The first term in the RHS of (2.98) is this pseudo-likelihood and corresponds point target detections collected by the sensor. The term

$$1 - p_D(x_k)$$

deals with only the case when no targets are present or the sensor(s) misses all the targets, i.e., $\Sigma_k = \{\phi\} = \{\text{No detections}\}$.

The term

$$\sum_{m=1}^{M_k} \frac{p_D(x_k)p(z_k^m|x_k)}{\int p_D(x_k) p(z_k^m|x_k) D(x_k|Z_{1:k-1}) dx_k}$$

deals with only the case where the sensor(s) detect a few (or all) targets, i.e., $\Sigma_k = \{\Sigma(x_k^1), \dots, \Sigma(x_k^{N_k})\} = \{\text{Target detections}\}$ such that $M_k \leq N_k$.

The term

$$\sum_{m=1}^{M_k} \frac{p_D(x_k)p(z_k^m|x_k)}{\lambda_F c(z_k^m) + \int p_D(x_k) p(z_k^m|x_k) D(x_k|Z_{1:k-1}) dx_k}$$

deals with only the case where the sensor(s) detect a few (or all) targets, and pick up a few false alarm/clutter detections, i.e., $\Sigma_k = \{\Sigma(x_k^1), \dots, \Sigma(x_k^{N_k}) \cup \Theta\} = \{\text{Target detections} \cup \text{False alarms/Clutter}\}$ such that $M_k \geq N_k$.

2.6.4 Historical progress of the PHD filter

Since its inception, the PHD filter has been receiving increasing interest from the research community. The PHD, being a density function in the space of

a single target, can naturally be implemented using a SMC approach. Early work on SMC based PHD filtering can be found in [161–164]; the SMC PHD filter described in [162] is later used in chapters 3 and 4 for comparison purpose. The auxiliary particle PHD filter implementation can be found in [165, 166]. The convergence result for the SMC implementation of the PHD filter is provided in [167, 168].

An important aspect of (SMC) PHD filtering is target state estimation. One approach (that gained much popularity) to this problem, as proposed by Clark and Bell [169] in 2005, is clustering [58]. Zhao et al. [170] developed a partition method that divides the PHD function into sub-PHDs, each which is a single target PHD. The state estimates are then evaluated using these single target PHD functions. More recently Ristic et al. [171] incorporated the state estimation step into the update process of the SMC PHD filter as follows: if the predicted particle set $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{I_k}$ and the sensor observation RFS $\Sigma_k = \{z_k^1, \dots, z_k^{M_k}\}$ is available, then the weights for each element in Σ_k are updated and denoted by $\{w_k^{i,m}\}_{i=1, \dots, I_k, m=1, \dots, M_k+1}$, i.e., the update process results in $M_k + 1$ replicated weighted particle sets. The additional $(M_k + 1)$ th set corresponds to the case when no targets are present. Then the state estimate of the m th element in Σ_k is

$$\hat{\mathbf{x}}_k^m = \sum_{i=1}^{I_k} w_k^{i,m} \mathbf{x}_k^i \quad (2.99)$$

and $\hat{\mathbf{x}}_k^m$ is reported only if $\sum_{i=1}^{I_k} w_k^{i,m}$ is above a certain threshold.

Another important aspect of SMC PHD filtering is track labelling (or continuity), i.e., to be able to identify (or tag) the targets as time progresses. Track labelling is difficult in PHD filtering because the amplitude of the peaks in the PHD function oscillate with time, i.e., some peaks may go flat at certain times. Lin et al. [172] proposed the SMC PHD resolution cell (RC) filter that successfully smoothed out the PHD peak oscillations by dividing the state space into discrete cells and evaluating the PHD of each cell separately. Then the MHT filter (described in section 2.3.3) is applied to

associate peaks to targets. In [173], Panta et al. proposed to feed the SMC PHD filter state estimates as input measurements to the joint filter and used JPDA to associate targets to measurements, which in this context are PHD filter outputs. Clark and Bell [169] used k-means clustering¹³ [58, 59] of labelled particles to achieve track labelling: the process can be summarised in three steps; a) particles are always clustered and so they have a label that identifies the cluster they belong, b) in the update step, particles are resampled along with their labels, and c) after resampling, particles are reclustered and if a majority of the particles in a cluster have the same label, then the clusters are associated. The authors later compared [174] the use of various clustering algorithms, for example, the expectation-maximisation (EM) algorithm [175], for track labelling and found that k-means is computationally more reliable than the others.

Hong et al. [176], were one of the first to focus on hardware implementation of the PHD filter. The elements in the sensor observation RFS Σ_k are time varying and usually large in number. Hence a series hardware implementation of the PHD update step leads to prohibitive latency. Parallel hardware implementation of the PHD update is hard because the number of parallel units (equal to $|\Sigma_k|$) to be initialised every time sample is unknown. The authors proposed a new update model for the SMC PHD filter by arranging the time-varying elements of Σ_k in a series/parallel combination. With a motivation to implement the SMC PHD filter in parallel hardware systems, Shi et al. [177] proposed the particle based observation selection (POSE) PHD filter that opportunistically selects only a *fixed* number of elements in the observation RFS for PHD update processing. The POSE PHD filter was implemented on a Xilinx Vertex-11 Pro field programmable gate array (FPGA) platform.

In order to be able to detect new targets, the birth distribution in the

¹³This procedure will be used when implementing the SMC PHD filter for the comparative analysis of the contributions of this thesis.

SMC PHD filter has to span the *entire* single target space \mathfrak{X} , thus requiring an enormous number of particles. If fewer particles are used such that they do not densely populate the state space, then new-born targets could be undetected. A topic of great interest in this context is to develop efficient SIS procedures to initialise particles near regions of target activity, i.e., to initialise new particles based on observation driven SIS as $b(\mathbf{x}_k) \rightarrow b(\mathbf{x}_k|Z_k)$. Yoon et al. [178] developed an improved SIS algorithm by using the unscented transform technique. Other techniques to improve the SIS procedure in SMC PHD filtering can be found in [179, 180].

In 2005, Vo and Ma [181, 182] implemented the PHD filter using Gaussian mixtures (GMs) [183]. This GM PHD filter has the advantage of providing a closed-form solution to the PHD prediction and update steps. In spite of being limited to linear Gaussian systems, the filter is computationally less expensive than a particle implementation. Moreover, track labelling and continuity can be performed very easily. The convergence result for the GM PHD filter can be found in [184]. The filter has been used for many applications, for example, extended target tracking [185] and SONAR tracking [186]. A more general form of the PHD filter is the cardinalised PHD (CPHD) filter [187]. This filter propagates the PHD and the entire distribution on the target number — the cardinality. As a result, the state estimation can be improved. The CPHD filter can be implemented using GMs [188] and particles [32].

The PHD filter has been widely used in several phased array processing applications, a few examples of which include: autonomous navigation of underwater vehicles using SONAR [189, 190], tracking in far-field array processing [191], tracking moving vehicles using acoustic sensing [192], source separation for acoustic signal processing in reverberant environments [193], multiple input multiple output (MIMO) RADAR tracking [194], bistatic RADAR tracking [51], vehicular traffic monitoring [195], etc. However, since the observation model for the PHD filter is a RFS comprising of noisy point

detections, these applications first convert the phased array data into image models and then preprocess the image to obtain the observation RFS. One of the goals of this thesis is to perform Bayesian MTT without these preliminary data processing stages.

2.7 The multi-Bernoulli filter

The intractability present in the MTT Bayes' joint filter due to high dimensional integration makes the filter computationally demanding for MTT problems. This intractability was overcome by the RFS based PHD approximation presented in section 2.6. Another RFS based MTT Bayes' filter of increasing interest is the multi-target multi-Bernoulli (MeMBer) filter [33] that approximates the JMPD using the parameters of a multi-Bernoulli¹⁴ RFS. While the PHD filter can operate *only* on a RFS observation model, the MeMBer filter can operate on an observation model which is: (i) a RFS — resulting in the “MeMBer filter for RFS observations [31–33],” or (ii) an image — resulting in the “MeMBer filter for image observations [49, 196, 197].” The MeMBer filter for RFS observations was first proposed in 2007 by R. P. S. Mahler [32]. But the filter had a incorrect Taylor series linearisation in the update step which was later corrected in 2007 [33] by Vo et al. The complete results were conclusively published in 2009 [31]. The term “MeMBer filter” in this thesis refers to the corrected approach [31]. Vo et al. further developed the MeMBer filter for image observations [49, 196]. The MeMBer filter for image observations differs from the MeMBer filter for RFS observations in only the update step. The focus of this thesis is on “MeMBer filter for image observations” and hence only its operation is discussed here.

In this section, the idea of Bernoulli approximation is first introduced in section 2.7.1. Then the image observation model is presented in section 2.7.2.

¹⁴Bernoulli statistics was invented and named after the Swiss mathematician Jacob Bernoulli who contributed extensively to the field of calculus and probability.

This is followed by the MeMBer filter recursion in section 2.7.3. Finally the historical development of MeMBer filtering is presented in section 2.7.4.

2.7.1 The multi-Bernoulli approximation

A Bernoulli set Ξ in the single target space \mathfrak{X} has a probability $1 - \epsilon$ of being a null set, and has a probability ϵ of containing a single element x that is distributed according to a pdf $p(\cdot)$. The pdf of Ξ , which is a hypothesis of a single target, is

$$p(\Xi) = \begin{cases} 1 - \epsilon & \text{if } \Xi = \emptyset \\ \epsilon p(x) & \text{if } \Xi = \{x\} \\ 0 & \text{otherwise} \end{cases} \quad (2.100)$$

A multi-Bernoulli RFS Ξ can be considered as the union of a fixed number of *independent* Bernoulli sets, such that

$$\Xi = \bigcup_{n=1}^N \Xi^n \quad (2.101)$$

Ξ is a hypothesis of N targets. The n th Bernoulli set Ξ^n is described by two parameters; the existence probability ϵ^n and the probability density $p^n(x)$. The number of Bernoulli sets N in Ξ is modelled as a random number — hence the approach is regarded as *multi-Bernoulli RFS filtering*. The MeMBer filter recursively propagates these Bernoulli parameters and hence is considered as a parameterised approximation to the sequential Bayes' filter. The JMPD at time k can then be approximated according to

$$p(\mathbf{X}_k | z_{1:k}) \approx \{\epsilon_k^n, p^n(x_k)\}_{n=1}^{N_k} \quad (2.102)$$

where N_k is the time-varying number of Bernoulli sets contained in Ξ_k at time k . The PHD function for $p(\mathbf{X}_k | z_{1:k})$ can be constructed using the Bernoulli parameters as

$$D(\mathbf{x}_k | z_{1:k}) = \sum_{n=1}^{N_k} \epsilon_k^n p^n(\mathbf{x}_k) \quad (2.103)$$

2.7.2 The image observation model

Since this thesis focusses on the MeMBer filter for image observations, the image sensing model and the procedure to construct its likelihood function is described here. The targets are assumed to be rigid bodies that do not overlap. The difference between overlapping and non-overlapping targets is illustrated in Fig. 2.6.

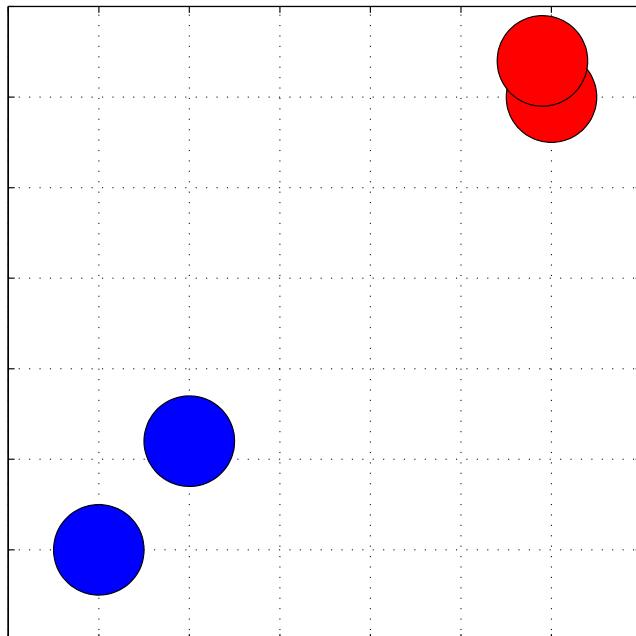


Figure 2.6: Illustration of overlapping (in red) and non-overlapping (in blue) targets in a image observation.

These targets contribute to the *illumination* (or brightness) of a certain region \mathfrak{R}_o within the image z_k , i.e., the region of influence $\mathfrak{R}_o(x_k)$ of a single target with state x_k will contain a set of pixels ρ whose centers lie within $\mathfrak{R}_o(x_k)$. This is illustrated in Fig. 2.7. The likelihood function of the (i, j) th pixel of

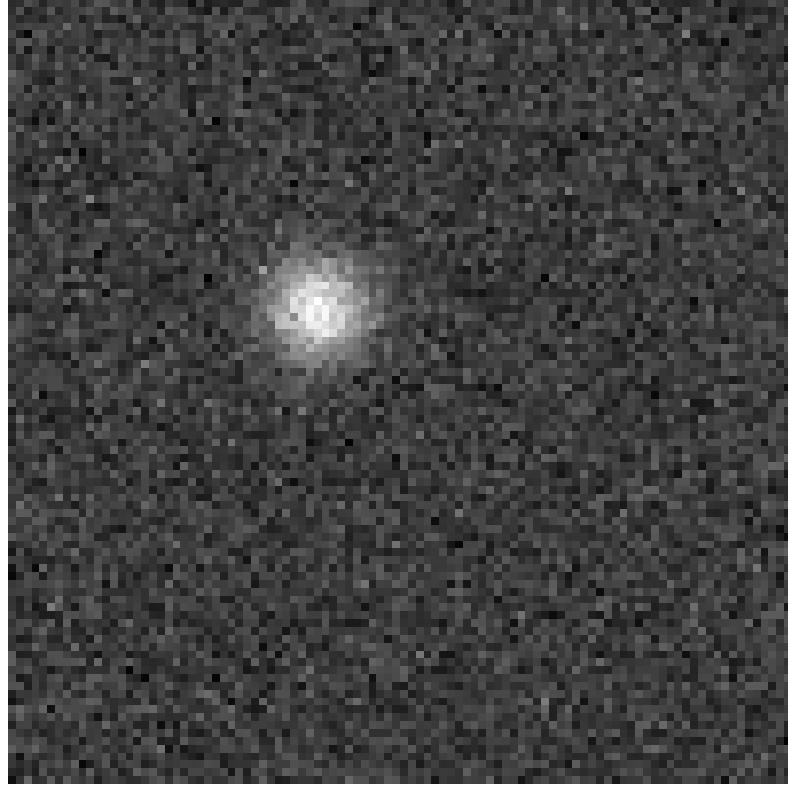


Figure 2.7: A 100×100 image. A single target exactly at the (40×40) th pixel illuminates a set of pixels within its region of influence — $\rho \in \mathfrak{R}_o$.

z_k in the presence of a single target state x_k is

$$p(z_k^{(i,j)} | x_k) = \begin{cases} p_{S+N}(z_k^{(i,j)}, x_k) & \text{if } (i, j) \in \mathfrak{R}_o(x_k) \\ p_N(z_k^{(i,j)}) & \text{if } (i, j) \notin \mathfrak{R}_o(x_k) \end{cases} \quad (2.104)$$

where $p_N(z_k^{(i,j)})$ is the background noise and $p_{S+N}(z_k^{(i,j)}, x_k)$ is the likelihood of the target signal in the presence of background noise.

Assuming the pixels are independent, the likelihood of the multi-Bernoulli

set Ξ_k is

$$\begin{aligned}
 p(z_k | \Xi_k) &= \left(\prod_{\vartheta \in \Xi_k} \prod_{(i,j) \in \mathfrak{R}_o(\vartheta)} p_{S+N}(z_k^{(i,j)}, \vartheta) \right) \left(\prod_{(i,j) \notin \mathfrak{R}_o(\vartheta)} p_N(z_k^{(i,j)}) \right) \\
 &= \left(\prod_{\vartheta \in \Xi_k} \prod_{(i,j) \in \mathfrak{R}_o(\vartheta)} p_{S+N}(z_k^{(i,j)}, \vartheta) \right) \left(\prod_{(i,j) \notin \mathfrak{R}_o(\vartheta)} p_N(z_k^{(i,j)}) \right) \frac{\prod_{(i,j) \in \mathfrak{R}_o(\vartheta)} p_N(z_k^{(i,j)})}{\prod_{(i,j) \in \mathfrak{R}_o(\vartheta)} p_N(z_k^{(i,j)})} \\
 &= \left(\prod_{\forall_i, \forall_j} p_N(z_k^{(i,j)}) \right) \left(\prod_{\vartheta \in \Xi_k} \prod_{(i,j) \in \mathfrak{R}_o(\vartheta)} \frac{p_{S+N}(z_k^{(i,j)}, \vartheta)}{p_N(z_k^{(i,j)})} \right) \\
 &= n(z_k) \prod_{\vartheta \in \Xi_k} \bar{s}_{z_k}(\vartheta)
 \end{aligned} \tag{2.105}$$

where

$$n(z_k) \triangleq \prod_{\forall_i, \forall_j} p_N(z_k^{(i,j)}) \tag{2.106}$$

is the noise component, and

$$\bar{s}_{z_k}(\vartheta) \triangleq \prod_{(i,j) \in \mathfrak{R}_o(\vartheta)} \frac{p_{S+N}(z_k^{(i,j)}, \vartheta)}{p_N(z_k^{(i,j)})} \tag{2.107}$$

is the likelihood ratio of the single target hypothesis $\vartheta \in \Xi_k$.

Considering a Gaussian distributed model (as used¹⁵ in Ch. 11 of [56], the signal and noise components in (2.104) respectively will be

$$p_{S+N}(z_k, \vartheta) = \mathcal{N}(\gamma_\vartheta, \sigma^2) \tag{2.108}$$

$$p_N(z_k) = \mathcal{N}(0, \sigma^2) \tag{2.109}$$

The hypothesis measure γ_ϑ (which is a image) is obtained, as from (2.3) in section 2.1, according to

$$\gamma_\vartheta = h_k(\vartheta) \tag{2.110}$$

Then the likelihood ratio in (2.107) becomes (derivation not provided here)

$$\bar{s}_{z_k}(\vartheta) = \prod_{(i,j) \in \mathfrak{R}_o(\vartheta)} \exp \left(-\frac{\gamma_\vartheta^{(i,j)} (\gamma_\vartheta^{(i,j)} - 2z_k^{(i,j)})}{2\sigma^2} \right) \tag{2.111}$$

¹⁵The image model used in [56] will be referred to and used in chapter 4 of this thesis.

2.7.3 The MeMBer filter recursion

Similar to Bayes' recursion, the MeMBer filter recursively operates in two steps; a) prediction: $\{\epsilon_{k-1}^n, p(\mathbf{x}_{k-1}^n)\}_{n=1}^{N_{k-1}} \rightarrow \{\hat{\epsilon}_k^n, \hat{p}(\mathbf{x}_k^n)\}_{n=1}^{N_k}$, and b) update: $\{\hat{\epsilon}_k^n, \hat{p}(\mathbf{x}_k^n)\}_{n=1}^{N_k} \rightarrow \{\epsilon_k^n, p(\mathbf{x}_k^n)\}_{n=1}^{N_k}$. The derivation of the MeMBer filter [31, 32, 196] is not central to this thesis and hence is not provided here. The MeMBer recursion is as follows:

The MeMBer prediction

Consider that the multi-Bernoulli parameters that describe the multi-target posterior $p(\mathbf{X}_{k-1}|z_{1:k-1})$ at time $k-1$ are known. This multi-Bernoulli approximation at time $k-1$ is given by

$$p(\mathbf{X}_{k-1}|z_{1:k-1}) \approx \{\epsilon_{k-1}^n, p^n(\mathbf{x}_{k-1}^n)\}_{n=1}^{N_{k-1}} \quad (2.112)$$

In the prediction step, the MeMBer filter propagates the previous multi-Bernoulli set $\{\epsilon_{k-1}^n, p^n(\mathbf{x}_{k-1}^n)\}_{n=1}^{N_{k-1}}$ and also appends a new multi-Bernoulli set that accounts for new-born targets¹⁶. Then the MeMBer filter approximation to the Bayes' prediction is given by

$$p(\mathbf{X}_k|z_{1:k-1}) \approx \{\hat{\epsilon}_k^n, \hat{p}^n(\mathbf{x}_k)\}_{n=1}^{N_k} \quad (2.113)$$

$$\approx \{\epsilon_{P,k}^n, p_P^n(\mathbf{x}_k)\}_{n=1}^{N_{k-1}} \cup \{\epsilon_{B,k}^n, b_B^n(\mathbf{x}_k)\}_{n=1}^{N_{B,k}} \quad (2.114)$$

Here, the filter initialises a new multi-Bernoulli set of $N_{B,k}$ elements; the n th element of this set is characterised by an existence probability $\epsilon_{B,k}^n$ and a birth pdf $b_B^n(\mathbf{x}_k)$. The number of new Bernoulli sets $N_{B,k}$ is usually modelled as Poisson distributed with mean λ_B . The birth pdfs $\{b_B^n(\mathbf{x}_k)\}_{n=1}^{N_{B,k}}$ correspond to the targets appearing at time k . If the information about regions of target appearance is not available, then these pdfs are chosen to be uniform. The initialisation of the existence probabilities $0 < \epsilon_{B,k}^n \leq 1$ for $n = 1, \dots, N_{B,k}$ depends on the knowledge about the location of new-born targets. If this knowledge is unavailable, then the values could be initialised randomly.

¹⁶The MeMBer filter does not take target spawning into consideration.

The persisting multi-Bernoulli parameters $\epsilon_{P,k}^n, p_P^n(\mathbf{x}_k)$ respectively are the existence probability and track pdf of the n th Bernoulli set that survives from time $k - 1$ to k . These multi-Bernoulli parameters are expressed as

$$\epsilon_{P,k}^n = \epsilon_{k-1}^n \int p_S(\mathbf{x}_{k-1}) \cdot p^n(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1} \quad (2.115)$$

and

$$p_P^n(\mathbf{x}_k) = \frac{\int p_S(\mathbf{x}_{k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}) p^n(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1}}{\int p_S(\mathbf{x}_{k-1}) p^n(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1}} \quad (2.116)$$

Here $p_S(\cdot)$ is the survival probability density of the n th Bernoulli set and $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the Markov transition prior. (2.115) gives the existence probability of the n th Bernoulli set that persists from time $k - 1$ to k with a survival chance of $p_S(\cdot)$. This is obtained by multiplying ϵ_{k-1}^n by the integral of the dot product of $p_S(\cdot)$ and $p^n(\cdot)$. If $p_S(\cdot)$ is chosen to be a constant value p_S , then $\epsilon_{P,k}^n$ is obtained by directly multiplying ϵ_{k-1}^n by p_S . (2.116) gives the predicted distribution of the persisting n th Bernoulli set (surviving into time k with $p_S(\cdot)$) by multiplying the Markov transition prior with its previous pdf and integrating over all possible previous states.

The MeMBer update

Consider that the MeMBer approximation to Bayes' prediction is available and given (from (2.113)) by

$$p(\mathbf{X}_k | z_{1:k-1}) \approx \{\hat{\epsilon}_k^n, \hat{p}^n(\mathbf{x}_k)\}_{n=1}^{N_k} \quad (2.117)$$

then the MeMBer approximation to Bayes' posterior is

$$p(\mathbf{X}_k | z_{1:k}) \approx \{\epsilon_k^n, p^n(\mathbf{x}_k)\}_{n=1}^{N_k} \quad (2.118)$$

where the updated multi-Bernoulli parameters $\epsilon_k^n, p(\mathbf{x}_k^n)$ are given by

$$\epsilon_k^n = \frac{\hat{\epsilon}_k^n \int \bar{s}_{z_k}(\mathbf{x}_k) \cdot \hat{p}^n(\mathbf{x}_k) d\mathbf{x}_k}{1 - \hat{\epsilon}_k^n + \hat{\epsilon}_k^n \int \bar{s}_{z_k}(\mathbf{x}_k) \cdot \hat{p}^n(\mathbf{x}_k) d\mathbf{x}_k} \quad (2.119)$$

and

$$p^n(x_k) = \frac{\bar{s}_{z_k}(x_k) \hat{p}^n(x_k)}{\int \bar{s}_{z_k}(x_k) \hat{p}^n(x_k) dx_k} \quad (2.120)$$

Here, the term $\bar{s}_{z_k}(x_k)$, defined in (2.107), is the likelihood ratio of the single target hypothesis corresponding to the n th predicted Bernoulli set. (2.120) gives the updated pdf of the n th Bernoulli set; this is obtained by the Bayes' update step — by multiplying the predicted pdf with the likelihood ratio and normalising the result. (2.119) gives the measurement updated existence probability that the n th predicted Bernoulli set contains an actual target. For a given target hypothesis, (2.119) can be written as

$$\epsilon_k^n = \frac{\bar{s}_{z_k} \hat{\epsilon}_k^n}{1 - \hat{\epsilon}_k^n + \bar{s}_{z_k} \hat{\epsilon}_k^n} \quad (2.121)$$

where \bar{s}_{z_k} is the likelihood value of the hypothesis. A large likelihood value indicates that the prediction is correct, hence ϵ_k^n will be nearly the same as $\hat{\epsilon}_k^n$. Conversely, a low likelihood value indicates that the prediction is incorrect, hence ϵ_k^n will be lower than $\hat{\epsilon}_k^n$. The change in ϵ_k^n from $\hat{\epsilon}_k^n$ for various values of \bar{s}_{z_k} can be seen in Fig. 2.8.

Pruning and merging

It is obvious from (2.114) that the MeMBER filter generates a large multi-Bernoulli RFS in which; a) most of the Bernoulli parameters do not correspond to an actual target, and/or b) multiple Bernoulli parameters correspond to the same target. In either case, these Bernoulli parameters are redundant. Hence a few tracks¹⁷ should be eliminated (or pruned). If the existence probability ϵ_k^n of the n th hypothesised track is less than a threshold $\mathcal{T}_{\text{prune}}$, the track is eliminated. Moreover, merging tracks that are close to one another aids in reducing the filter computation. This is done as follows: two Bernoulli sets $\{\epsilon_k^i, p^i(x_k)\}$ and $\{\epsilon_k^j, p^j(x_k)\}$ can be merged if their probability

¹⁷A track refers to a single Bernoulli set parameters.

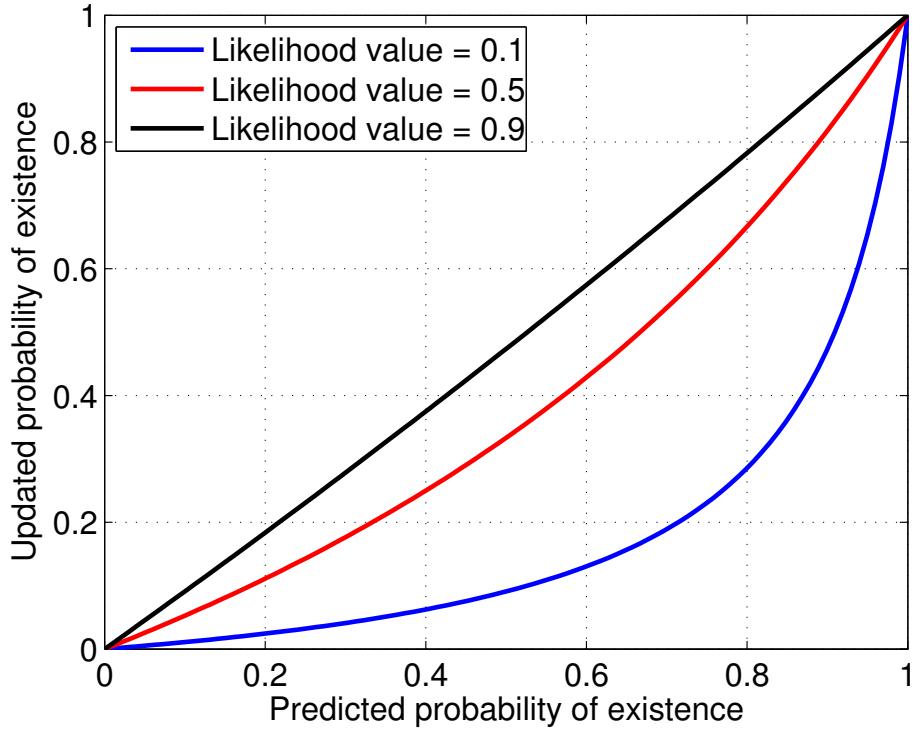


Figure 2.8: The predicted existence probability versus the updated existence probability of a Bernoulli set, for various likelihood values.

of association $P_{i,j}$, given by

$$P_{i,j} = \int p^i(x_k) \cdot p^j(x_k) dx_k \leq 1 \quad (2.122)$$

exceeds a chosen threshold. The higher the proximity of two tracks i, j , the larger their $P_{i,j}$, and vice-versa. Then the tracks are merged according to

$$\epsilon_k^{i,j} = \epsilon_k^i + \epsilon_k^j \quad (2.123)$$

and

$$p^{i,j}(x_k) = \frac{p^i(x_k) \cdot p^j(x_k)}{P_{i,j}} \quad (2.124)$$

2.7.4 Historical progress of the MeMBer filter

Here, a brief review of progress in the MeMBer filter is presented.

MeMBer filtering for RFS observations: The MeMBer filter originally proposed in 2007 by R. P. S. Mahler [32] was for a RFS observation model in low clutter scenarios. The derivation for the update step in [32] had an incorrect Taylor series linearisation which resulted in it over-estimating the actual number of targets (the cardinality). This was later corrected in 2007 by Vo et al. [33] and the approach was termed the cardinality balanced MeMBer (CB MeMBer) filter. The results were published comprehensively in 2009 in [31]. The Bernoulli approach that approximates Bayes' posterior using the parameters of a single Bernoulli set is called the *Bernoulli filter* [198] and is a single target tracker.

The MeMBer filter can be implemented using particles [33, 196] (the SMC MeMBer filter described in [196] is later used in chapters 3 and 4 for comparison purpose) or Gaussian mixtures [31]. The convergence result of the SMC MeMBer filter is reported in [199]. While a closed-form solution for the SMC MeMBer filter is not possible, the GM MeMBer filter provides analytic approximation to Bayes' filtering under linear Gaussian conditions. In 2010, Yin et al. proposed the Guassian particle MeMBer (GP MeMBer) filter [200] in which each pdf in the multi-Bernoulli set is approximated using a set of Gaussians. The novelty in this approach is that the SMC is integrated into the prediction and update of the existence probabilities and pdfs of the multi-Bernoulli set. This procedure aids in the use of non-linear motion and observation models under Gaussian conditions. However, the resulting approach no longer has a closed-form solution. More work on Gaussian particle and similar implementations can be found in [201, 202]. Their use in RFS filtering can be found in [203].

It is known that probabilistic graphical models can be used for inference [204] in MTT problems. In 2011, Williams [205] used loopy belief propagation

techniques in a MeMBer filter framework. The resulting filter, termed the belief propagation MeMBer (BP MeMBer) filter, outperformed the MeMBer filter in terms of its track accuracy at low SNR conditions. In 2012, Williams proposed to unite the operating structure of the PHD and MeMBer filters; the resulting filter is called hybrid PHD-MeMBer filter [206]. The conventional MeMBer filter has to initialise a large number of Bernoulli sets to account for target birth. On the contrary, the hybrid PHD-MeMBer filter uses the PHD filtering strategy of modelling the number of unknown targets to be Poisson in a MeMBer framework. The merits: a) faster track initiation¹⁸ for new-born targets, and b) improved performance using fewer Bernoulli sets.

Ouyang et al. [207] observed that the number of elements in the new-born multi-Bernoulli set will proportionally scale down the existence probabilities of the elements in the persisting multi-Bernoulli set. In other words, the greater the number of new-born tracks $M_{B,k}$, the less will be the existence probability of the n th persisting track $\epsilon_{P,k}^n$. A persisting track with low $\epsilon_{P,k}^n$ could be rejected even though it hypothesises a target correctly, thus leading to the detrimental result that a target detected by the sensor is rejected by the filter. The authors corrected this problem by modifying the update step of the new-born tracks using the missed detection probability.

More recently, robust MeMBer filtering is gaining increasing interest. The conventional MeMBer filter assumes that the clutter intensity, the surveillance region perimeters and the target detection profile are known a priori. However, these parameters are usually unknown in real time. Robust MeMBer filtering provides a new formulation for the prediction and update of MeMBer parameters such that the filter attains the ability to perform even when the aforestated parameters are unknown or partially known. Related work can be found in [208–210].

MeMBer filtering for image observations: The MeMBer filter for im-

¹⁸Track initiation refers to the declaration of the presence of a target

age observations [49, 196] has been used primarily for visual tracking [211]. In 2012, Hoseinnezhad et al. [212] developed a likelihood equation for the sensor model in which the observation is an ensemble of images, each containing a single target. The authors reported the MeMBer filter performance in tracking multiple football players in a single image. The method gives good results for numerous targets with similar visual pattern. More work on visual tracking can be found in [211]. A TBD MeMBer filter approach for detecting and tracking targets moving along an interconnected network of roads is presented in [213].

In 2011, Hoseinnezhad et al. [214] used MeMBer filtering for real time talker localisation and tracking in a audio visual observation environment. The input observation to the filter is the information from both video and audio. The authors demonstrated that the filter tracks effectively even when either of the inputs is unavailable — for example, the talkers go out of the image frame or stop talking. That said, the MeMBer filter does not possess the mathematical framework to operate *directly* on acoustic data (or for that case, any phased sensor array data). Hence in [214], the acoustic signals are fed into the MeMBer filter after converting them into direction-of-arrival estimates assuming a far-field sensing scenario. One of the goals of this thesis is to operate the MeMBer filter more directly on phased arrays.

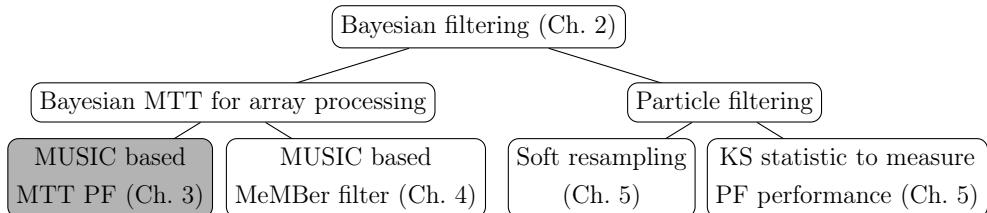
2.8 Summary

This chapter provided a mathematical overview of state-of-the-art Bayesian filtering techniques that are of interest to this thesis. Firstly, the state space modelling process and the single/multiple target Bayes' filtering was described. Then the contemporary Bayes' filters — the Kalman filter, the SMC method, the PHD filter and the MeMBer filter were described. The historical progress of these filters was also presented. To overcome the many drawbacks of these contemporary filters is the focus of this thesis. Hereafter, the contributions of this thesis are presented.

2.8. SUMMARY

3

The MUSIC based MTT PF



The primary interest of this thesis is in Bayesian multiple target tracking (MTT) [11, 35] using phased sensor arrays. The Bayes' filter aids in MTT [1, 2] by recursively building a posterior probability distribution function (pdf) of the target states using the sensor data. In the context of array processing, this data is the noisy signals impinging on the sensor array [38]. However as described in section 2.3 of chapter 2, the Bayes' filtering is computationally intractable and this problem is overcome using approximate solutions. The approximation technique which is central to this thesis is the particle filter (PF) [21, 52].

One approach to tackle the MTT problem is to use the Bayes' joint filter

[44, 71, 72, 124] described in section 2.3 of chapter 2. Another approach is to use the random finite set (RFS) filters — the probability hypothesis density (PHD) filter [26, 32] described in section 2.6 of chapter 2, or the multi-target multi-Bernoulli (MeMBer) filter [31, 32, 196] described in section 2.7 of chapter 2. Although these contemporary Bayes’ filters have been successfully used for MTT in array signal processing applications [127, 191], they are not without drawbacks. The joint filter models the uncertainty in the MTT system using a joint target state hypothesis. The major drawback in the joint filter, as identified in problem 1 of section 1.3.1 in chapter 1, is that its computational complexity increases with the number of targets and consequently the use of the joint filter is impeded for tracking many targets. The RFS filters overcome this complexity problem by operating in the dimensionality of a single target space. However, the drawback in RFS filters, as identified in problem 2 of section 1.3.1 in chapter 1, is that they do not possess the mathematical framework to operate directly on phased array data and consequently the array data has to be converted into a suitable format and this conversion leads to information and resolution loss. The research goal 1 in section 1.3.1 of chapter 1 aims to overcome these drawbacks. The goal is to develop a MTT PF that overcomes a) the computational complexity of the joint filter, and b) the information and resolution loss of the RFS filters. This chapter presents the development of this novel MTT PF and illustrates (using simulation results) how the goal is achieved.

The chapter is organised as follows: The drawbacks of state-of-the-art Bayes’ MTT filters are outlined in section 3.1 and the motivation is stated in section 3.2. After presenting the sensor array observation model in section 3.3 and briefly revisiting the single target PF in section 3.4, the proposed MTT PF is presented in section 3.5 and section 3.6. This is followed by the simulation results in section 3.7 and the conclusions in section 3.8.

3.1 Related work

This section outlines the drawbacks and the related work in state-of-the-art Bayes' MTT PF methods. There are currently three main approaches to Bayesian tracking of multiple targets: one approach is to use a separate filter for each target [215], but when implemented using particles [216], the situation can easily arise that more than one filter tracks one of the targets — the *filter hijack problem* [80]. A few techniques have been proposed to overcome this hijack problem, for example, the mixture PF technique [126]. However, the separate filter approach has not received widespread attention because the initialisation or the removal of filters require knowledge of the actual number of targets which is frequently unknown.

The second approach to Bayes' MTT is the joint filter (described in section 2.3 of chapter 2) that aims to construct the joint multi-target probability distribution (JMPD) [70] of the joint target state hypothesis $X_k = (x_k^1, x_k^2, \dots, x_k^{N_k})$, such that the joint target state hypothesis tracks the location of every target [124]. The JMPD addresses the ignorance about the actual number of targets N_k by modelling N_k as a random variable. A few methods proposed to treat this ignorance are; a) the independent partitioning method [127], b) the improved a priori sampling method [217, 218], c) the method of assigning existence parameters to each individual target in the joint state [219], and d) the existence grid method that initiates tracks near the regions of target activity [220, 221]. An overview of these schemes can be found in [222]. A general problem in the JMPD approach when the observations are a set of point target detections is that it is uncertain as to which observation should be associated with which single target hypothesis in the joint state X_k — the *data association problem* [48] described in section 2.3.3 of chapter 2. A few methods proposed to treat the problem are; a) the nearest neighbour method [223], b) the Gibbs sampling [224] unification method [225], c) the Markov Chain Monte Carlo (MCMC) based method [226], d) the feedback-control method [227], e) the game theory based method [228], f) the multiple

hypothesis tracking (MHT) method [229], g) the joint probabilistic data association (JPDA) method [43], and h) the probabilistic multiple hypothesis tracking (PMHT) method [230, 231]. The major challenge of joint filtering, however, is its computational complexity. The increase in the number of targets leads to the curse of dimensionality [45] (described in section 2.3.4 of chapter 2). The dimensionality of the state space increases linearly and consequently, the filtering process becomes computationally expensive. Some methods that overcome this complexity can be found in [47, 127, 232]. The joint filter *resource complexity problem* is still an interesting challenge to the research community.

The third approach to Bayes' MTT is to use RFS filters — the PHD [26, 32, 233] or the MeMBer [31, 49, 196] filters. This chapter focusses only on the PHD filter and investigation on the MeMBer filter is postponed until chapter 4. The PHD filter described in section 2.6 of chapter 2 operates in the space of a single target thereby avoiding the computational complexity and data association problems prevalent in the joint filter. However, it can be observed in (2.98) that the weight update in the PHD filter is calculable only for sensor evidence which is in the form of a finite set of point target detections: that is to say, the PHD filter does not possess a suitable mathematical framework to directly operate on phased sensor array data. To date, the *direct use of sensor array signals as observations in the PHD filter is restricted*. This problem is usually bypassed by converting the sensor array signals into an image [50, 189] by means of a single target beamformer or fast Fourier transform (FFT) [191], and then pre-processing the image [192] using thresholding or segmentation to obtain the finite set observation feed. This two-stage conversion causes substantial *information loss* at low signal-to-noise (SNR) and *prohibits the resolution* of closely spaced targets.

3.2 Motivation

The aim now is to develop a MTT PF that overcomes the high computational complexity in the joint filter and the information and resolution loss in the PHD filter. The key idea to overcome the complexity problem is to find a function similar to the PHD, which will act as a likelihood function, but the dimensionality of which is the dimensionality of the state of a single target. The Multiple SIgnal Classification (MUSIC) pseudo-spectrum [57] is such a function. MUSIC is one of what are called “super-resolution techniques.” Using MUSIC as a proxy for the likelihood in a PF framework allows the particles to operate in the space of a single target. But this leads to particle hijack — a general MTT problem where all the particles ultimately converge around one target. The key idea to overcome this problem is to proportionally group (or separate) the particles. This can be achieved by *clustering* [59] the particles and regulating the clusters using *soft resampling* [61].

3.3 Phased sensor array model

The primary objective in this chapter is to develop a MTT PF for a phased sensor array observation model [38], the formulation of which is presented in this section.

Signal processing involves the extraction and analysis of information from signals, e.g., speech, images, etc. The interest of this thesis is in *array signal processing* for MTT applications, the goal of which is to extract spatial and temporal information contained in the noise-corrupted signals emitted/reflected from a finite number of sources (a.k.a targets). In other words, the goal here is to estimate parameters of interest, i.e., the number and locations/direction-of-arrivals/range of the targets present in the surveillance region [234]. These targets emit/reflect energy waves (or *signals*), examples of which include radio frequency electro-magnetic waves (for radar appli-

cations) [235], sound waves (for sonar or acoustic applications) [236, 237], seismic waves (for under-earth exploration), etc. A sensor array is a group of sensors distributed spatially according to a known geometry, e.g., linear, planar, circular, etc. Multiple superimposed signals that contain target information are collected at the sensor array by sampling the wavefield generated by the targets. These signals undergo some attenuation, i.e., reduction in signal strength, by the time they reach the sensors. Moreover, since the distance from the target to each sensor in the array is different, the signal recorded at each sensor will be a phase-shifted replica of the other, hence the name *phased sensor arrays*. The sensor array then converts the received energy into an electrical response which can then be processed using signal processing techniques. This thesis considers passive sensor arrays [238], i.e., the signals from the targets are generated by the targets themselves. This thesis also considers a near-field sensing scenario for short range MTT applications. The mathematical model for this sensing model is presented below.

Assume an array of M sensors. Let there be N_k targets at time k and the n th target at location ζ_k^n generates a signal s_k^n . Here, it is assumed that the signals are narrowband and of frequency ω , although the approach can be extended to wideband signals using techniques such as [239]. The sensor model can be summarised as

$$z_k = A_k s_k + v_k \quad (3.1)$$

where z_k is the vector of sensor signals, s_k is the signal vector and the noise vector v_k is complex Gaussian distributed as $v_k \sim \mathcal{CN}(0, R)$. If the noise at each sensor is assumed to be independent, then R is a diagonal matrix with noise variance σ^2 corresponding to each sensor. Assuming near field conditions, A_k is a $M \times N_k$ steering matrix, the n -th column of which (called

the steering vector)

$$a_k^n = \left[\frac{e^{-j \omega \frac{d_{1,n,k}}{\nu}}}{d_{1,n,k}}, \frac{e^{-j \omega \frac{d_{2,n,k}}{\nu}}}{d_{2,n,k}}, \dots, \frac{e^{-j \omega \frac{d_{M,n,k}}{\nu}}}{d_{M,n,k}} \right]^T \quad (3.2)$$

describes the delay and attenuation the signal s_k^n undergoes when it reaches each of the M sensors. ν is the propagation velocity and $d_{m,n,k}$ is the distance between the n th target and the m th sensor at time k . For multiple targets, their signals are added at the sensors.

Consider a target hypothesis X_k generating a signal vector s_k for which the hypothesised steering matrix is A_k . Then the likelihood equation [240] for this model is

$$p(z_k | A_k, s_k) = \frac{1}{\pi^M \det(R)} \exp\left(- (z_k - A_k s_k)^H R^{-1} (z_k - A_k s_k)\right) \quad (3.3)$$

where $(.)^H$ denotes Hermitian transpose. It is obvious that the knowledge about the actual number of targets N_k is required in forming A_k and then computing (3.3). A few approaches to treat this problem in the joint MTT PF have been mentioned in section 3.1. However, the PHD filter that overcomes the computational complexity of the joint filter does not possess a framework to directly evaluate (3.3). This non-provision is bypassed using image models [50, 189].

The objective now is to localise and track the targets at $\zeta_k^1, \dots, \zeta_k^{N_k}$ using the z_k obtained from (3.1).

3.4 The single target PF revisited

The single target PF described in sections 2.5.3 and 2.5.4 of chapter 2 is briefly repeated here for the sake of completeness. The MTT PF proposal presented in subsequent sections then builds on this single target PF.

Target inference is achieved by estimating the target state x_k at time

k using the noisy sensor signals $z_{1:k}$ received until the k th time step. The sequential Bayes' filter aids in this inference by constructing a posterior pdf $p(\mathbf{x}_k|z_{1:k})$. The PF represents the posterior by a set of weighted particles. Here $p(\mathbf{x}_{k-1}|z_{1:k-1})$ is represented by the weighted particles $\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^I$, where i is the particle index and I is the total number of particles. To obtain a representation of $p(\mathbf{x}_k|z_{1:k})$, the PF operates by sampling a new set of particles from the previous ones using the importance distribution as

$$\mathbf{x}_k^i \sim q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, z_k) \quad (3.4)$$

The particle weights are updated according to

$$\tilde{w}_k^i = w_{k-1}^i \frac{p(z_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, z_k)} \quad (3.5)$$

Then the weights are normalised $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^I \tilde{w}_k^j}$ to represent the posterior. To prohibit any one particle having all the weight, the particles are resampled by replacing those particles that have small weights by those that have large weights [53, 122, 241, 242]. Various resampling procedures were outlined in section 2.5.4 of chapter 2.

3.5 Multiple signal classification

Here, the process of target localisation using the non-Bayesian MUSIC [57] is first described and then the proposed MUSIC based MTT PF idea is presented.

3.5.1 Classical MUSIC

The goal of array signal processing is to estimate the target parameters from the signal recorded at the phased sensor array. One of the first methods to address this estimation problem is *beamforming* [243], the idea of which is to steer the array to one location at a time and measure the output power.

Description of various beamforming techniques is available at [244]. Beamformers suffer greatly from resolution problem [38], i.e., the ability to separate spatially close signal emitters (targets). This problem can be overcome using the MUSIC algorithm.

The MUSIC is a sub-space based method that utilises the eigenstructure of the data covariance matrix. Consider that the sensor array signals z_k have been received using (3.1). Let the number of targets be N_k and the number of sensors be M . For convenience in this analysis up to (3.10), the time index k is dropped and the target indices are noted in the sub-script instead of the super-script. The spatial data covariance matrix of z is

$$\begin{aligned}\tilde{R} &= E[z z^H] \\ &= E[(As + v)(As + v)^H] \\ &\stackrel{(a)}{=} E[(As)(As)^H] + E[v v^H] \\ &\stackrel{(b)}{=} E[(As)(As)^H] + \sigma^2 \mathbf{I} \\ &= E[A s s^H A^H] + \sigma^2 \mathbf{I} \\ &= A E[s s^H] A^H + \sigma^2 \mathbf{I} \\ &= A \tilde{P} A^H + \sigma^2 \mathbf{I}\end{aligned}\tag{3.6}$$

where $\stackrel{(a)}{=}$ follows under the assumption that the signals and noise are uncorrelated and $\stackrel{(b)}{=}$ follows under the assumption that the noise at each sensor is independent. Assuming the targets are independent of each another, \tilde{P} in (3.6) is a $N \times N$ diagonal matrix described as

$$\tilde{P} = \begin{bmatrix} \tilde{P}_1 & 0 & . & . & 0 \\ 0 & \tilde{P}_2 & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & \tilde{P}_N \end{bmatrix}\tag{3.7}$$

where

$$\tilde{P}_n \triangleq \mathbb{E}[s_n s_n^H] = \mathbb{E}[|s_n|^2] \quad (3.8)$$

Hence, (3.6) can be written as

$$\tilde{R} = \sum_{n=1}^N \tilde{P}_n a_n a_n^H + \sigma^2 \mathbf{I} \quad (3.9)$$

If e is a steering vector orthogonal to the sub-space defined by the signals s_n for $n = 1, \dots, N$, then

$$\begin{aligned} \tilde{R} e &= \left(\sum_{n=1}^N \tilde{P}_n a_n a_n^H + \sigma^2 \mathbf{I} \right) e \\ &= \sum_{n=1}^N \tilde{P}_n a_n a_n^H e + \sigma^2 e \\ &\stackrel{(a)}{=} \sigma^2 e \end{aligned} \quad (3.10)$$

where $\stackrel{(a)}{=}$ follows because $a_n^H e = 0$. Hence, e is an eigenvector of \tilde{R} and has an eigenvalue of σ^2 . However, if e lies in the sub-space defined by the signals s_n for $n = 1, \dots, N$, then it will still be an eigenvector of \tilde{R} but will have an eigenvalue greater than σ^2 .

Eigendecomposing the space of the data covariance matrix into dominant and sub-ordinate components using its eigenvalues reveals which of its sub-space is defined by the signals and which by the noise. The dominant sub-space, i.e., the vector space spanned by those vectors that capture the maximum possible energy in the observed samples, is called the *signal sub-space*. The eigenvectors corresponding to the largest N eigenvalues of \tilde{R} span the signal sub-space — the sub-space defined by the signals. The remaining $M - N$ eigenvectors span the noise sub-space — the sub-space orthogonal to the signal sub-space.

The MUSIC algorithm operates in the noise sub-space. Hence, after the

eigendecomposition of \tilde{R}_k , the eigenvectors corresponding to the smallest $M - \eta_k$ eigenvalues, represented as

$$U_n = [u_1, \dots, u_{M-\eta_k}] \quad (3.11)$$

are obtained. U_n contains the noise eigenvectors and its columns span the noise sub-space. The MUSIC algorithm assumes that knowledge about N_k is available, hence η_k is equal to N_k (the MTT PF being proposed in this chapter does not make that assumption). If any location $\hat{\zeta}_k$ in the field of view is hypothesised to be a target, its steering vector $a_k^{\hat{\zeta}}$ can be formed using (3.2) and the MUSIC pseudo-spectrum value of $\hat{\zeta}_k$ is given by

$$S_{MU}(\hat{\zeta}_k) = \frac{1}{(a_k^{\hat{\zeta}})^H U_n U_n^H a_k^{\hat{\zeta}}} \quad (3.12)$$

The term $U_n^H a_k^{\hat{\zeta}}$ is the projection of the hypothesised steering vector $a_k^{\hat{\zeta}}$ onto the noise sub-space. Since the signal sub-space and the noise sub-space are orthogonal, the projection (squared to avoid negative numbers) should be zero when $\hat{\zeta}_k$ corresponds to an actual target, and hence its inversion in (3.12) should result in a large value.

Since MUSIC computes only the power density spectrum [245], it is referred to as a *pseudo-spectrum*. Unlike the conventional FFT or beamforming methods, the MUSIC algorithm can be evaluated for any (even closely-spaced) frequencies. Moreover, MUSIC (and other sub-space methods like root MUSIC [246], etc.) offer better spatial resolution than the Rayleigh resolution [247] — a resolution limit attainable by FFT and beamformer methods. Hence MUSIC is considered as a *super-resolution* algorithm.

An example 2D MUSIC pseudo-spectrum is shown in Fig. 3.1. It can be observed that the spectrum gives large narrow peaks at the locations of the three targets.

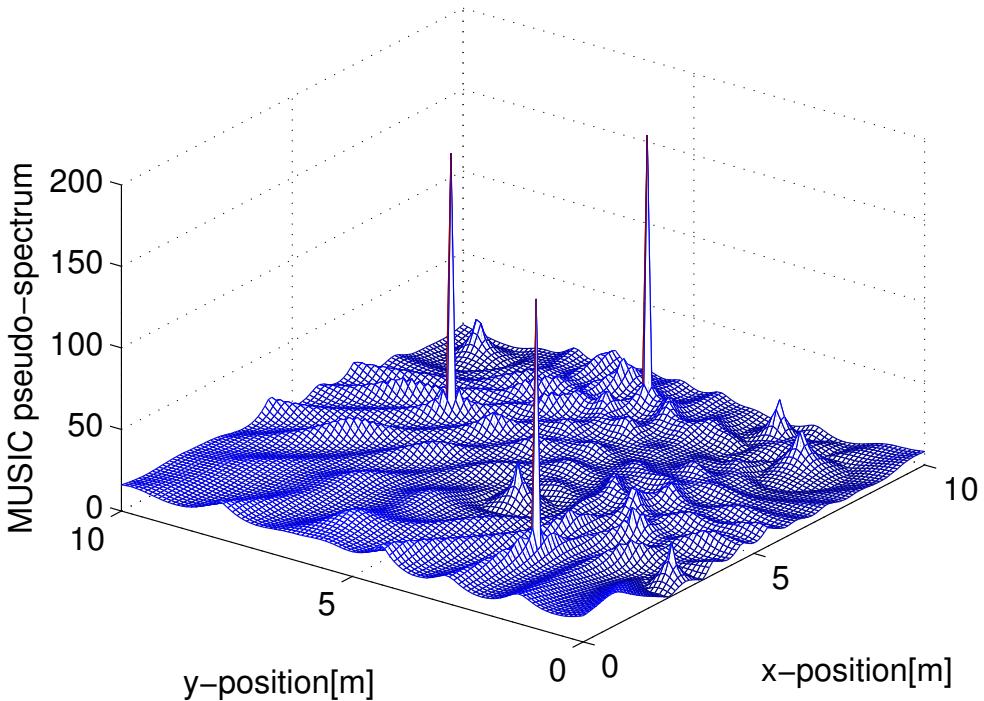


Figure 3.1: The 2D MUSIC pseudo-spectrum. The region of surveillance is a 10×10 2D grid with ten sensors placed at random. The three large peaks correspond to three targets.

3.5.2 MUSIC as a pseudo-likelihood in the PF

In this work, the proposal is to *keep the particles in the space of a single target and use the MUSIC pseudo-spectrum value as a pseudo-likelihood of a particle*. Similar to the PHD function, the MUSIC pseudo-spectrum is also a function that operates in the dimensionality of a single target. The MUSIC pseudo-spectrum is derived more directly from the phased array signals, i.e., by the eigendecomposition of the data covariance matrix, unlike the PHD filter that is derived after thresholding the beamformer spectrum. It is known that the data covariance matrix and the eigenvalues are sufficient statistics for estimating the target location from the phased array data [248–

250]. Note that the proposed method does not compute the entire MUSIC pseudo-spectrum which is a computationally expensive procedure [251], but computes only the MUSIC pseudo-spectrum value for each particle. Hence, the use of MUSIC pseudo-spectrum value as pseudo-likelihood of the particles has two-fold advantage, (*i*) a more direct use of the array data when compared to the PHD filter, and (*ii*) reduced computational cost when compared to the joint filter.

Based on the proposal, (3.5) is now written as

$$\tilde{w}_k^i = w_{k-1}^i \frac{S_{\text{MU}}(\mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, z_{1:k})} \quad (3.13)$$

Then the MTT PF approximation to $p(\mathbf{x}_k | z_{1:k})$ is largest for those particles that are close to the targets and smallest for those particles that are far from the targets. The MUSIC pseudo-likelihood value $S_{\text{MU}}(\mathbf{x}_k^i)$ of a particle \mathbf{x}_k^i is not truly a likelihood (and hence $p(\mathbf{x}_k | z_{1:k})$ is not truly a posterior state distribution), but possesses the characteristics necessary for tracking the mode of the posterior and provides some indication of its spread which is often an adequate estimate of the target state [252]. Other pseudo-likelihood functions have been successfully used in PF based target tracking [253–257].

Handling no information about the number of targets: To be able to compute $S_{\text{MU}}(\mathbf{x}_k^i)$ from (3.12), the noise eigenvectors need to be evaluated from (3.11). η_k in (3.11) should ideally be equal to the actual number of targets N_k which is frequently unknown. Therefore, the proposal is to evaluate η_k using the minimum-mean-square-error (MMSE) based minimum description length (MDL) algorithm [258] that provides an estimate of the actual number of targets in the surveillance region. Using the MMSE based MDL method, η_k is calculated according to

$$\eta_k = \arg \min_{m=1,\dots,M} N_{\text{sp}} \log(\lambda_m) + \frac{1}{2}(m + m^2) \log N_{\text{sp}} \quad (3.14)$$

where N_{sp} is the length of the signal vector s_k at time k and λ_m is the m th eigenvalue obtained after eigendecomposing the $M \times M$ spatial correlation

matrix \tilde{R}_k . A few other methods that aid in computing η_k are the MDL algorithm [259] and the Akaike information criterion (AIC) algorithm [260]. Nevertheless, it is known that the MUSIC algorithm does not catastrophically fail if N_k is over estimated [261].

The major benefit of using MUSIC as the pseudo-likelihood of a particle is that all the particles can now operate in the dimensionality of a single target space. This is illustrated in Fig. 3.2. This feature aids in avoiding the computational complexity problem suffered by the joint filter. Moreover, unlike the PHD filter, the proposed filter can operate more directly on the

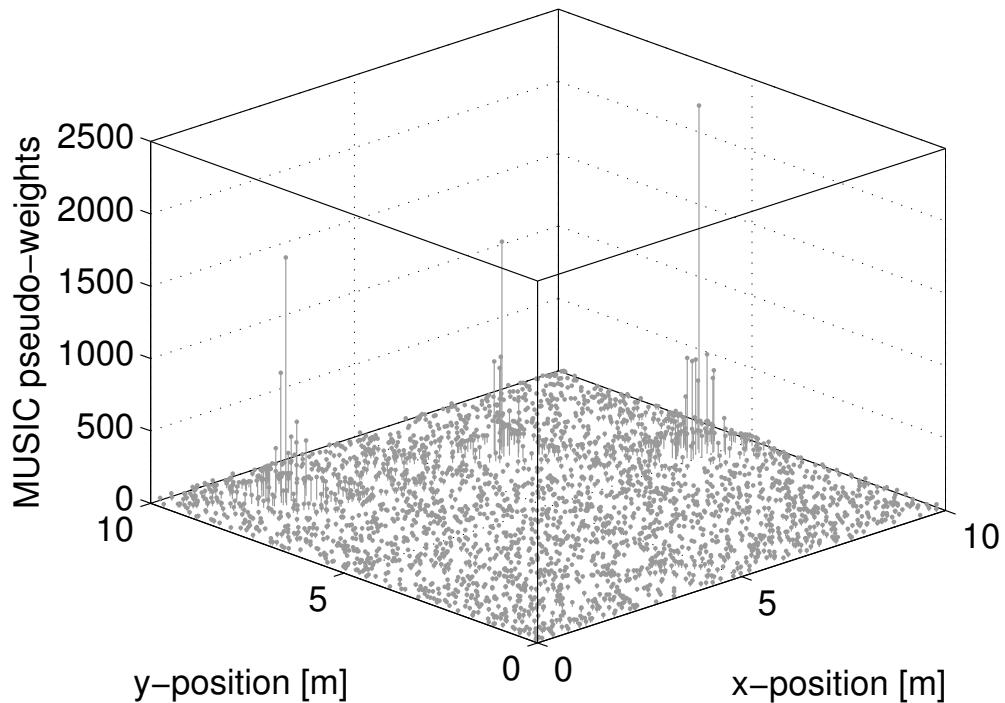


Figure 3.2: Illustration of MUSIC pseudo-weighted particles in the space of a single target. 3000 particles are initialised uniformly in the 2D space. The MUSIC pseudo-weights of the particles are shown in the figure. There are three targets in the space and it can be seen that particles near the targets obtain very large weights.

phased sensor array data. It is expected that resampling the pseudo-weighted particles shown in Fig. 3.2 will result in three clusters of particles with one cluster at each of the three target locations. However, as time progresses, all the particles ultimately concentrate around a single target — the particle hijack problem. This results in the loss of the track of weak (or covert) targets. The particle hijack problem is addressed in the following section.

3.6 Regulated clustering

In this proposal, the particle hijack problem is overcome by employing the concept of *grouping the particles in a regulated fashion*. The well known k-means clustering [59] is used to group the particles and a modified form of soft resampling [61] is proposed to regulate the particle clusters. The regulation of clusters is arranged to specifically prevent any one cluster from containing more than a fixed proportion of the total number of particles. Clustering has been used previously for other PF problems [169]. The procedure is as follows: the I particles are predicted and their MUSIC weights are evaluated using (3.13). Then the I particles are divided into C clusters. Ideally, C should be equal to the number of actual targets N_k . However, since N_k is commonly unknown, the number of clusters formed is chosen to be $C = N_{\max}$ where N_{\max} is the maximum number of targets the filter can track. The process of clustering using the k-means algorithm is now concisely presented.

3.6.1 k-means clustering

The goal of k-means clustering [58, 59] is to categorise the I particles into C clusters in which each particle belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells [262]. The principal idea is to define C centroids at random, one for each cluster, and associate (using a label) each particle to the nearest centroid. Once all the particles have been associated, the C centroids are recalculated

as the center of the cluster of the particles belonging to each of the centroids. The process is repeated until the centroids no longer change. The objective of the algorithm, therefore, is to minimise a chosen distance measure of the I particles from their respective cluster centers. In this proposal, the Euclidean distance metric is used as the distance measure.

Given a set of C initial cluster centroids μ^1, \dots, μ^C , the algorithm operates in two steps;

Assignment: Assign each particle to the nearest centroid, i.e., the labelled particle $x_k^{\bar{c},i}$ will have the label $\bar{c} = c$ if

$$\|x_k^i - \mu^c\|^2 \leq \|x_k^i - \mu^j\|^2 \forall 1 \leq j \leq C \quad (3.15)$$

where $\|x_k^i - \mu^j\|^2$ is the Euclidean distance between the i th particle and the j th centroid.

Centroid update: Assign the mean of each of the C clusters to be the new centroid. Therefore, the new centroids are

$$\mu^c = \frac{\sum x_k^{\mu^c}}{\mathcal{L}_c} \quad (3.16)$$

where $x_k^{\mu^c}$ is the set of particles and \mathcal{L}_c is the number of particles belonging to the c th cluster. The algorithm is repeated until the assignment in (3.16) no longer changes. Various modifications of k-means have been proposed to allow using other distance measures. Other clustering algorithms can be found in [263–265].

In the proposed MUSIC based MTT PF, at this stage, the predicted particles are clustered. A cluster may contain any number of particles and every particle will now have a label \bar{c} that specifies the cluster to which the particle belongs. Therefore, the weighted particle set is now represented as $\{x_k^{\bar{c},i}, w_k^{\bar{c},i}\}_{i=1}^I$ and $\bar{c} \in \chi$ where the index set $\chi = \{1, \dots, C\}$. Then the particles are resampled using a modified form of *soft resampling* [61]. The cluster information of the particles (the labels) is also fed to the resampler

so that the cluster label of each resampled particle is also known. The soft resampling (presented in detail in chapter 5) is briefly described here.

3.6.2 Soft resampling

In the soft resampling stage [61], i.e., after the prediction, weight update and clustering steps, all the I weights are normalised and sorted in descending order. Then for each index $i = 1, \dots, I$, the number of duplications of the i th particle \mathbf{x}_k^i is given by

$$\Gamma_{\mathbf{x}_k^i} = \max\{1, \lfloor Iw_k^i \rfloor\} \quad (3.17)$$

Based on this rule, the new resampled particles with indices $j = \ell, \dots, \ell + \Gamma_{\mathbf{x}_k^i} - 1$ are

$$\check{\mathbf{x}}_k^{\bar{c},j} = \mathbf{x}_k^i \quad (3.18)$$

and their weights are

$$\check{w}_k^{\bar{c},j} = \frac{w_k^i}{\Gamma_{\mathbf{x}_k^i}} \quad (3.19)$$

where

$$\ell = \begin{cases} 1 & \text{if } i = 1 \\ \ell + \Gamma_{\mathbf{x}_k^{i-1}} & \text{if } i > 1 \end{cases} \quad (3.20)$$

It can be understood that a particle having weight less than $2/I$ will appear only once with its weight unchanged, and a particle having weight greater than $2/I$ will be duplicated $\lfloor Iw_k^i \rfloor$ times, with its weight equally divided amongst its duplicated copies.

The number of reweighted particles after this process is

$$\check{I} = \sum_{i=1}^I \Gamma_{\mathbf{x}_k^i} \quad (3.21)$$

and usually $\check{I} > I$. The algorithm, as originally proposed, then discards the particles after the I th particle and preserves their weights. However, in the MUSIC based MTT PF being proposed here, a modified approach is used.

Instead of discarding the particles and preserving their weights, the entire set $\{\tilde{x}_k^{\bar{c},i}, \tilde{w}_k^{\bar{c},i}\}_{i=1}^{\check{I}}$ is retained. These clustered-resampled particles are then regulated in order to limit the number of total particles to I . The process of this regulation is now presented.

3.6.3 Regulation of the clusters

The soft resampling ensures that the particles with large weights are examined for selection first. The number of particles in each cluster is regulated to at most $\beta : \beta \ll I$ particles. This is done as follows:

If the number of particles \mathcal{L}_c that belong to the c th cluster is greater than β , the first β particles and their weights are retained and the rest are removed. Large \mathcal{L}_c indicates that the cluster is near a target because most of its particles are replicated in the resampled particle set. Choosing the first β soft resampled particles ensures that these large weight particles are always retained. On the other hand, if $\mathcal{L}_c < \beta$, all the \mathcal{L}_c particles are retained and $\beta - \mathcal{L}_c$ new particles are appended to the cluster. The $\beta - \mathcal{L}_c$ new particles are initialised uniformly in the observation space and will account for targets appearing in the subsequent time steps. The initialised weights are set to $1/I$. Low \mathcal{L}_c is indicative that the cluster is not near a target because its particles have not gained many replications. However, it is common that some targets generate weak signals at certain times and the clusters tracking these targets might have low \mathcal{L}_c . It is therefore critical not to lose the track of these low intensity targets. The idea of withholding the discarding step in the soft resampling and using the entire set $\{\tilde{x}_k^{\bar{c},i}, \tilde{w}_k^{\bar{c},i}\}_{i=1}^{\check{I}}$ in the regulation process ensures that the particles that are near a low intensity target are not eliminated and hence the information contained in these particles is not lost. The track of this target can hence be preserved.

Overall, this regulation ensures that the particles with large weights that contain information about the targets are always retained, the particles with moderate weights but not tracking any target (these weights could be gradu-

ally increasing with time) are also retained and the particles with low weights are reinitialised to search for new targets (as long as $N_k < C$). Indeed, the retention number of large/moderate weight particles in each cluster is restricted to β . If $N_k = C$, all the clusters are busy tracking the targets and new targets cannot be detected. The total number of clusters C could be chosen such that $C\beta \leq I$. If $C\beta < I$, then $I - C\beta$ new particles are initialised uniformly in the surveillance region and their corresponding weights are set to $1/I$. It is known that $C = N_{\max}$ and if $C\beta \leq I$, then $I - C\beta$ particles never track a target. However, these particles are not a fixed set, i.e., at every time sample k , these particles are only those with low weights — they are not in the process of tracking a target nor in the process of declaring the presence of a covert target. Initialising these particles uniformly in the observation space aids in detecting targets appearing in succeeding time steps.

State estimation: The state estimate can be obtained by taking the mean of the particles in a cluster. The tracks are initiated (i.e., a target declared present) when the cluster looseness, empirically measured as

$$\mathsf{T}_c = \frac{\sum_c ||\check{\mathbf{x}}_k^{\mu^c} - \mu^c||^2}{\mathcal{L}_c} \quad (3.22)$$

falls below a suitable threshold. μ^c is the center of the c th cluster, $\check{\mathbf{x}}_k^{\mu^c}$ are the set of particles that belong to μ^c and \mathcal{L}_c is the number of particles in the cluster $\check{\mathbf{x}}^{\mu^c}$. Track labelling and continuity is maintained even for crossing targets by incorporating the velocity of the targets into the clustering procedure, and weighting the velocity component to be two times more than the actual value.

The complete proposal is outlined in algorithm 1. It is important to note that both the use of a pseudo-likelihood, and the continual reinitialisation of some particles to search of new targets mean that the resulting particle cloud is not an accurate representation of the JMPD nor the PHD. The proposed solution is however effective for detecting and tracking multiple targets as the next section demonstrates.

Algorithm 1 $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^I = \text{PF}[\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^I, z_k]$

1. **Prediction:** $\mathbf{x}_k^i \sim q(\cdot | \mathbf{x}_{k-1}^i, z_k), i = 1, \dots, I$
 2. **Weight update:** $w_k^i = w_{k-1}^i \frac{S_{\text{MU}}(\mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\cdot | \mathbf{x}_{k-1}^i, z_{1:k})}, i = 1, \dots, I$
 3. **Normalise the weights:** $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^I \tilde{w}_k^j}$
 4. **Clustering:** $\{\mathbf{x}_k^{\bar{c},i}, w_k^{\bar{c},i}\}_{i=1}^I \xleftarrow{\text{clustering}} \{\mathbf{x}_k^i, w_k^i\}_{i=1}^I ; \bar{c} \in \chi = \{1, \dots, C\}$
 5. **Soft resample:** $\{\check{\mathbf{x}}_k^{\bar{c},i}, \check{w}_k^{\bar{c},i}\}_{i=1}^I \xleftarrow{\text{resample}} \{\mathbf{x}_k^{\bar{c},i}, w_k^{\bar{c},i}\}_{i=1}^I$
 6. **Cluster regulation:**

```

for  $c = 1, \dots, C$  do
    Obtain the subset of the weighted particles that belong to the cluster  $c$ ,
    i.e.,  $\{\hat{\mathbf{x}}_k^l, \hat{w}_k^l\}_{l=1}^{\mathcal{L}_c} \in \{\check{\mathbf{x}}_k^{\bar{c},i}, \check{w}_k^{\bar{c},i}\}_{i=1}^I : \bar{c} = c$ 
    if  $\mathcal{L}_c \geq \beta$  then
         $\{\xi_k^l\}_{l=(c-1)\beta+1}^{c\beta} = \{\hat{\mathbf{x}}_k^l\}_{l=1}^{\beta}$ 
         $\{\Psi_k^l\}_{l=(c-1)\beta+1}^{c\beta} = \{\hat{w}_k^l\}_{l=1}^{\beta}$ 
    else
         $\{\xi_k^l\}_{l=(c-1)\beta+1}^{c\beta} = \{\hat{\mathbf{x}}_k^l\}_{l=1}^{\mathcal{L}_c} \cup \{\hat{\mathbf{x}}_k^j\}_{j=1}^{\beta-\mathcal{L}_c}$  where  $\hat{\mathbf{x}}_k^j \sim \mathcal{U}(\cdot)$ 
         $\{\Psi_k^l\}_{l=(c-1)\beta+1}^{c\beta} = \{\hat{w}_k^l\}_{l=1}^{\mathcal{L}_c} \cup \{\hat{w}_k^j\}_{j=1}^{\beta-\mathcal{L}_c}$  where  $\hat{w}_k^j = 1/I$ 
    end if
end for
if  $C\beta < I$  then
    Initialise new particles and their weights
     $\xi_k^j \sim \mathcal{U}(\cdot), j = C\beta + 1, \dots, I$ 
 $\Psi_k^j = 1/I, j = C\beta + 1, \dots, I$ 
end if

```
 7. **Normalise the weights:** $\Phi_k^i = \frac{1}{\sum_{j=1}^I \Psi_k^j}, i = 1, \dots, I$
 8. **The output:** $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^I = \{\xi_k^i, \Phi_k^i\}_{i=1}^I$
-

3.7 Simulation results

All the simulations presented here consider acoustic sensing in a 2D environment which includes a 1 second reverberation time for 60 dB decay. The state of the target is defined as $\mathbf{x}_k = (x, y, v_x, v_y)^T$ where x, y are the positions and v_x, v_y are the velocities along the x, y -axis. \mathbf{x}_k evolves with time in accordance to a Markov model in which the position is updated using a constant velocity (CV) model and the velocity is updated using a random walk (RW). This process is described as

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + q_{k-1} \quad (3.23)$$

where the duration between two observations is $T = 1\text{s}$ and the process noise is $q_{k-1} \sim \mathcal{N}(0, Q)$. Q is the uncorrelated covariance matrix with a variance of 0.5 along each dimension. The sensor signals are obtained using the model described in (3.1). The sensor noise vector is $v_k \sim \mathcal{CN}(0, R)$ where R is a diagonal with a variance of $\sigma^2 = 0.1$ corresponding to each sensor.

Demonstration

Firstly, Fig. 3.3 demonstrates the proposed filter operation for tracking single/multiple and appearing/disappearing targets. It can be seen that the filter quickly locks onto the targets as they appear, thereby initialising the tracks. If a target disappears, the particles tracking it disperse, thereby removing the track and these particles then start to search for new targets.

In Fig. 3.4 the particle evolution for the scenario described in Fig. 3.3 is shown. Here, only the particle clusters whose $\exp(-T_c)$ exceed 0.75 are considered for the filter estimate and shown in Fig. 3.4. The use of the Markov transition prior as the importance distribution is known to be sub-optimal.

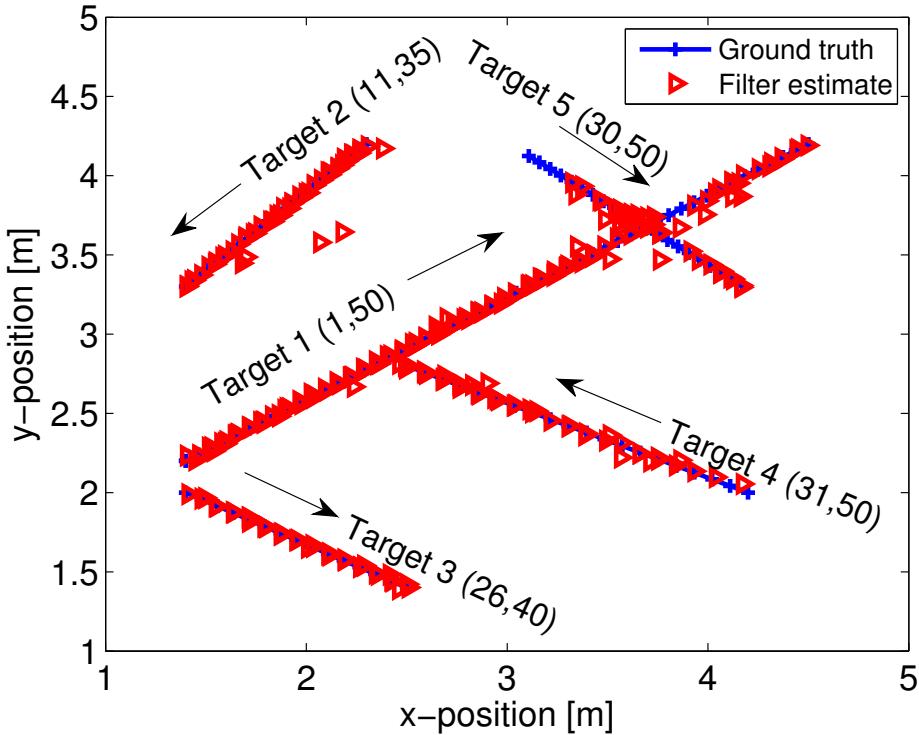


Figure 3.3: This figure shows the ground truth and the filter estimates. Targets move in a 4×4 2D space. A linear array of 5 sensors at an offset of 10m from grid of target movement is used. The number of particles $I = 5000$. At $k = 0$, the particles are initialised uniformly in the surveillance region. The number of clusters is $C = 10$, the maximum number of particles in a cluster is $\beta = 480$, the cluster looseness threshold to initiate and remove tracks is $\exp(-T_c) = 0.75$, the frequency of operation is $f = 4\text{KHz}$ and the propagation velocity is $\nu = 1500\text{m/s}$. The SNR is 27dB. The filter runs for 50 time steps. The text in the figure denotes – Target # (appears at # time step, disappears at # time step). The direction in which each target moves is denoted using arrows.

An importance distribution leveraging the current observation has been proposed for the joint filter [71, 127, 220] and the PHD filter [180]. It may be that the proposed MTT PF could be enhanced using an improved importance distribution. Nevertheless, it can be seen in Fig. 3.3 and Fig. 3.4 that track initialisation and deletion are conducted very effectively.

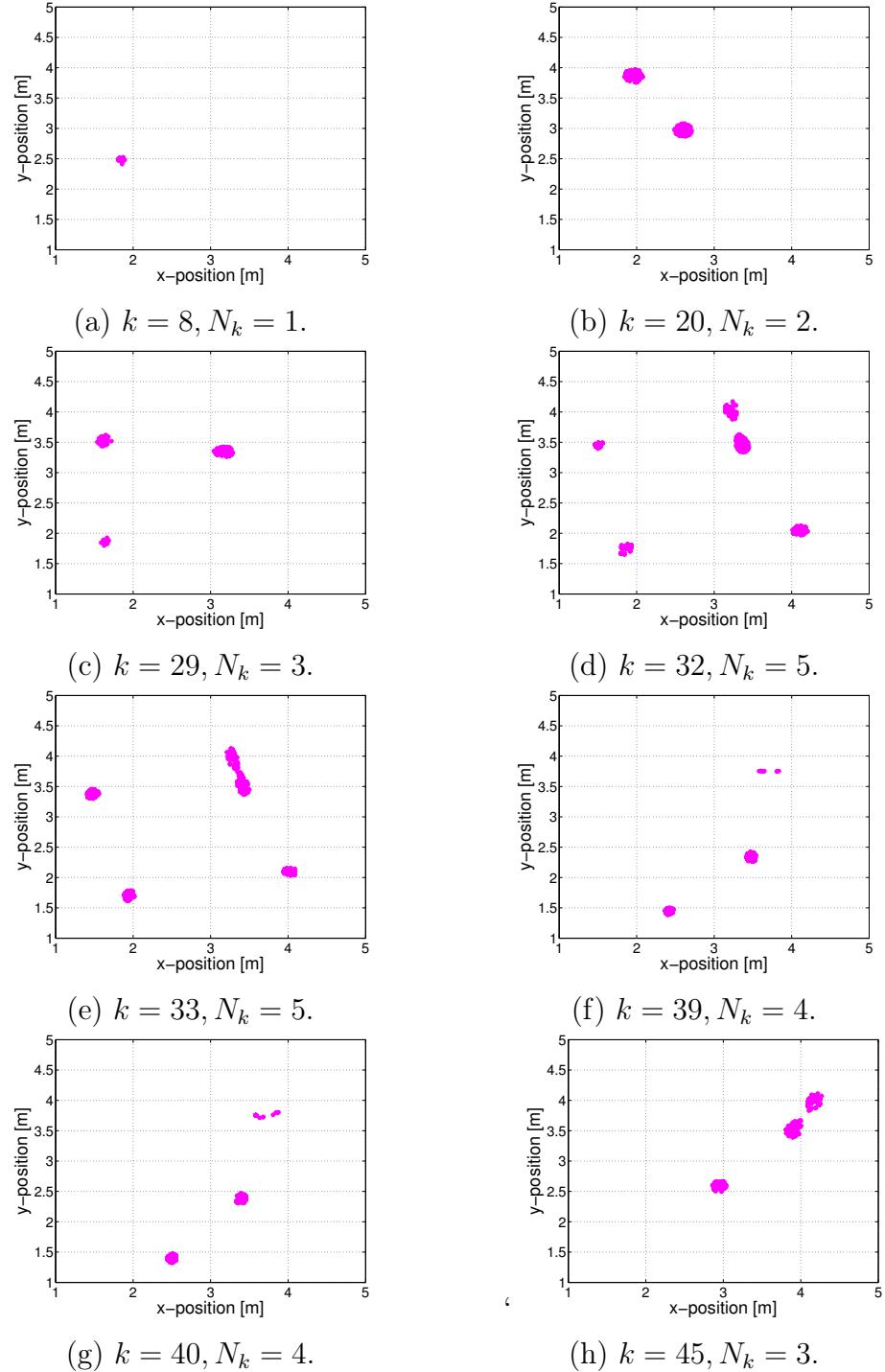


Figure 3.4: Evolution of the particles in the proposed MTT PF for the scenario described in Fig. 3.3. The magenta dots represent the particles. The specifications in the caption of Fig. 3.3 apply to this figure also.

3.7. SIMULATION RESULTS

Fig. 3.4 shows only the particles belonging to the clusters in which $\exp(-T_c) > 0.75$. However, as long as $N_k < C$, the filter has particles associated with the other clusters that are not near the targets. These clusters are free to search for new targets and hence the particles belonging to these clusters are distributed uniformly in the entire observation space. This is illustrated in Fig. 3.5. The figure shows a stem plot of the MUSIC pseudo-weights of all the particles after the soft resampling stage in algorithm 1. The figure corresponds to the 29th time step at which $N_k = 3$. For ease of visualisation, $I = 1000$, $C = 4$, $\beta = 200$. It can be seen that three clusters keep track of the three targets. Moreover, these clusters have gained more replications since they are near the targets. It can also be observed that the particles belonging to the fourth cluster and the remaining $I - C\beta = 200$ particles are free to

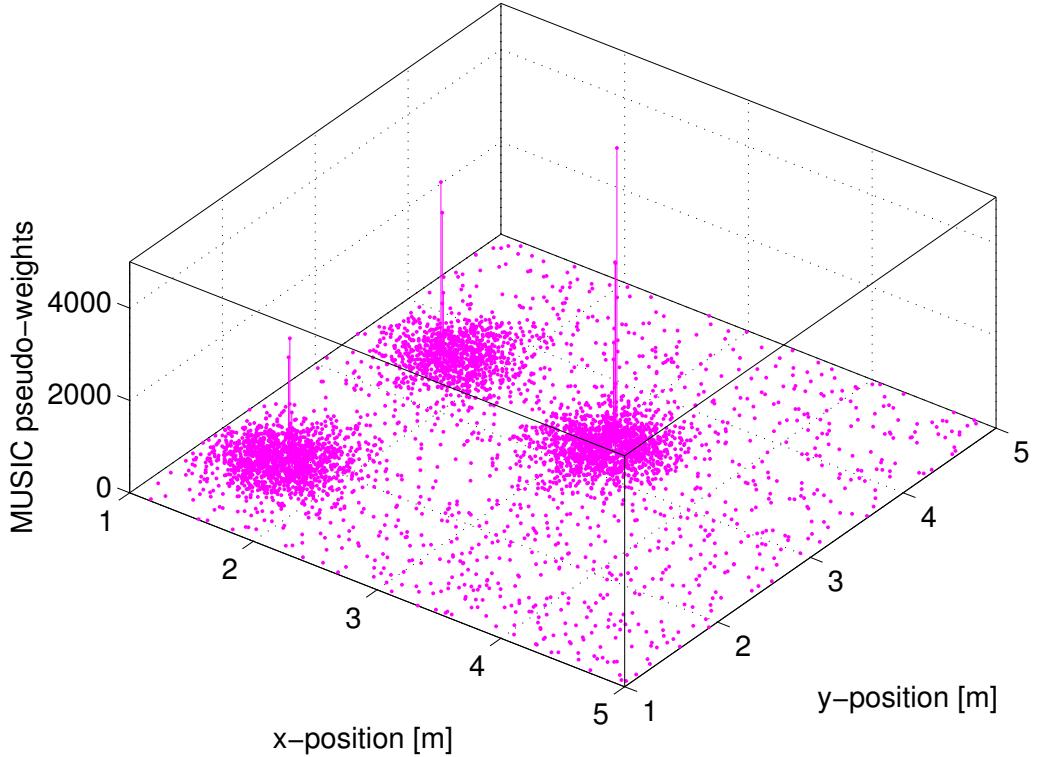


Figure 3.5: The MUSIC pseudo-weights of the particles for the MTT scenario shown in Fig. 3.3 at $k = 29$. The specifications in the caption of Fig. 3.3 apply to this figure also.

search for the new target. The redundant replications in the clusters near the three targets will later be discarded in the regulation process. Particles not in these three clusters have not gained many replications and hence keep searching for new targets.

For the MTT scenario used in Fig. 3.3, Fig. 3.6 shows the MMSE-based-MDL estimate η_k of the number of targets used to evaluate (3.11) for a single run of the 50 time steps. It can be observed that the technique provides an accurate estimate of the actual number of targets N_k .

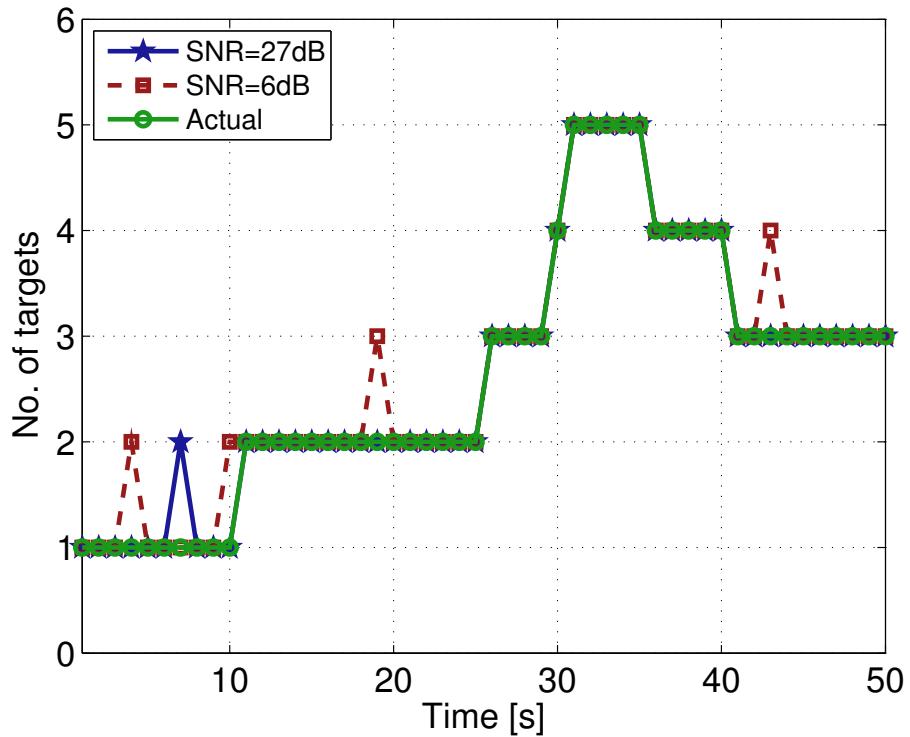


Figure 3.6: The MMSE-based-MDL estimate of the number of targets, versus time.

Demonstration of particle hijack and target resolution

Secondly, the ability of the proposed MTT PF to avoid particle hijack and to resolve close targets is demonstrated. For these tests, the scenario described

3.7. SIMULATION RESULTS

in Fig. 3.7 is used, in which two targets cross each other twice. The targets are present all the time. The motion of the targets is realistic for real-world scenarios where the targets often move in straight lines with constant velocity and occasionally make an abrupt manoeuvre [222].

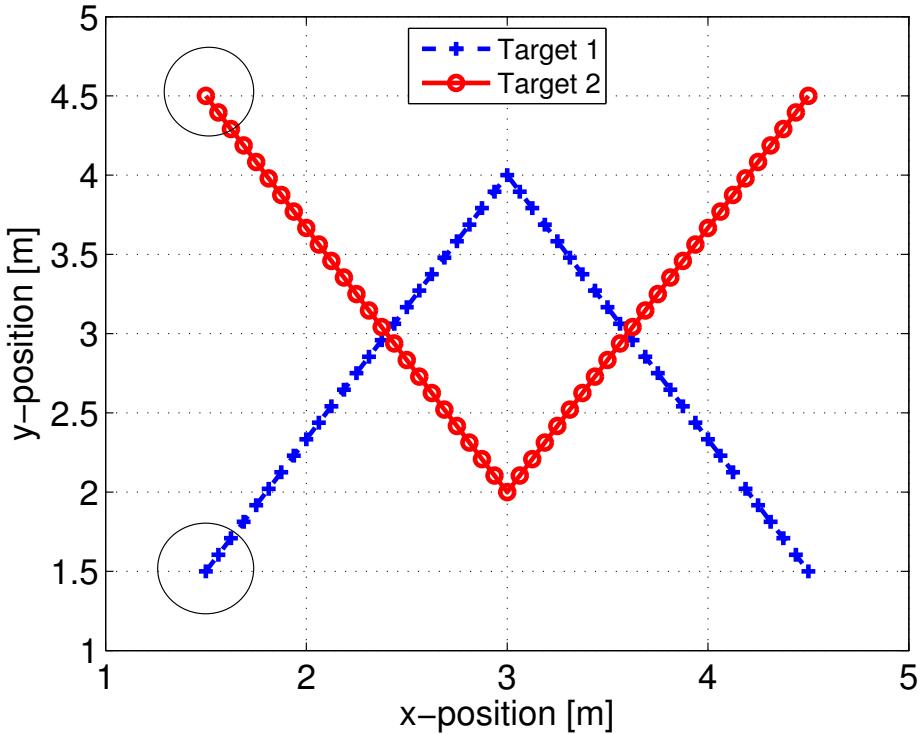


Figure 3.7: The ground truth of the two targets used to obtain the results in Fig. 3.8 and Fig. 3.10. The starting location of the targets is circled. The target positions coincide at $k = 15$ and at $k = 34$.

Fig. 3.8 demonstrates the ability of the proposed MTT PF to overcome particle hijack. While the joint PF and the particle-PHD filter do not suffer from particle hijack, the problem is evident in the separate PF technique and is demonstrated in Fig. 3.8(a). It can be seen that the mode of one filter starts to get hijacked when the targets cross for the first time and eventually the track is lost after some time. The proposed MTT PF overcomes this

hijack problem by virtue of limiting the cluster size to β and reinitialising the redundant lower weight particles.

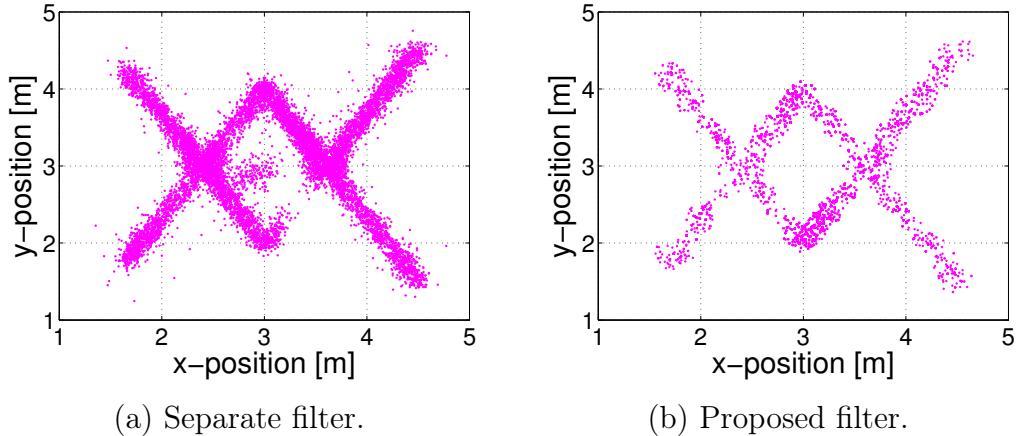


Figure 3.8: Evolution of the particles in various PF’s with time for the scenario described in Fig. 3.7. A linear array of 5 sensors with an offset of 10m from the grid of target movement is used. $f = 4\text{KHz}$, $\nu = 1500\text{m/s}$ and SNR= 27dB. The particles are initialised uniformly in the surveillance region and a burn-in time of 4s is applied before starting the plot. For the separate filter technique, two filters, each using 1000 particles, are initialised. For the proposed filter, $I = 1000$, $C = 2$, $\beta = 500$ and $\exp(-T_c) = 0.75$.

Now the ability of the proposed MTT PF to resolve close targets better than that of the PHD filter is demonstrated. In view of the fact that the latter cannot operate directly on the array data, the sensor signals are first converted into the minimum variance distortionless response (MVDR) beamformer image [244] which is then thresholded to generate the observation feed. In the proposed method, the observations are the sensor signals. Fig. 3.9 illustrates the super-resolution property of MUSIC when compared with the beamformer image. The PF implementation for the ground truth in Fig. 3.7 is illustrated in Fig. 3.10 and it can be seen that the proposed method resolves the targets for a longer duration. At the first crossing at $k = 15$, the PHD filter resolved the targets only until $k = 12$ while the proposed filter until $k = 14$. Before the second crossing at $k = 34$, the PHD filter resolved the targets only until $k = 31$ while the proposed filter until

3.7. SIMULATION RESULTS

$k = 33$. The PHD filter resolution is dependent on how the beamformer image is thresholded — a low threshold deteriorates resolution but a high threshold could degrade the track accuracy at low SNR.

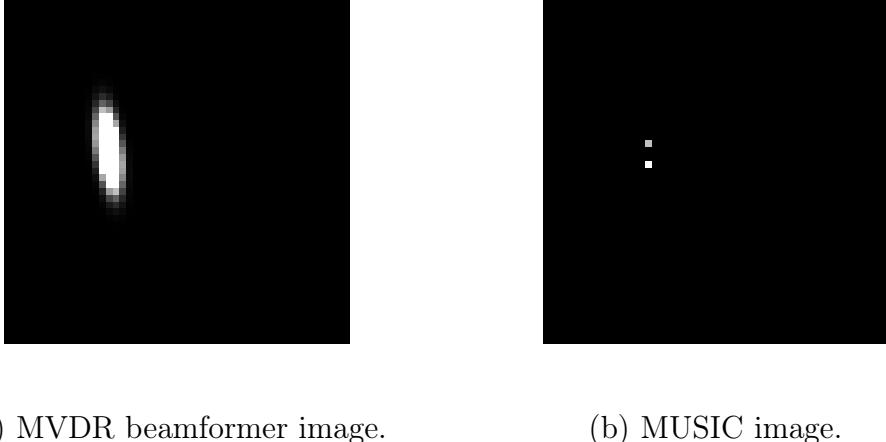


Figure 3.9: Beamformer and MUSIC outputs with two closely spaced targets.

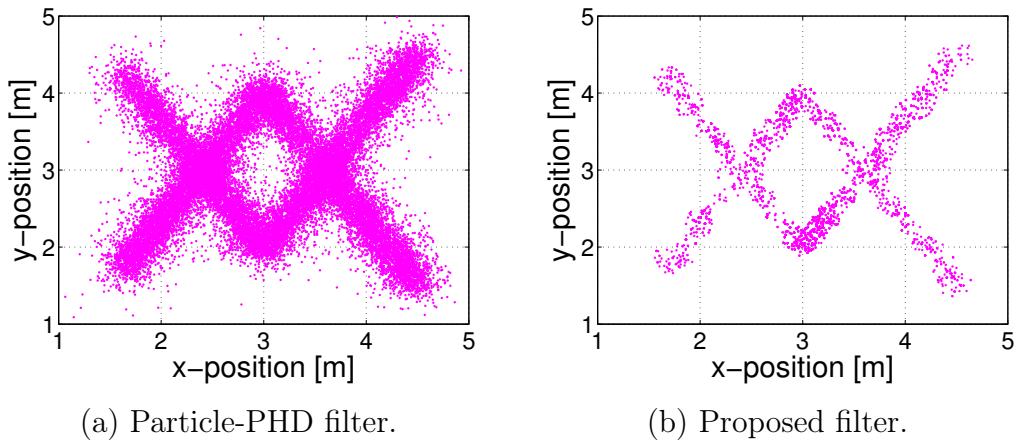


Figure 3.10: Evolution of the particles with time. For the proposed filter, the specifications in the caption of Fig. 3.8 apply to this figure also. For the particle-PHD filter; (i) detection probability $p_D = 0.95$, (ii) target birth is not considered, (iii) thresholding is done at 50% of the maximum peak in the beamformer image, (iv) 1000 particles are used, and (v) targets are separated using k-means clustering [169].

Evaluation of track accuracy

Thirdly, the accuracy of the proposed technique is evaluated by plotting the target position root mean square (rms) error for different SNR levels. This is shown in Fig. 3.11 and it can be seen that the proposed technique is more accurate than the particle-PHD filter. The performance advantage of the proposed filter as against the PHD filter can be attributed to the fact that the former operates directly on the sensor signals while the latter undergoes information loss due to the use of an image processing stage. As the SNR reduces, the false alarms in the finite set measurement feed for the PHD filter increase causing degradation of the accuracy of the track. On the other hand, the separate filter and the joint filter techniques that operate with the true likelihood function in (3.3) exhibit better accuracy than the proposed MTT PF and the particle-PHD filter.

Evaluation of computational complexity

Fourthly, the computational complexity of the proposed technique is evaluated by plotting the computational time taken to evaluate one time step for varying numbers of particles. This is shown in Fig. 3.12. It can be observed that the proposed technique requires fewer computational resources than the others because it uses only I particles (in the domain of one target) that are clustered amongst the two targets. The separate PF technique uses $2I$ particles, i.e., I particles for each target, and the multi-dimensional joint PF uses I particles of double the dimension. In the particle-PHD filter, the majority of the time is consumed evaluating the beamformer image for the entire grid to obtain the finite set measurement feed for the filter. As mentioned in the caption of Fig. 3.11, the number of pixels in the beamformer image is 1681 and hence the filter is slower if $I \lesssim 1700$.

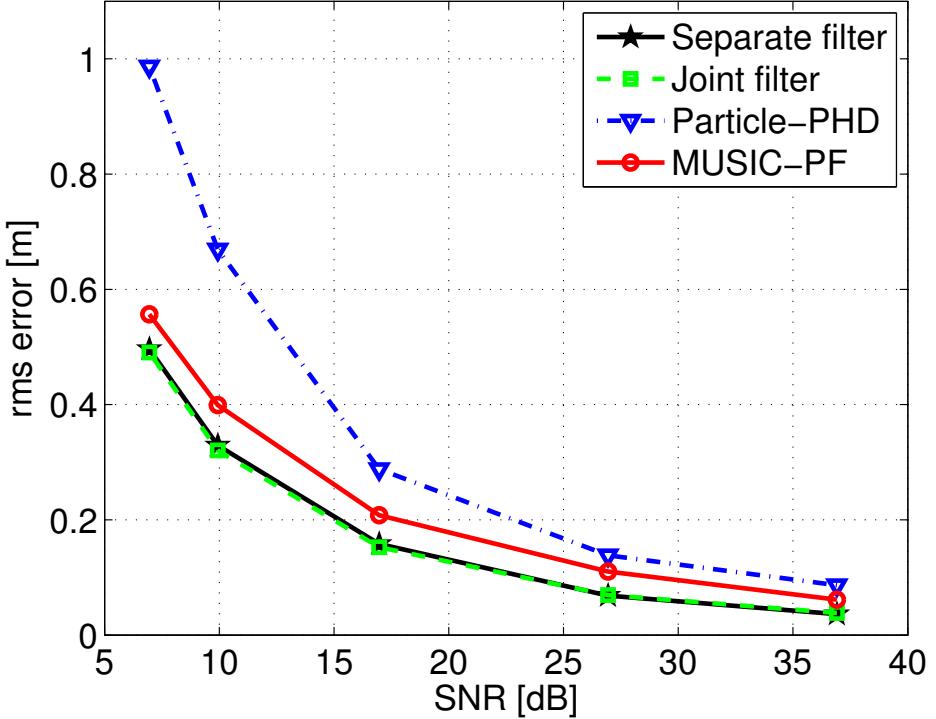


Figure 3.11: rmse error of various PFs versus the SNR for the scenario described in Fig. 3.7. A linear array of 5 sensors with an offset of 10m from the grid of target movement is used. $f = 4\text{KHz}$ and $\nu = 1500\text{m/s}$. It is assumed that the knowledge about the number of targets is available. Hence, for the separate filter technique, two filters, each using 1000 particles are initialised. The joint filter uses 1000 particles that operate in the state space of two targets. For the particle-PHD filter, the beamformer image is evaluated over the observation space with a resolution of 0.1m. Therefore, the total number of pixels in the beamformer image are 1681. During the particle-PHD filter implementation; a) the detection probability $p_D = 0.95$, b) target birth is not considered, c) thresholding is done at 50% of the maximum peak in the beamformer image, and d) 1000 particles are used. To obtain state estimates, the k-means clustering approach of [169] is used. For the proposed filter, $I = 1000$, $C = 2$, $\beta = 500$ and $\exp(-T_c) = 0.75$. No burn-in time is considered. The results are averaged over 30 scenarios, each comprising 50 time steps.

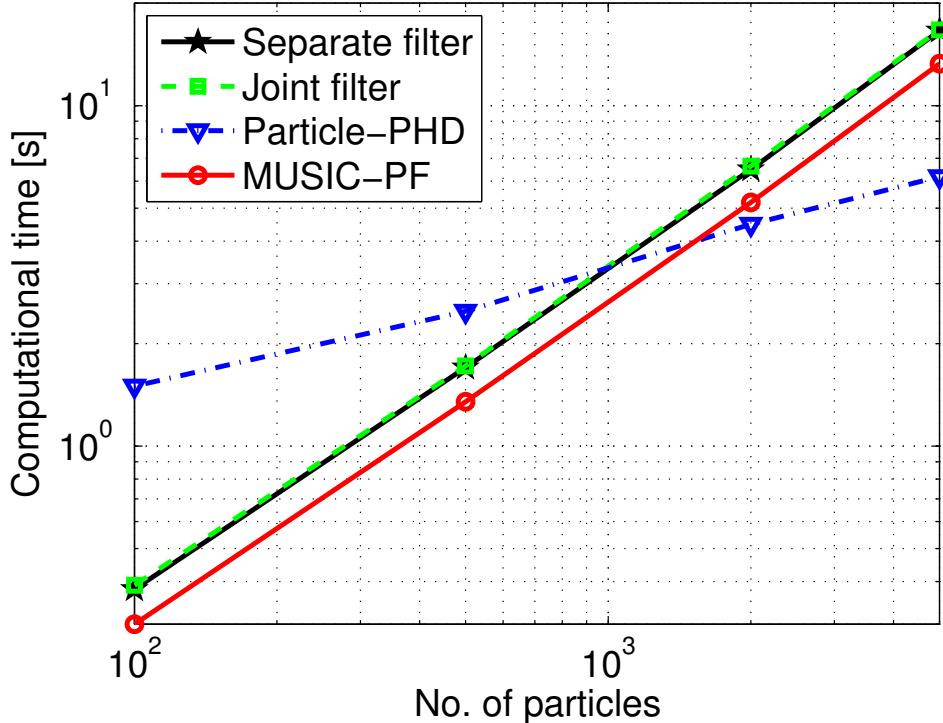


Figure 3.12: Computational time for one time step of various PFs versus the number of particles. The specifications in the caption of Fig. 3.11 apply to this figure, except that now, I is varied and the SNR=27dB. The results are averaged over 30 scenarios, each comprising 50 time steps.

Demonstration of track continuity

Finally, the track continuity in the proposed MTT PF is demonstrated. As proposed in section 3.6.3, the track continuity is maintained by incorporating the velocity into the clustering procedure with a weighting that is twice its actual value. For a three target case, Fig. 3.13 shows the track continuity in the proposed filter for crossing targets using colored particle clusters. It can be seen that the filter successfully maintains the track of the crossing targets. However, it would be difficult to maintain tracks if two targets travel together for some duration and then separate with opposite velocities.

3.7. SIMULATION RESULTS

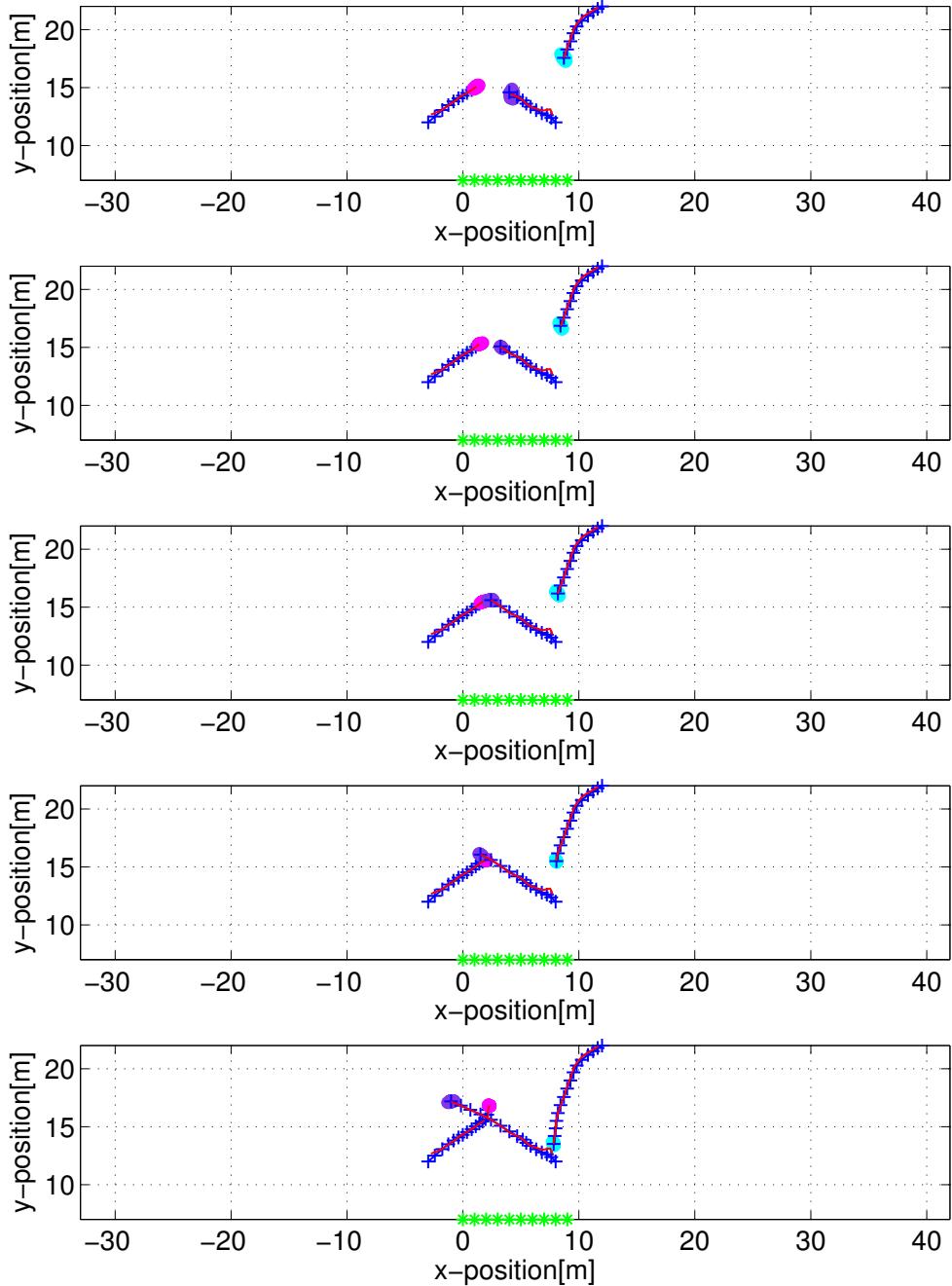


Figure 3.13: Particle evolution in the proposed MTT PF with time. Here $I = 6000$, $C = 3$ and $\beta = 2000$. The figures from top to bottom correspond to time steps $k = 9, 10, 11, 12, 15$ respectively. The blue crossed line is the ground truth. The red solid line is the filter estimate. The three clusters are represented in different colors. The green stars depict a linear array of 10 sensors.

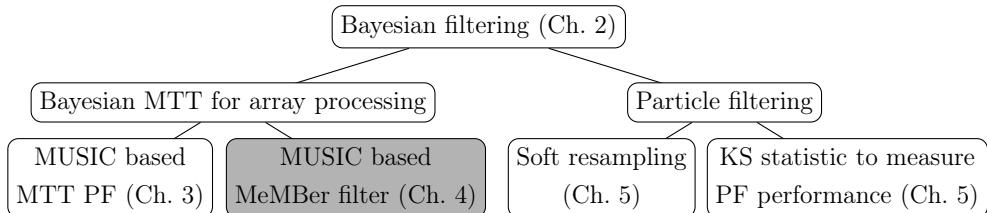
3.8 Conclusion and summary

This chapter addressed the problem of Bayesian MTT using phased sensor arrays under the PF framework. State-of-the-art approaches to this problem include the separate filter approach which suffers from filter hijack, the joint approach which suffers from high computational complexity and the PHD filter [26] which suffers from the inability to operate directly on phased sensor array signals. This chapter proposed MUSIC based MTT PF, the key innovation of which is to use for the likelihood function, a function which has as its domain, the state space of a single target. The MUSIC pseudo-spectrum is such a function, and has not previously been used in this context. The particle hijack problem is overcome using regulated clustering. This work is the first to propose soft resampling for cluster regulation — a novel mechanism that provided means for initialising new particles that account for appearing targets. The advantages of the developed filter are; a) lower computational complexity by virtue of operating in the single target space, b) better accuracy by virtue of operating more directly on sensor array signals, c) effective separation of close targets by virtue of the super resolution property of MUSIC, and d) avoidance of particle hijack by virtue of regulated clustering. The validity and the efficacy of the technique were illustrated through a simulated study. Referring back to section 1.3.1 of chapter 1, it can be stated that the research goal 1 that aims to overcome problems 1,2 has been successfully achieved by the proposed MUSIC based MTT PF. Further evaluation of the performance of this algorithm is included in section 4.6.1 of the next chapter.

3.8. CONCLUSION AND SUMMARY

4

The MUSIC based MeMBER filter



Bayesian filtering [35] provides a rigorous mathematical framework for multiple target tracking (MTT) using phased sensor arrays — an application of primary interest to this dissertation. However, Bayes’ filtering involves high dimensional integration which makes the process computationally intractable [45]. This intractability is overcome using approximate solutions. The Bayes’ approximation which is central to this thesis is the particle filter (PF) [21] described in section 2.5 of chapter 2.

The theoretically accurate approach to Bayesian MTT is the joint filter approach [44, 124]. However, the joint PF is computationally expensive [232] and this drawback restricts its use for tracking many targets. The high

computational complexity suffered by the joint filter has been successfully overcome using the random finite set (RFS) filters [179] — the probability hypothesis density (PHD) filter [26, 32, 167] and the multi-target multi-Bernoulli (MeMBer) filter [31, 32, 196, 198, 208, 209]. The major drawback in RFS filters, as identified in problem 2 of section 1.3.1 in chapter 1, is that they do not possess the mathematical framework to operate directly on phased array data and consequently the array data has to be converted into a suitable format. This conversion leads to high information and resolution loss. The research goal 2 in section 1.3.1 of chapter 1 aims to explore the possibility and develop a method of operating either the PHD or the MeMBer filters directly on phased sensor array signals. This chapter presents the development of such a method and illustrates its efficacy.

The chapter is organised as follows: The drawbacks and related work in state-of-the-art RFS filters are outlined in section 4.1. After stating the motivation in section 4.2, the phased sensor array model and the particle-MeMBer filter for image observations are revisited in section 4.3 and section 4.4 respectively. This is followed by the proposed method in section 4.5, the simulation results in section 4.6 and the concluding remarks in section 4.7.

4.1 Related work

The class of Bayesian sub-optimal filters developed for tracking many targets are the RFS filters [233] — the PHD [26] and the MeMBer [31, 32] filters. The PHD filter described in section 2.6 of chapter 2 operates by approximating the joint multi-target probability distribution (JMPD) ((2.13) in section 2.3.2 of chapter 2) by its first moment. The PF implementation of the PHD filter can be found in [161, 162, 164]. The MeMBer filter [198, 209] described in section 2.7 of chapter 2 approximates the JMPD using its multi-Bernoulli RFS parameters. The PF implementation of the MeMBer filter can be found in [31, 33, 49, 196]. The PHD and MeMBer filters are derived under the RFS theoretic framework [34, 233] that allows them to operate in the dimension-

ality of a single target, i.e., their complexity does not increase exponentially with the number of targets. However, since the filters are sub-optimal, their performance is poor in many situations [266, 267], for example, in high noise scenarios [161].

The major impediment in RFS filtering, as mentioned in section 3.1 of chapter 3, is that the observation feed to the filters can *only* be a finite set of noisy point target measurements [32], thereby *impeding the direct use of phased sensor array data*. This drawback is only partially overcome when the MeMBer filter has been extended to image observations [49, 196]. To date, the RFS filters do not provide a suitable mathematical formalism to evaluate the likelihood function for phased array data, because these filters operate in the dimensionality of a single target, while the phased array data requires a joint state hypothesis to evaluate the likelihood. In the PHD filter, this problem is usually bypassed by first converting the array data into a beamformer image [50, 189, 191], and then pre-processing the image [192] to obtain the measurement feed. The “MeMBer filter for image observations” does not require image pre-processing, but still necessitates the conversion of the sensor array data into a image. A closely related work in [268, 269] deals with super-positional sensors that generate a “single measurement” from multiple targets. However, this single measurement was considered to be a point estimate of the targets rather than a signal. This still requires the array data to be converted to a beamformer image. These conversions lead to substantial information and resolution loss.

4.2 Motivation

This thesis proposes that the formulation in the “MeMBer filter for image observations” provides a possibility to operate the filter more directly on the sensor array data. The aim now is to adapt the MeMBer filter for image observations to phased sensor array observations. The key idea for this adaption is to find a function (generated using the array data), whose domain is

the domain of a single target, and which can act as a proxy to the MeMBer update equation without changing its formulation. The MULTiple SIgnal Classification (MUSIC) pseudo-spectrum [57] is such a function. MUSIC is one of what are called “super-resolution techniques.” This work proposes to use MUSIC as the pseudo-likelihood in the MeMBer filter for image observations.

The MUSIC based MTT PF proposed in chapter 3 compared favourably with the PHD filter and the comparison with the MeMBer filter has been deferred until this chapter. In this chapter, the comparative analysis with the conventional PHD and MeMBer filters, the previously proposed MUSIC based MTT PF, and the MUSIC based MeMBer filter of this chapter is presented.

4.3 Phased sensor array model

Here, the phased sensor array observation model [38] (previously presented in section 3.3 of chapter 3) is briefly described. The observation model is formulated as

$$z_k = A_k s_k + v_k \quad (4.1)$$

where z_k is the vector of sensor signals, s_k is the signal vector and the noise vector $v_k \sim \mathcal{CN}(0, R)$. A_k is a $\mathbf{M} \times N_k$ steering matrix where \mathbf{M} is the number of sensors and N_k is the number of targets at time k . For this sensing model, the likelihood equation is

$$p(z_k | A_k, s_k) = \frac{1}{\pi^{\mathbf{M}} \det(R)} \exp\left(- (z_k - A_k s_k)^H R^{-1} (z_k - A_k s_k)\right) \quad (4.2)$$

where $(.)^H$ denotes Hermitian transpose. The PHD and MeMBer filters do not possess a framework to evaluate (4.2). This non-provision can be bypassed using beamformer image models [50, 189, 191, 192]. The proposal in this chapter is based on the idea that using MUSIC¹ as a proxy for the likeli-

¹It is known from the MUSIC based MTT PF in chapter 3 that MUSIC allows the particles to operate in the space of a single target, a requisite for the MeMBer filter.

hood provides a framework to feed the sensor array data as an observation to the MeMBer filter (for image observations) without changing its formulation. Moreover, the super resolution property of MUSIC aids in separating very close targets. The objective now is to adapt the MeMBer filter for image observations to phased sensor array signals z_k obtained from (4.1).

4.4 The particle-MeMBer filter for image data

The operation of the MeMBer filter for image data [49, 196, 208, 211, 214] was described in section 2.7.3 of chapter 2. Here its particle implementation is briefly described. The filter is derived and explained in more detail in [49, 196]. The proposal of adapting the MeMBer filter to array data then builds on the formulation provided in this section.

In MeMBer filtering, the uncertainty in the knowledge of the number of targets and their states is modelled using a multi-Bernoulli RFS Ξ . A multi-Bernoulli RFS Ξ_k at time k can be considered as the union of a fixed number of independent Bernoulli sets. The n th Bernoulli set in Ξ_k is described by its a) existence probability ϵ_k^n , and b) probability density function (pdf) $p^n(x_k)$. In the particle approach, the pdf is approximated using a set of I weighted particles, i.e., $p^n(x_k)$ is now represented by the weighted sample set $\{x_k^{n,i}, w_k^{n,i}\}_{i=1}^I$.

The particle-MeMBer filter recursion is as follows (the explanation provided in section 2.7.3 of chapter 2 is directly applicable to the particle-MeMBer filter): If the particle-MeMBer approximation to the posterior at time $k - 1$ is available and represented by

$$p(X_{k-1}|z_{k-1}) \approx \left\{ \epsilon_{k-1}^n, \{x_{k-1}^{n,i}, w_{k-1}^{n,i}\}_{i=1}^I \right\}_{n=1}^{N_{k-1}} \quad (4.3)$$

where we define \approx to mean “is represented by,” then the particle-MeMBer approximation to the Bayes’ prediction is represented by

$$\begin{aligned} p(\mathbf{X}_k | z_{1:k-1}) &\approx \{\hat{\epsilon}_k^n, \{\hat{x}_k^{n,i}, \hat{w}_k^{n,i}\}_{i=1}^I\}_{n=1}^{N_k} \\ &\approx \{\epsilon_{P,k}^n, \{x_{P,k}^{n,i}, w_{P,k}^{n,i}\}_{i=1}^I\}_{n=1}^{N_{k-1}} \cup \{\epsilon_{B,k}^n, \{x_{B,k}^{n,i}, w_{B,k}^{n,i}\}_{i=1}^I\}_{n=1}^{N_{B,k}} \end{aligned} \quad (4.4)$$

The multi-Bernoulli RFS $\{\epsilon_{P,k}^n, \{x_{P,k}^{n,i}, w_{P,k}^{n,i}\}_{i=1}^I\}_{n=1}^{N_{k-1}}$ in (4.4) contains the Bernoulli parameters (or target hypotheses) that persist from time $k-1$ to time k . For $n = 1, \dots, N_{k-1}$, the Bernoulli parameters are predicted according to

$$\epsilon_{P,k}^n = \epsilon_{k-1}^n \sum_{i=1}^I w_{k-1}^{n,i} p_S(x_{k-1}^{n,i}) \quad (4.5)$$

$$x_{P,k}^{n,i} \sim q(\cdot | x_{k-1}^{n,i}, z_k) = p(x_{P,k}^{n,i} | x_{k-1}^{n,i}) ; i = 1, \dots, I \quad (4.6)$$

$$w_{P,k}^{n,i} = \frac{w_{k-1}^{n,i} p_S(x_{k-1}^{n,i})}{\sum_{l=1}^I w_{k-1}^{n,l} p_S(x_{k-1}^{n,l})} ; i = 1, \dots, I \quad (4.7)$$

where $p_S(\cdot)$ is the survival probability of the particle.

In (4.5), $\epsilon_{P,k}^n$ is the existence probability that the n th persisting Bernoulli RFS contains an actual target; if the survival probability is a constant value p_S , then $\epsilon_{P,k}^n$ is obtained by directly scaling the existence probability ϵ_{k-1}^n of the Bernoulli RFS at the previous time step with p_S (since the particle weights sum to one). In (4.6), the particles that represent the pdf at time $k-1$ are propagated to time k by drawing a new set of particles from the Markov transition prior. In (4.7), the weights of the predicted particles survive from time $k-1$ to k in accordance to the particle survival probabilities. These weights are then normalised.

The multi-Bernoulli RFS $\{\epsilon_{B,k}^n, \{x_{B,k}^{n,i}, w_{B,k}^{n,i}\}_{i=1}^I\}_{n=1}^{N_{B,k}}$ in (4.4) contains the multi-Bernoulli parameters for the $N_{B,k}$ new-born targets at time k , where $N_{B,k} \sim \text{Pois}(\lambda_B)$. The new particles are initialised near regions of target appearance and the existence probabilities are set accordingly. If the information about regions of target appearance is not available, then $\epsilon_{B,k}^n \sim \mathcal{U}(0, 1]$,

and for $i = 1, \dots, I$, $\mathbf{x}_{B,k}^{n,i} \sim \mathcal{U}(.)$ and $w_{B,k}^{n,i} = 1/I$.

Once the new image observation z_k and the predicted particle-MeMBer parameters $\{\hat{\epsilon}_k^n, \{\hat{\mathbf{x}}_k^{n,i}, \hat{w}_k^{n,i}\}_{i=1}^I\}_{n=1}^{N_k}$ are available, the posterior at time k can be approximated [49] using the updated Bernoulli parameters as

$$p(\mathbf{X}_k | z_{1:k}) \approx \{\epsilon_k^n, \{\mathbf{x}_k^{n,i}, w_k^{n,i}\}_{i=1}^I\}_{n=1}^{N_k} \quad (4.8)$$

where

$$\epsilon_k^n = \frac{\hat{\epsilon}_k^n \sum_{i=1}^I \hat{w}_k^{n,i} \bar{s}_{z_k}(\hat{\mathbf{x}}_k^{n,i})}{1 - \hat{\epsilon}_k^n + \hat{\epsilon}_k^n \sum_{i=1}^I \hat{w}_k^{n,i} \bar{s}_{z_k}(\hat{\mathbf{x}}_k^{n,i})} \quad (4.9)$$

$$w_k^{n,i} = \frac{\hat{w}_k^{n,i} \bar{s}_{z_k}(\hat{\mathbf{x}}_k^{n,i})}{\sum_{l=1}^I \hat{w}_k^{n,l} \bar{s}_{z_k}(\hat{\mathbf{x}}_k^{n,l})} \quad (4.10)$$

Here, (4.9) gives the updated existence probability of the n th Bernoulli RFS (its functionality was discussed in Fig. 2.8 of section 2.7.3 in chapter 2). (4.10) gives the updated particle weights; this is obtained using a PF update step — by multiplying the predicted weights with the likelihood value of the particles. The weights are then normalised. The updated particles $\{\{\mathbf{x}_k^{n,i}\}_{i=1}^I\}_{n=1}^{N_k}$ in (4.8) are obtained by resampling the particles (based on their updated weights) in each predicted Bernoulli set separately.

The state estimate is the mean of all the particles belonging to a Bernoulli set. If the distance between the estimates of two Bernoulli sets is closer than a threshold $\mathcal{T}_{\text{merge}}$, their tracks are merged [32]. The state estimates of Bernoulli sets whose existence probability exceeds a threshold $\mathcal{T}_{\text{prune}}$ are only reported.

The term $\bar{s}_{z_k}(\hat{\mathbf{x}}_k^{n,i})$ in (4.9) and (4.10) is the likelihood ratio of the i th particle in the n th predicted Bernoulli set and its formulation was given in (2.107) of section 2.7.2 in chapter 2. Using the image observation model described² in Ch. 11 of [56] in which the signal and noise components follow a Gaussian distribution, the $\bar{s}_{z_k}(\hat{\mathbf{x}}_k^{n,i})$ in (2.107) can be written up to a

²The complete model is presented further in section 5.6.1 of chapter 5.

normalising constant as (shown in (2.111))

$$\bar{s}_{z_k}(\hat{x}_k^{n,i}) = \prod_{(x,y) \in \mathfrak{R}_o(\hat{x}_k^{n,i})} \exp\left(-\frac{\gamma_{\hat{x}_k^{n,i}}^{(x,y)}(\gamma_{\hat{x}_k^{n,i}}^{(x,y)} - 2z_k^{(x,y)})}{2\sigma^2}\right) \quad (4.11)$$

Here, z_k is the received image. The hypothesis measure $\gamma_{\hat{x}_k^{n,i}}$ (also a image) is the intensity contribution of the particle $\hat{x}_k^{n,i}$ and is constructed from the image observation model considering the absence of noise. The particle $\hat{x}_k^{n,i}$ illuminates a region $\mathfrak{R}_o(\hat{x}_k^{n,i})$ of the image, i.e., the target hypothesis $\hat{x}_k^{n,i}$ will illuminate a set of pixels, the centers (x, y) of which will lie within the region of illuminance $\mathfrak{R}_o(\hat{x}_k^{n,i})$. This was illustrated in Fig. 2.7 in which a target at position (40, 40) illuminates a set $\mathfrak{R}_o(\cdot)$ of pixels within the frame. For image models, the set of pixels with centers $(x, y) \in \mathfrak{R}_o(\hat{x}_k^{n,i})$ required to evaluate (4.11) is usually fixed and known in advance. If this information is not available, then $\mathfrak{R}_o(\cdot)$ can be obtained by thresholding $\gamma_{\hat{x}_k^{n,i}}$ using an illumination threshold I_l which is usually a certain proportion of the peak in the image $\gamma_{\hat{x}_k^{n,i}}$. Then the likelihood $\bar{s}_{z_k}(\cdot)$ is obtained by taking the product of the exponential at each of these pixels. Note that (4.11) is computed for all the IN_k particles in the predicted multi-Bernoulli set.

4.5 The MUSIC based MeMBer filter

The non-Bayesian MUSIC algorithm for target localisation has been described in section 3.5.1 of chapter 3 and will not be repeated here. In this work, the proposal is to *use the MUSIC value as a pseudo-likelihood* for the particle [270]. Hence, the proposed MUSIC based particle-MeMBer filter operation is just as described in section 4.4, except that (4.9) and (4.10) are now computed according to

$$\epsilon_k^n = \frac{\hat{\epsilon}_k^n \sum_{i=1}^I \hat{w}_k^{n,i} S_{\text{MU}}(\hat{x}_k^{n,i})}{1 - \hat{\epsilon}_k^n + \hat{\epsilon}_k^n \sum_{i=1}^I \hat{w}_k^{n,i} S_{\text{MU}}(\hat{x}_k^{n,i})} \quad (4.12)$$

$$w_k^{n,i} = \frac{\hat{w}_k^{n,i} S_{\text{MU}}(\hat{x}_k^{n,i})}{\sum_{l=1}^I \hat{w}_k^{n,l} S_{\text{MU}}(\hat{x}_k^{n,l})} \quad (4.13)$$

where

$$S_{\text{MU}}(\hat{x}_k^{n,i}) = \frac{1}{\left(a_k^{\hat{x}_k^{n,i}}\right)^H U_n U_n^H a_k^{\hat{x}_k^{n,i}}} \quad (4.14)$$

is the MUSIC pseudo-likelihood of the predicted particle $\hat{x}_k^{n,i}$. In (4.14), the steering vector $a_k^{\hat{x}_k^{n,i}}$ of $\hat{x}_k^{n,i}$ is computed from (3.2) in section 3.3 of chapter 3. U_n , computed from (3.11) in section 3.5.1 of chapter 3 contains the noise eigenvectors. The evaluation of (3.11) involves the term η_k which should ideally be equal to the actual number of targets N_k which is frequently unknown. This thesis proposes to overcome the ignorance of N_k using the minimum-mean-square-error (MMSE) based minimum description length (MDL) algorithm [258] that provides good estimate of the actual number of targets $\eta_k \approx N_k$. However, it is known that the MUSIC algorithm does not catastrophically fail if N_k is over estimated [261].

The fundamental contribution of this proposition is that, while the MeMBER filter in its originally proposed form requires the sensor observation z_k to be a image for evaluating the likelihood ratio in (4.11), the proposed technique allows z_k to be expressed directly as the phased sensor array observation model described in (4.1). This is because the particle's likelihood $\bar{s}_{z_k}(\hat{x}_k^{n,i})$, which is now chosen to be $S_{\text{MU}}(\hat{x}_k^{n,i})$ is evaluated according to (4.14) which requires the observation feed z_k to be sensor array signals. Hence the proposed method makes a more direct use of the array data and this will limit the information loss problem from which the MeMBER and PHD filters suffer. However, inasmuch as $S_{\text{MU}}(\cdot)$ is not truly a likelihood, the posterior is not truly a multi-target posterior state distribution, but possesses the attributes necessary for tracking the mode of the posterior which is often an adequate estimate of the target state [252]. A few other pseudo-likelihood based PF approaches can be found in [253, 255–257, 271].

The proposal also aids in significantly reducing the computational cost. In order to evaluate (4.11), the “MeMBER filter for beamformer images” requires

the computation of the minimum variance distortionless response³ (MVDR) [244] beamformer values $\gamma_{(.)}^{(x,y)}$ for the pixels that fall within the region of influence $(x, y) \in \mathfrak{R}_0(.)$ of the target hypothesis. In beamforming methods, the number of these pixels is fluctuating and is large if the target hypothesis is far from the sensors. This is shown using a simple 1D example in Fig. 4.1.

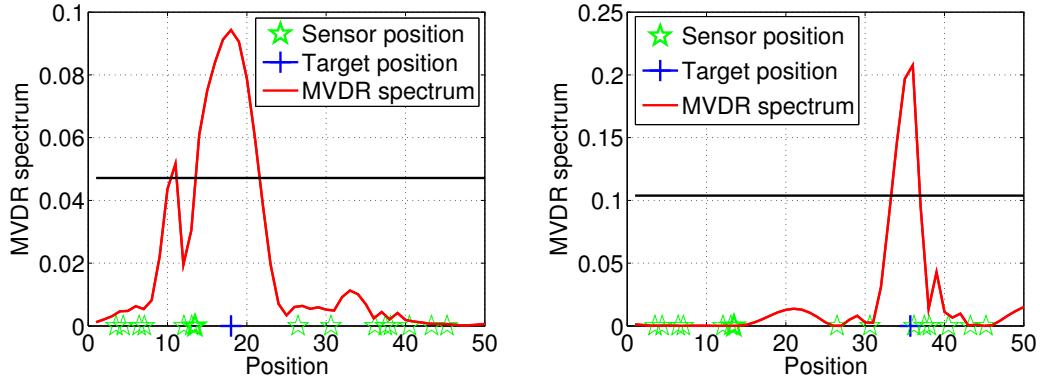


Figure 4.1: The MVDR spectrum for a simple 1D case. The solid black line shows the MVDR image thresholded at 50% of its maximum value. The figure shows that the number of pixels illuminated by a target hypothesis (the blue square) is dependent on its proximity to the sensors.

Consider that each predicted particle $\hat{x}_k^{n,i}$ at time k will influence $\beta_k^{n,i}$ pixels (from Fig. 4.1, it is obvious that always $\beta_k^{n,i} > 1$), then the number of beamformer computations per time is $\sum_{n=1}^{N_k} \sum_{i=1}^I \beta_k^{n,i}$. If the illumination threshold I_l is high, i.e., the beamformer image is thresholded using a higher value, then $\mathfrak{R}_0(.)$ is small and the amount of sensor information z_k processed by each particle is low. In such a case, the filter performance will depend on the number of particles I spanning the observation space — large I allows more data z_k to be filtered but also increases computational complexity. On the other hand, if I_l is low, then $\mathfrak{R}_0(.)$ is large and this again contributes to high computational complexity. The proposed MUSIC based MeMBer filter

³A example MVDR image for a two-target case was shown in Fig. 3.9(a) in chapter 3.

requires the computation of only IN_k MUSIC values every time step. This feature makes the filter computationally superior to the MeMBer filter for beamformer images. In this work, for implementing the “MeMBer filter for beamformer images,” the complete image $\gamma_{(.)}$ is first constructed using MVDR beamforming and then thresholded using the illumination threshold I_l to obtain that part of the image $\gamma_{(.)}^{(x,y)}$ that is influenced by the target hypothesis. $\gamma_{(.)}^{(x,y)}$ is then fed into (4.11) to compute the likelihood.

4.6 Simulation results

All the simulations presented here consider tracking acoustic sources in a 2D environment which includes three echo paths with 1 second reverberation time for 60 dB decay. The targets are constrained to move with position 0m to x_{\max} m in a line located 10m from another parallel line containing the 5 sensors whose positions are chosen from $\mathcal{U}[0, x_{\max}]$. The state of the target is $\mathbf{x}_k = (x, v_x)^T$ where x is the position and v_x is the velocity. $(\cdot)^T$ denotes matrix transpose. The time evolution of \mathbf{x}_k is described as

$$\mathbf{x}_k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + q_{k-1} \quad (4.15)$$

where the duration between two observations is $T = 1$ s and the additive process noise $q_{k-1} \sim \mathcal{N}(0, Q)$ where the uncorrelated covariance matrix is

$$Q = \begin{bmatrix} 3 & 0 \\ 0 & 0.5 \end{bmatrix} \quad (4.16)$$

The sensor signals are obtained using the model described in (4.1). The sensor noise vector is $v_k \sim \mathcal{CN}(0, R)$ where R is a diagonal with a variance of $\sigma^2 = 0.1$ corresponding to each sensor.

Demonstration

The proposed MUSIC based particle-MeMBer filter for tracking single/multiple, appearing/disappearing and crossing targets is demonstrated in Fig. 4.2. It can be seen that the filter quickly locks on and tracks the targets with high accuracy. If a target disappears, the existence probability of the Bernoulli RFS becomes low, thereby eliminating the track. The use of the Markov transition prior as the proposal distribution is known to be sub-optimal and the performance of the proposed scheme could be enhanced using an im-

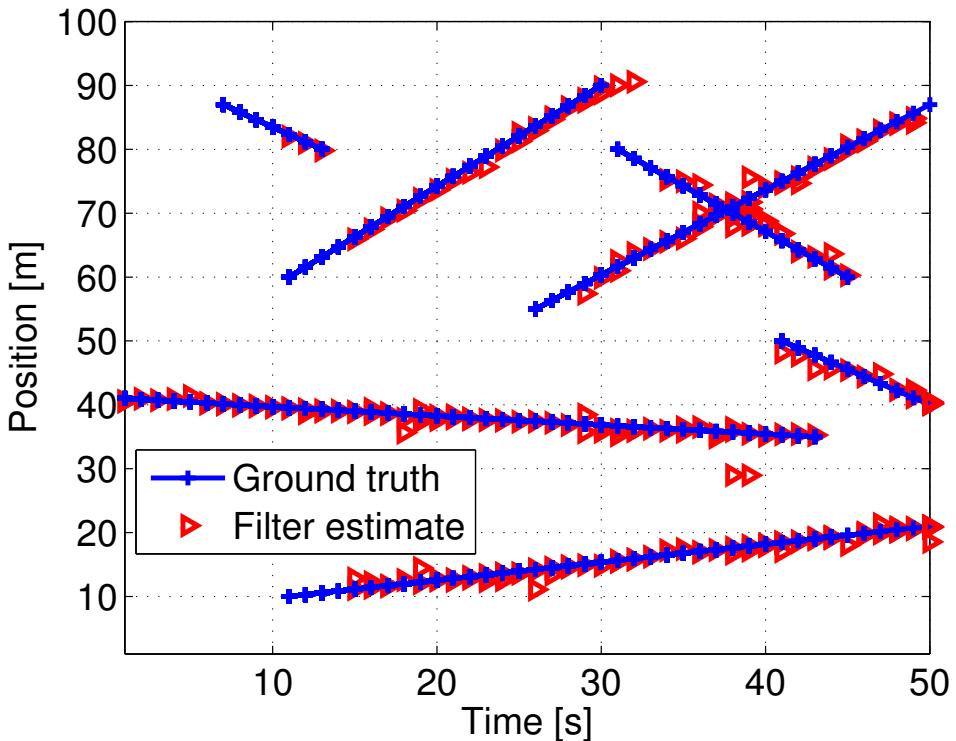


Figure 4.2: Filter estimates compared to the ground truth. The targets move along a 1D grid of $x_{\max} = 100$ m width. The number of particles that approximate the pdf of each element in the multi-Bernoulli RFS is $I = 1000$. $f = 4$ KHz and $\nu = 1500$ m/s. The SNR is 30dB. The Poisson mean of the number of new-born targets is $\lambda_B = 10$. The probability of target survival (chosen to be constant) is $p_S = 0.8$. The pruning threshold is $T_{\text{prune}} = 0.6$. The merging threshold is $T_{\text{merge}} = 1$ m. The filter runs for 50 time steps.

proved importance distribution. Nevertheless, it can be seen that the track initialisation and deletion are conducted very effectively.

For the multi-target scenario described in Fig. 4.2, the MMSE-based-MDL estimate used to evaluate the noise eigenvectors U_n used in (4.14) is shown in Fig. 4.3 and it can be observed that the technique provides an accurate estimate of the actual number of targets (even at low SNR levels).

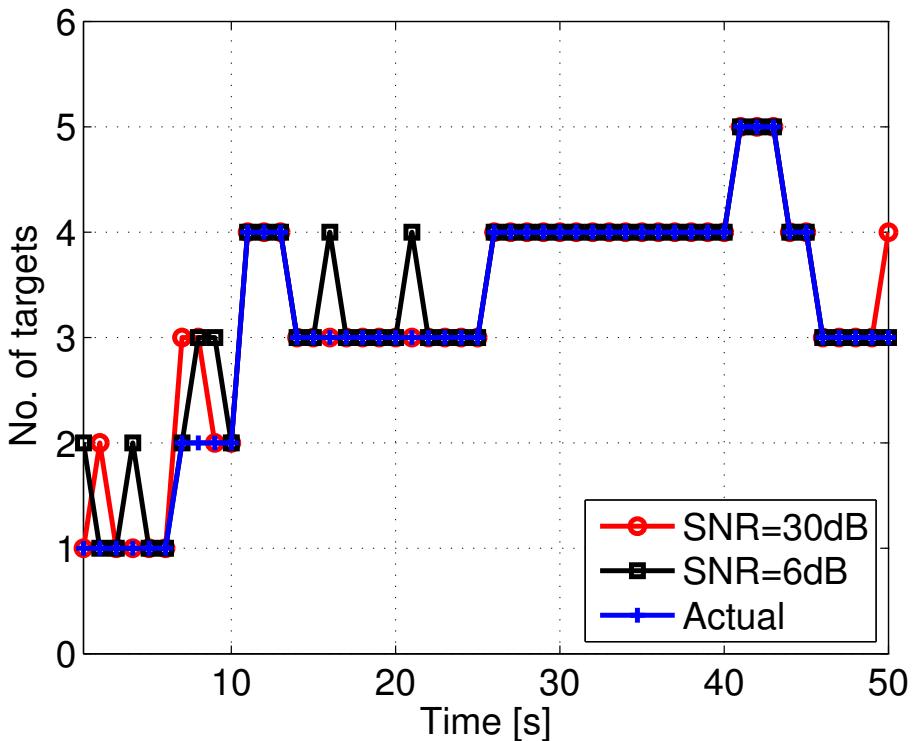


Figure 4.3: The MMSE-based-MDL estimate of the number of targets, versus time.

Demonstration of target resolving ability

Secondly, the ability of the proposed MUSIC based particle-MeMBER filter to effectively resolve close targets is demonstrated. For this, the scenario

4.6. SIMULATION RESULTS

described in Fig. 4.4 is used. Here, two targets cross each other twice. This motion is realistic for real-world scenarios where targets often move in approximately straight lines with constant velocity and occasionally make an abrupt manoeuvre [222].

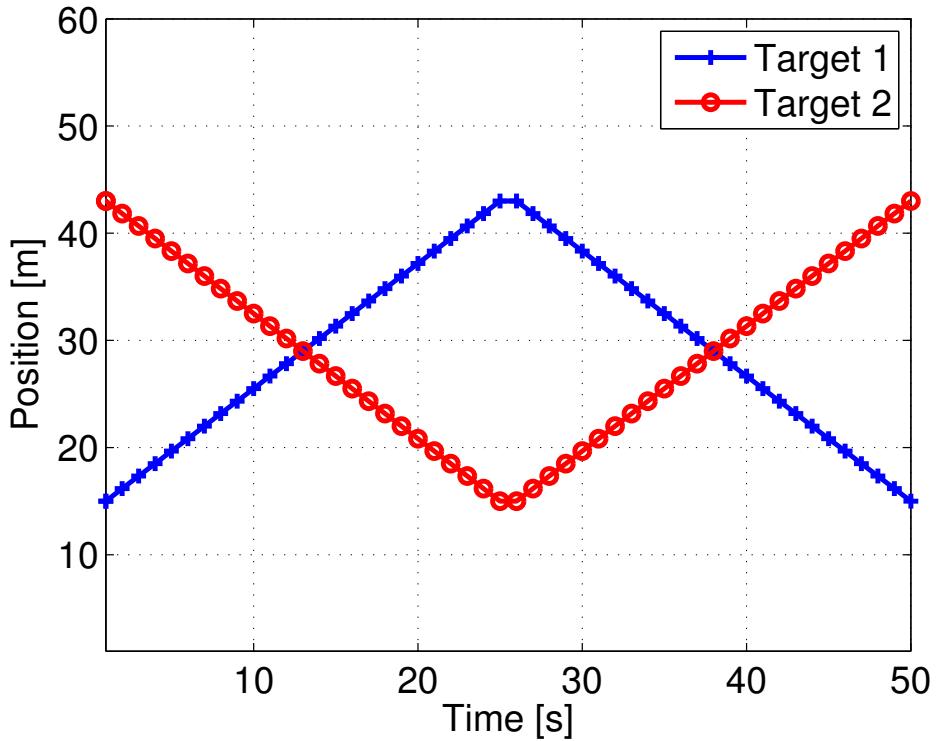


Figure 4.4: The ground truth of the two targets used to obtain the evaluation results in Fig. 4.5, Fig. 4.6, Fig. 4.7, Fig. 4.9, Fig. 4.10 and Fig. 4.14. The target positions coincide exactly at $k = 13$ and at $k = 38$.

In view of the fact that the PHD and the MeMBer filters cannot operate directly on sensor array data, for comparison with the proposed method, the acoustic signals are first converted into the minimum variance distortionless response (MVDR) beamformer image which is the observation for the MeMBer filter. For the PHD filter, the image is further thresholded to generate the finite set point measurement feed. The proposed MUSIC based MeMBer

filter, however, operates on the acoustic signals. The super resolution property of MUSIC aids in resolving close targets and keeping the tracks separate for a longer duration.

Fig. 4.5, Fig. 4.6 and Fig. 4.7 show the particle evolution (for the ground truth in Fig. 4.4) for the proposed MUSIC based MeMBer, the MeMBer for image and the PHD filters respectively. It can be observed that the proposed filter resolves close targets better than the others. Before the first crossing at $k = 13$, it can be seen that while the PHD and the MeMBer filters keep

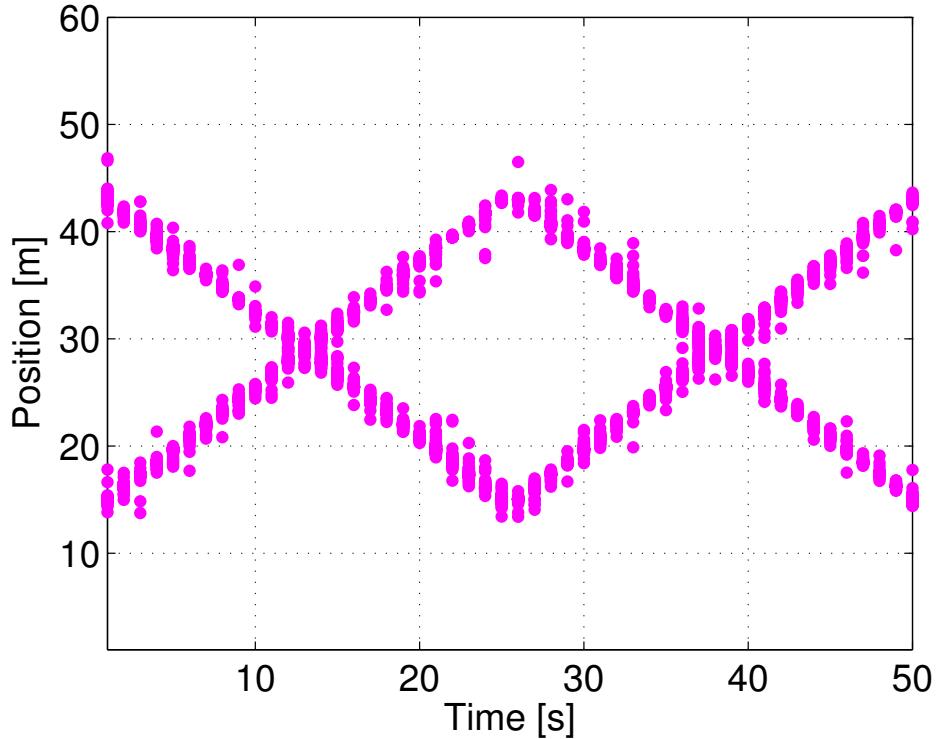


Figure 4.5: Evolution of the particles with time in the proposed MUSIC based particle-MeMBer filter. The targets move along a one dimensional grid grid of $x_{\max} = 60$ m width. The number of particles for each target state hypothesis is $I = 1000$. $f = 4$ KHz and $\nu = 1500$ m/s. The SNR is 30dB. For simplicity, it is assumed that the number of targets is known; hence $\lambda_B = 0$. The survival probability is $p_S = 0.8$. The pruning threshold is $\mathcal{T}_{\text{prune}} = 0.6$. The merging threshold is $\mathcal{T}_{\text{merge}} = 1$ m. The filter runs for 50 time steps.

the targets separate only until the 6th and 9th time samples respectively, the proposed filter keeps the targets separate until the 11th time step. After the second crossing at $k = 38$, the PHD and MeMBer filters resolve the targets only after $k = 46$ and $k = 44$ respectively, and the proposed MeMBer filter separates the targets from the 40th time sample.

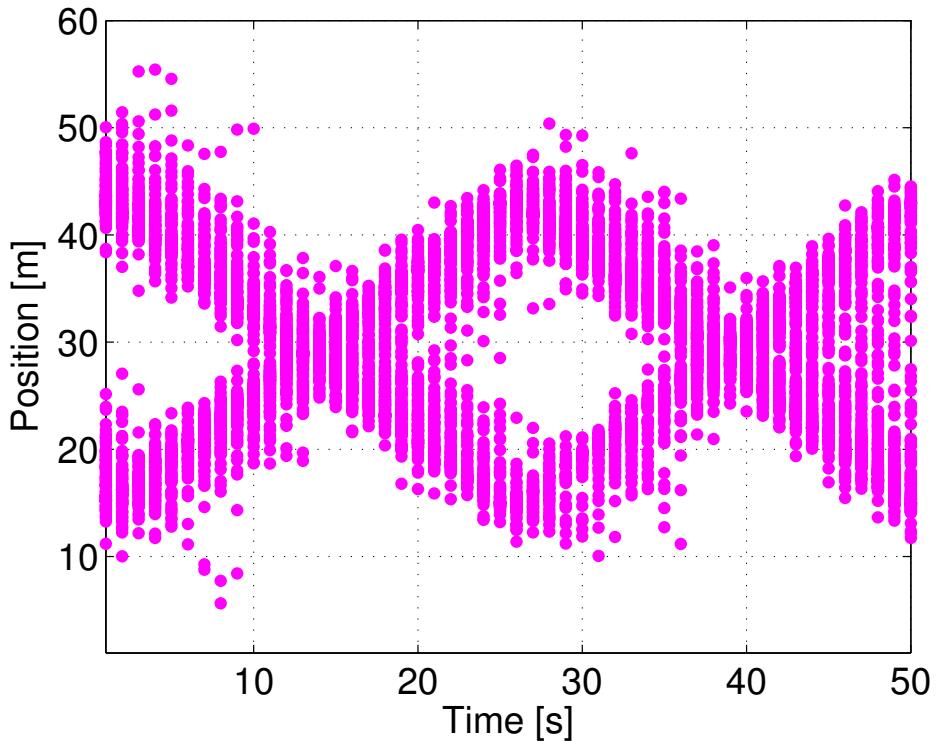


Figure 4.6: Evolution of the particles with time in the particle-MeMBer filter (for image data) [49, 196]. The system specifications described in the caption of Fig. 4.5 apply to this figure also. The illumination threshold I_l is chosen to be 0 so that the entire image is fed to the filter.

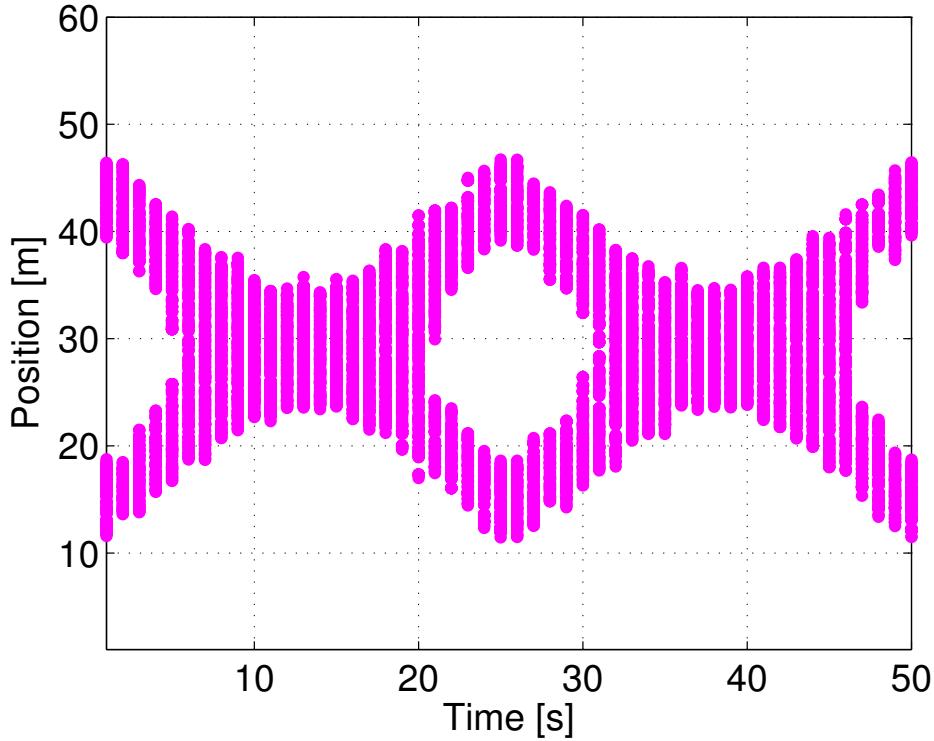


Figure 4.7: Evolution of the particles with time in the particle-PHD filter [272]. The system specifications described in the caption of Fig. 4.5 apply to this figure also. The detection probability is $p_D = 0.95$. Target birth is not considered. To obtain the point target measurement feed, the beamformer image is thresholded at 70% of the maximum peak. 1000 particles are used. To obtain state estimates, the k-means clustering approach of [169] is used.

To further illustrate the filters' target resolving ability, the normalised PHD function (evaluated over the entire 60m grid) for the proposed MUSIC based MeMBER, MeMBER and PHD filters is shown in Fig. 4.8. This PHD corresponds to the ground truth in Fig. 4.4 at the 11th time step. For the MeMBER filter, the PHD function is evaluated from (2.103) in section 2.7.1 of chapter 2. It can be observed that the proposed method successfully resolves the two targets while the other filters merge them inextricably (as is also seen in Fig. 4.5, Fig. 4.6 and Fig. 4.7).

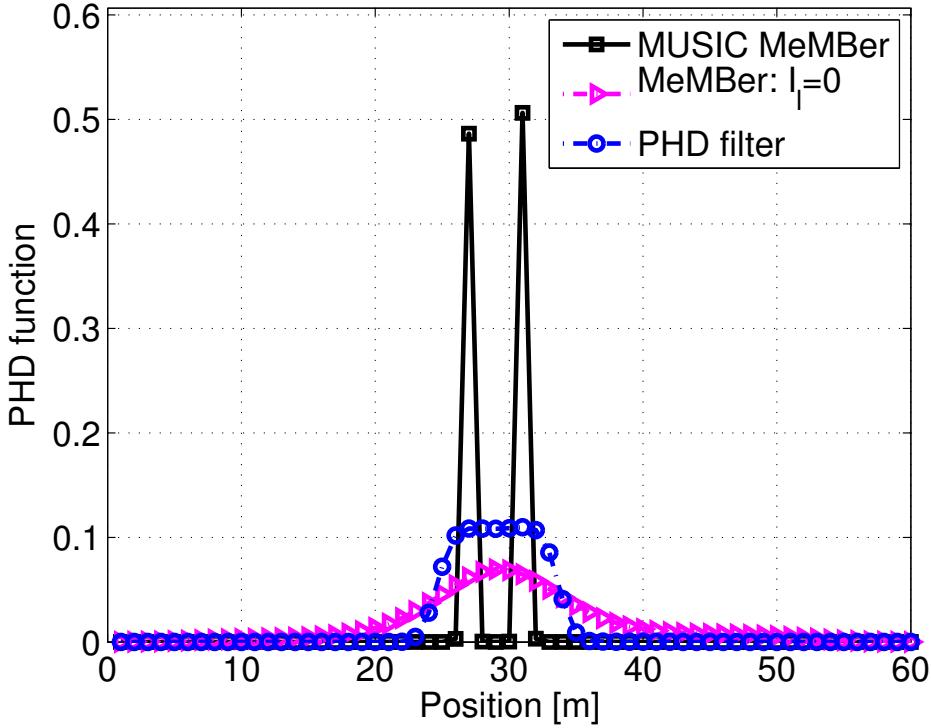


Figure 4.8: Normalised PHD function of the MUSIC based MeMBer, MeMBer and PHD filters at the 11th time step.

Evaluation of track accuracy

Thirdly, the accuracy of the filters (using the ground truth shown in Fig. 4.4) is tested. Fig. 4.9 shows the root mean square error (RMSE) for varying SNR levels. It can be observed that that the proposed technique is very accurate. Its performance advantage as against the PHD and the MeMBer filters can be attributed to the direct use of acoustic signals. At low SNR, spurious peaks in the beamformer image degrade the MeMBer filter performance. Moreover, its error increases with increasing I_l . The unwanted peaks at low SNR, in turn, increase the false alarms in the PHD filter observation set and adversely effects its performance. The joint filter operates using the true likelihood function in (4.2) with the number of targets fixed at

the correct value and performs better than the others. The MUSIC based MTT PF previously proposed in chapter 3 compares equally with the MUSIC based MeMBer filter although its error increases slightly at low SNR. This is because the former uses $\beta = 500$ particles to track one target while the latter uses $I = 1000$ particles.

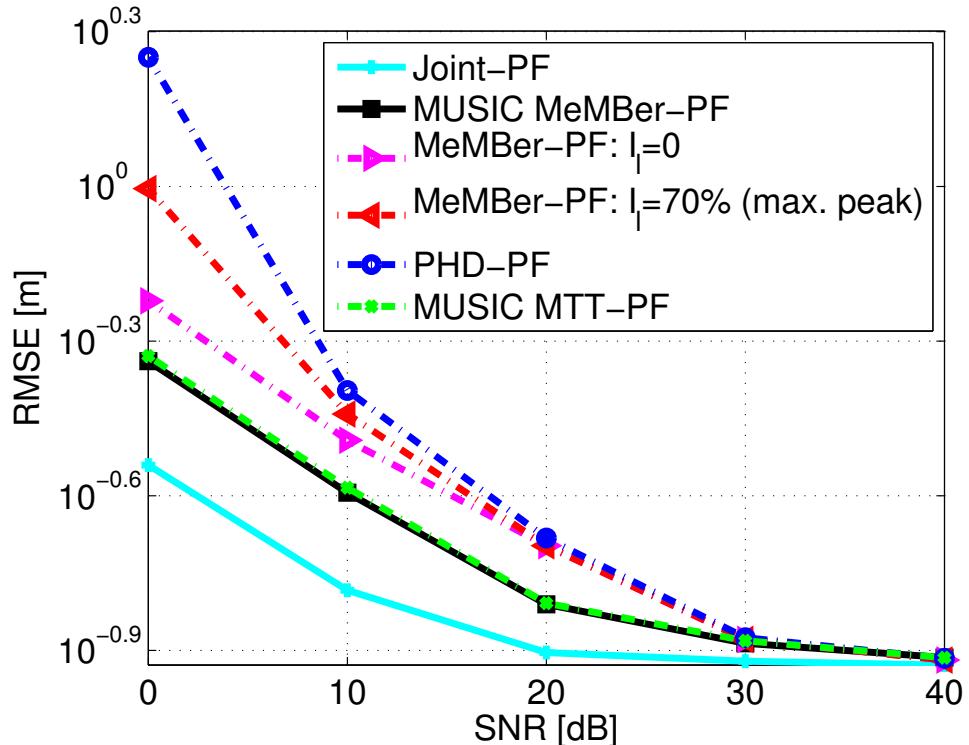


Figure 4.9: RMSE versus SNR. The specifications described in the caption of Fig. 4.5, Fig. 4.6 and Fig. 4.7 apply to this figure also. For the PHD filter, the beamformer image is thresholded at 70% of the highest peak. Since it is assumed that the number of targets is known, the joint filter operates in double the single target dimensionality. For the MUSIC based MTT PF previously proposed in chapter 3, the total number of particles is $I = 1000$, the number of clusters formed every time step is $C = 2$ and the number of particles in a cluster are limited to $\beta = 500$. Results are averaged over 20 trials, each having 50 time steps.

Evaluation of computational complexity

Fourthly, the complexity of the filters (using the ground truth in Fig. 4.4) is shown in Fig. 4.10 by measuring the computational time to evaluate one complete time step for varying numbers of particles. It can be observed that the proposed filter requires fewer computational resources than the MeMBer filter operating on the image data. This is because the MeMBer filter needs to construct $2I$ images, each having 60 pixels for the evaluation of (4.11) while the proposed technique requires only $2I$ MUSIC values. In the particle-PHD filter, the majority of the time is consumed while evaluating the beamformer image and hence the filter will be slower than the others when $I \lesssim x_{\max}$. In this simulation, $x_{\max} = 60\text{m}$ is width of the surveillance region. The MUSIC

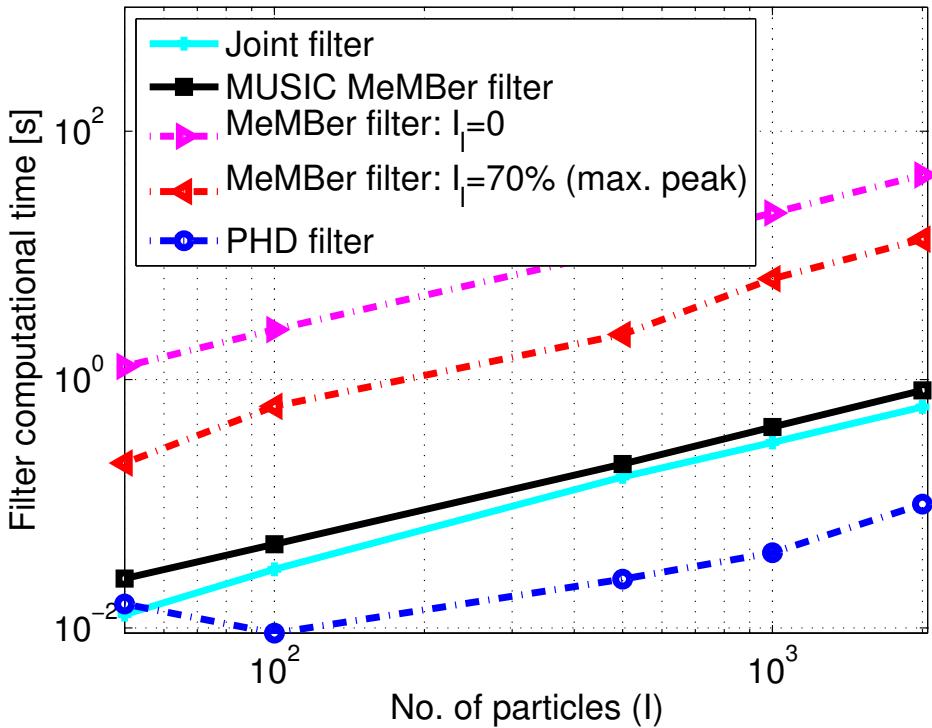


Figure 4.10: Time consumption for one time step versus I . The specifications described in the caption of Fig. 4.9 apply to this figure also.

based MTT PF exhibits low complexity requirements because it uses only I particles (in the dimensionality of a single target) to track the two targets while the proposed MUSIC based MeMBer filter uses $2I$ particles. The joint filter uses I particles of double the dimension, and its complexity becomes unfavourable for larger numbers of targets. This can be observed in Fig. 4.11. The figure shows the computational time taken to evaluate the true likelihood in (4.2) (used in the joint filter) and the MUSIC pseudo-likelihood in (4.14) (used in the proposed MUSIC based MeMBer filter) for 1000 particles while varying the number of targets. It can be observed that the true likelihood computation quickly becomes resource expensive. The high computational complexity of the proposed filter for the two target case in Fig. 4.10 can be attributed to merging and pruning operations. However, it can be seen from

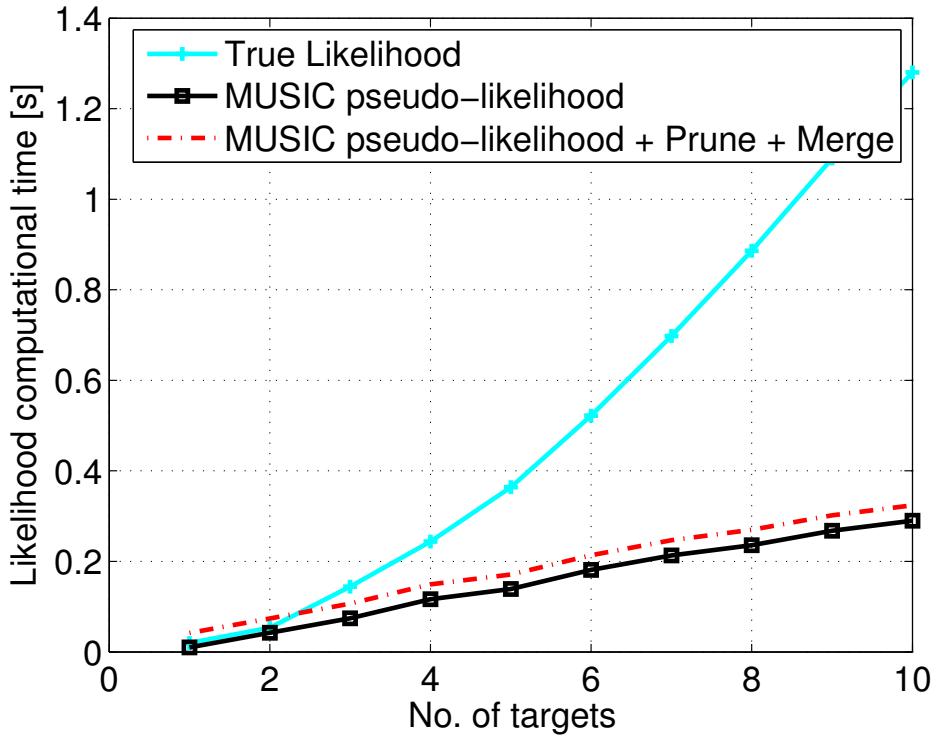


Figure 4.11: The computational time consumed in computing the likelihoods. The number of particles used is 500. The results are averaged over 10000 trials.

Fig. 4.11 that the computation cost of pruning and merging operation does not rise like the likelihood for the joint filter (here the number of new-born targets is $N_B \sim \text{Pois}(\lambda_B)$ with $\lambda_B = 18$).

4.6.1 Evaluation of MUSIC based MTT filters

Finally, the MUSIC based MTT filtering methods proposed in this thesis — the MUSIC based MTT filter of chapter 3 and the MUSIC based MeMBER filter of this chapter — are further analysed and tested. Firstly the effects of using MUSIC as a pseudo-likelihood are studied, and then the track accuracy of the methods for changing numbers of targets and harder target movements is evaluated.

Numerical analysis on MUSIC pseudo-likelihood

Given that the MUSIC based MTT PF of chapter 3 and the MUSIC based MeMBER filter of this chapter both use MUSIC as the pseudo-likelihood of a particle, it is valid to investigate the faithfulness of MUSIC in accurately representing the *true* likelihood. Fig. 4.12 shows the true likelihood and the MUSIC pseudo-likelihood evaluated over the entire 60m grid in the presence of a single target. It can be observed that the MUSIC pseudo-spectrum has a much narrower peak at the target location than that of the true likelihood function, and hence will provide a good estimate of the target location but a poor estimate of the location uncertainty. This is further shown by the corresponding cumulative distribution functions (CDFs) in Fig. 4.13. The narrow peak in MUSIC pseudo-spectrum causes its CDF to be truncated near the tails of the CDF of the true likelihood. This is shown using double arrows.

In a PF context, the narrow peaky nature of MUSIC will result in many replications of large weight particles during the resampling step, i.e., the particle cloud could be too constrained. This can be overcome using a auxiliary [135] PF approach or by introducing a few Markov chain Monte Carlo (MCMC) steps after resampling [127]. These approaches will increase the di-

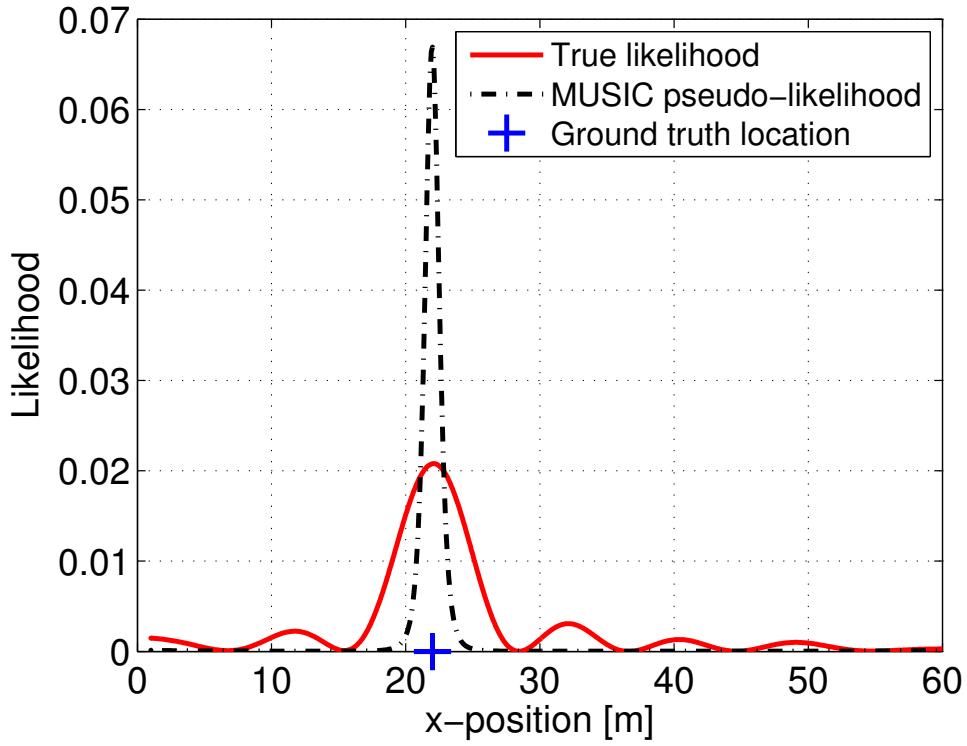


Figure 4.12: The true likelihood and MUSIC pseudo-likelihood spectrums. The target is exactly at 22m.

versity among particles. Nevertheless, the particle cloud can easily follow targets if the process covariance Q is chosen such that its norm $\|Q\| = \sum_{i,j} Q_{i,j}^2$ is large.

If a single target scenario is considered with no target appearance or disappearance, the MUSIC based MTT PF of chapter 3 and the MUSIC based MeMBER filter of this chapter are simply single target PFs, except that the weight of a particle is now the MUSIC pseudo value. Now, the faithfulness of the MUSIC pseudo-likelihood based single target PF is numerically tested in terms of its conformity with the true-likelihood based single target PF. For this numerical testing, the well known Kolmogorov-Smirnov (KS) statistic [62] is used. The KS test has the advantage of providing a reliable measure

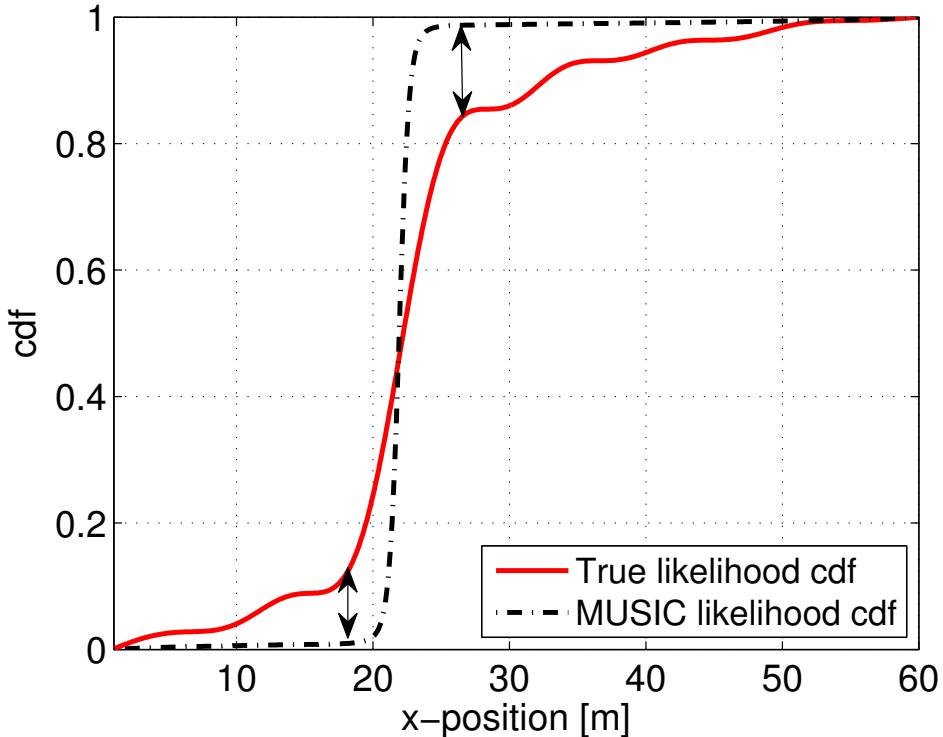


Figure 4.13: CDFs for the normalised results in Fig. 4.12. The locations of maximum deviation are shown with the arrows.

of the accuracy of the estimate of the posterior. The classical one-sample KS test evaluates the misfit between the CDFs of two univariate ordered datasets [273] by computing the largest absolute difference. For the KS testing in multi-dimensional models [274], the two-sample KS test [275, 276] operates by comparing the CDFs from two multi-dimensional datasets which are ordered in some fashion. One method (as used here) is to compare the CDFs of the two samples with all combinations of sorting, and take the largest of the set of resulting KS statistics. KS testing for multi-dimensional data has been used previously for other PF problems [277]. For the Markovian process model described in this section, and considering one target from the ground truth shown in Fig. 4.4, Fig. 4.14 shows the two-sample KS statistic misfit of the MUSIC pseudo-likelihood based PF with the PF that operates using

the true likelihood equation in (4.2). The KS values for the PF in which the likelihood is the MVDR value is also shown. Since the MUSIC and MVDR functions are not identical to a true likelihood, there is considerable disagreement in the two filter representations. Although the disagreement decreases with increasing I , it can be observed that the statistic begins to exhibit asymptotic behaviour from $I \approx 500$.

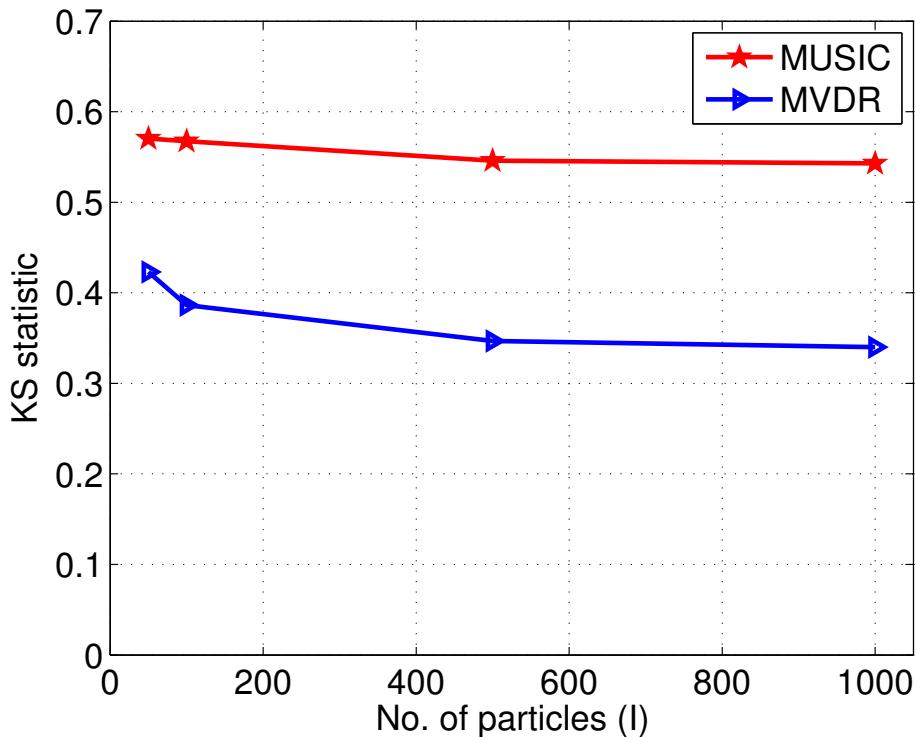


Figure 4.14: The KS statistic disagreement [274] versus the number of particles. The ground truth shown in Fig. 4.4 is used, except that now only one target is considered. The specifications in the caption of Fig. 4.5 apply to this figure also. The results are averaged over 20 trials, each comprising 50 time steps.

Evaluation of track accuracy and target number estimation

Here, the track accuracy of the MUSIC based MTT filters is evaluated for changing numbers of targets, i.e., target appearance and disappearance, and

for harder target movements. For this study, the 2D state model used in section 3.7 of chapter 3 is employed. The target state is defined as $\mathbf{x}_k = (x, y, v_x, v_y)^T$ where x, y are the positions and v_x, v_y are the velocities along the x, y -axis. \mathbf{x}_k evolves with time in accordance to a Markov model in which the position is updated using a constant velocity model and the velocity is updated using a random walk. The process noise is zero-mean Gaussian distributed with uncorrelated covariance matrix having a variance of 0.5 along each dimension. The specifications are as follows: *(i)* the surveillance region is a 4×4 2D space which includes a one second reverberation time for 60 dB decay, *(ii)* a linear array of 5 sensors, located at an offset of 10m along the third dimension from the grid of target movement, is used, i.e., the sensor positions are $x = 1, y = \{1, 2, 3, 4, 5\}$ and $z = 10$, *(iii)* the frequency of operation is $f = 4$ kHz, *(iv)* the propagation velocity is $\nu = 1500$ m/s, and *(v)* the total number of time steps is 50.

In this testing, two MTT scenarios are used. The ground truth used for the first scenario is shown in Fig. 4.15 (and also in Fig. 4.16). The actual target paths are plotted as solid lines in different colors. In this MTT scenario, two targets — the ones depicted using green and black solid lines — travel close to one another with nearly opposite velocities during times $k = 34, 35, 36$. Three targets — the ones depicted using red, cyan and blue solid lines — travel very close to one another approximately during times $k = 35, \dots, 42$. The targets represented in blue and red make a sudden manoeuvre nearly at time $k = 45$. The ground truth used for the second scenario is shown in Fig. 4.17 (and also in Fig. 4.18). The target paths are again plotted as solid lines in different colors. In this MTT scenario, two targets — the ones depicted using red and black solid lines — travel close to one another approximately during times $k = 19, \dots, 22$. Three targets — the ones depicted using blue, red, and cyan solid lines — travel close to one another approximately during times $k = 26, \dots, 32$. Two targets — the ones depicted using green and black solid lines — travel close to one another approximately during times $k = 35, 36$. The targets represented in red and

blue travel with opposite velocities when travelling close to each other. The target represented in black makes a u-turn during times $k = 29, \dots, 35$. In both the examples, it is ensured that two targets are not at the same position at the same time. The target starting position is indicated by a circle and the time of the target appearance is noted next to the circle. The final position of the target before its disappearance is indicated by a star and the time step at which the target reaches this position is noted next to the star. Additionally, Fig. 4.15 and Fig. 4.17 demonstrate the operation of the MUSIC based MTT PF of chapter 3 while Fig. 4.16 and Fig. 4.18 demonstrate the operation of the MUSIC based MeMBer filter of this chapter. The fil-

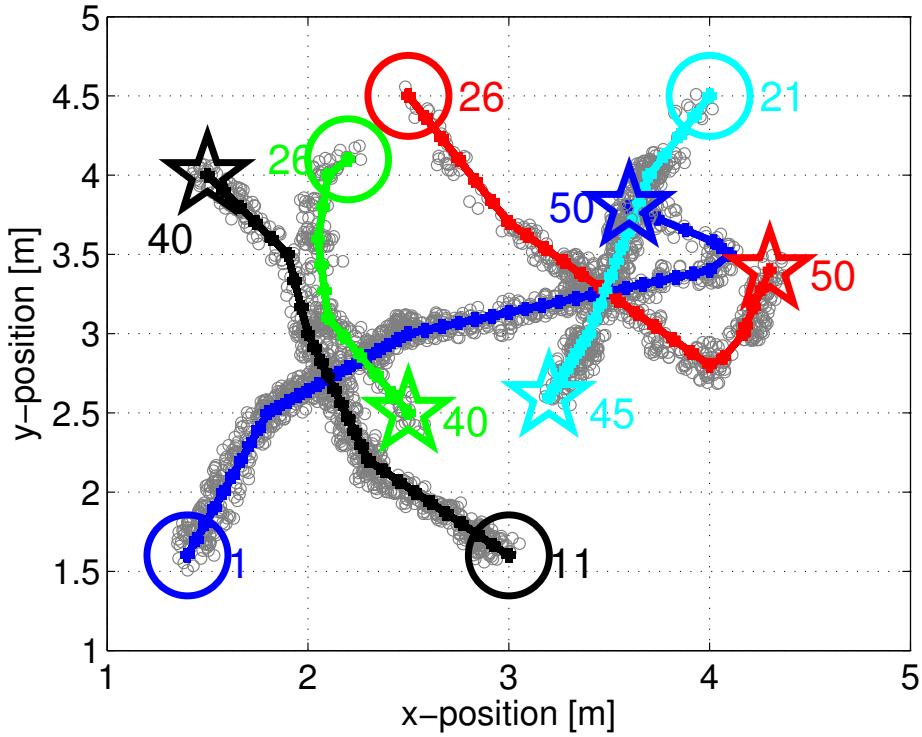


Figure 4.15: The first MTT scenario and the evolution of particles in the MUSIC based MTT PF of chapter 3. The SNR is 27dB. In the filter, the number of particles $I = 5000$, the number of clusters is $C = 8$, the number of particles per cluster is regulated to $\beta = 500$, and the cluster looseness threshold to initiate and remove tracks is $\exp(-T_c) = 0.75$.

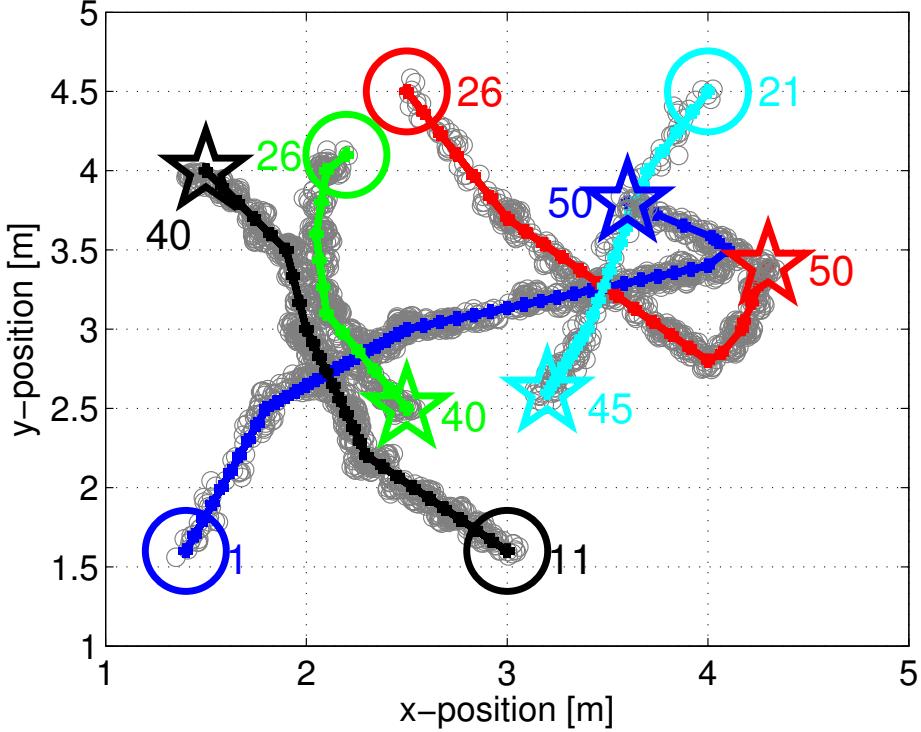


Figure 4.16: The first MTT scenario and the evolution of particles in the MUSIC based MeMBER filter of this chapter. The SNR is 27dB. In the filter, the survival probability is $p_s = 0.95$, the number of new-born targets is $N_B \sim \text{Pois}(\lambda_B)$ with $\lambda_B = 10$, the total number of particles per Bernoulli set is $I = 1000$, the pruning threshold is $\mathcal{T}_{\text{prune}} = 0.65$, and the merging threshold is $\mathcal{T}_{\text{merge}} = 1\text{m}$.

ters are demonstrated by showing the evolution of particles. These particles are plotted as grey circles. While demonstrating the MUSIC based MTT PF, the figures shows only those particle clusters whose cluster looseness is $\exp(-T_c) \geq 0.75$. To overcome the problem of a too constrained particle cloud due to the narrow peaky nature of MUSIC, 7 MCMC transitions are introduced [127] for each cluster after the regulation step. This slightly increases the diversity of the particles. It can be seen from the figures that the MUSIC based MTT filters *(i)* accurately track the targets, *(ii)* quickly detect new-born targets, and *(iii)* successfully release lock of disappearing

targets. It can also be observed that the track continuity is successfully maintained by weighting the velocity component to be (in this example, three times) higher than the position component. Simulation videos demonstrating the filters' implementation for both the MTT scenarios are available at <http://ecs.victoria.ac.nz/Groups/CSP/PublicationExtras>.

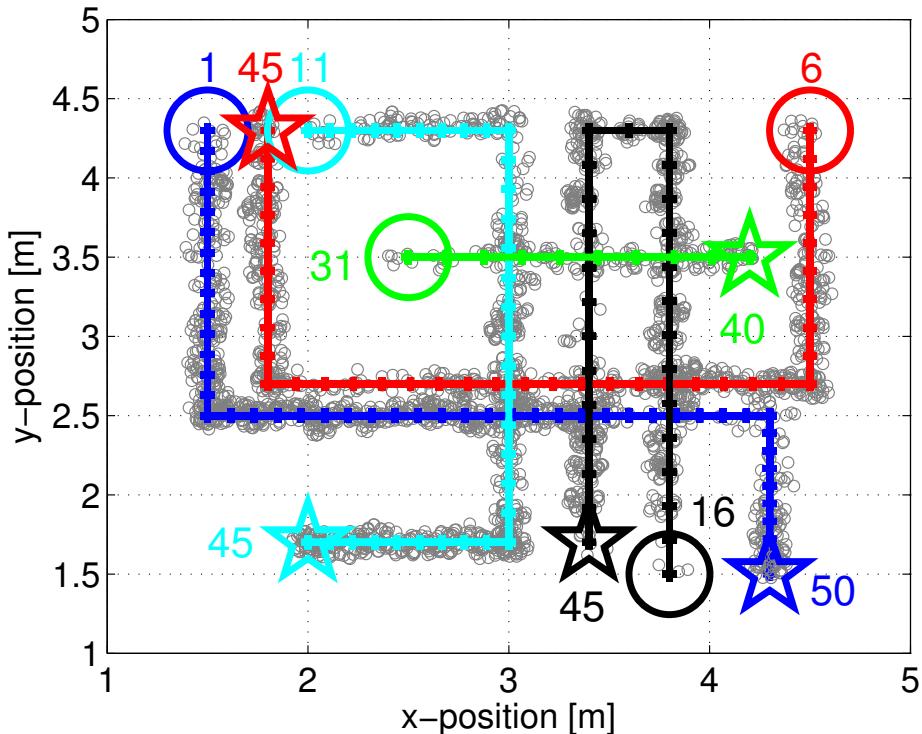


Figure 4.17: The second MTT scenario and the evolution of particles in the MUSIC based MTT PF. The SNR is 27dB. The filter specifications listed in the caption of Fig. 4.15 apply to this figure also.

The track accuracy of various MTT filters for the first MTT scenario given in Fig. 4.15 (and Fig. 4.16) and the second MTT scenario given in Fig. 4.17 (and Fig. 4.18) is shown in Fig. 4.19 and Fig. 4.20 respectively. The figures show the RMSE of the MUSIC based MTT filters proposed in this thesis and other state-of-the-art Bayes' MTT filters, for varying SNR

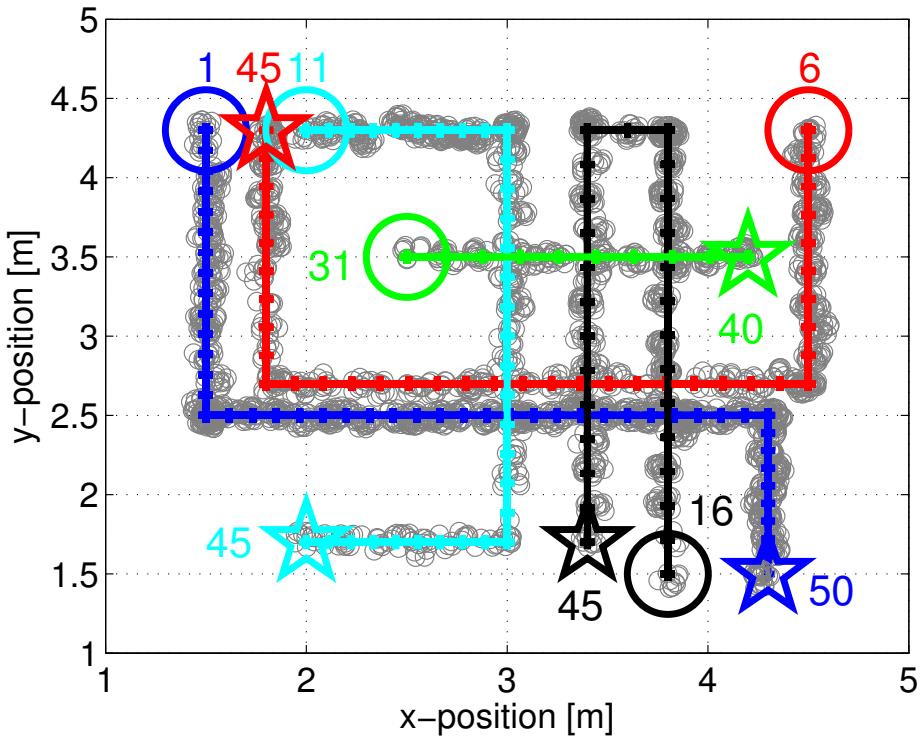


Figure 4.18: The second MTT scenario and the evolution of particles in the MUSIC based MeMBER filter. The SNR is 27dB. The filter specifications listed in the caption of Fig. 4.16 apply to this figure also.

levels. Error bars that represent a 90% confidence interval are also shown. It can be observed that the results are consistent with those in Fig. 3.11 and Fig. 4.9. Except for the joint filter, no other filter assumes knowledge about the actual number of targets. The joint filter provides a best possible target state estimate when the actual number of targets is known and hence can be used as a reference for analysing the performance of the other MTT methods. In the other filters, the error is computed from the time the filter detects a target. In the RFS filters — the PHD and the MeMBER filters — the MVDR beamformer image is evaluated over the observation space with a resolution of 0.1m. From both the figures, it can be observed that joint filter exhibits the superior performance. The performance of the RFS filters drops heavily

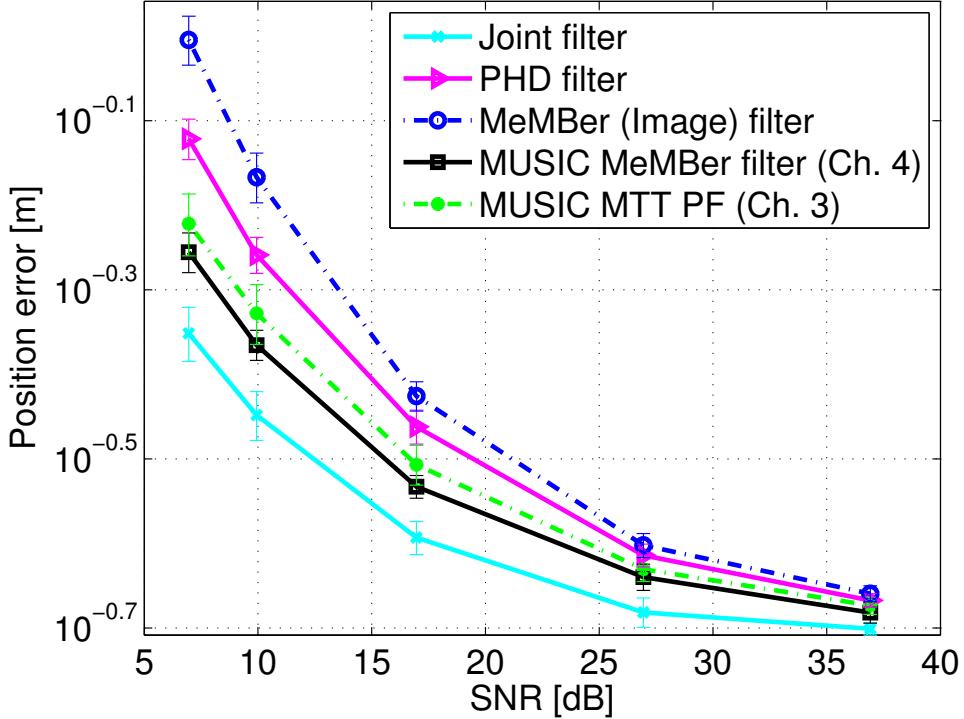


Figure 4.19: RMSE versus SNR for the first MTT scenario. The results are averaged over 100 trials, each having 50 time steps. The filter specifications are as follows. The joint filter operates on $I = 1000$ particles with dimensionality equal to the number of targets. The PHD filter: (i) the detection and survival probabilities are $p_D = 0.9$ and $p_D = 0.95$ respectively, (ii) the number of new-born targets is $N_B \sim \text{Pois}(\lambda_B)$ with $\lambda_B = 10$, (iii) the Poisson mean of the number of false alarms is $\lambda_F = 10$, (iv) the point target detections are obtained by thresholding the beamformer image at 70% of its maximum peak, (v) the total number of particles is $I = 1000$, and (vi) the state estimates are obtained using the k-means clustering approach of [169]. For the MeMBer filter for image data: (i) the survival probability is $p_S = 0.95$, (ii) the number of new-born targets is $N_B \sim \text{Pois}(\lambda_B)$ with $\lambda_B = 10$, (iii) the beamformer image is thresholded at 70% of its maximum peak while computing the likelihood ratio, (iv) the total number of particles per Bernoulli set is $I = 1000$, (v) the pruning threshold is $T_{\text{prune}} = 0.65$, and (vi) the merging threshold is $T_{\text{merge}} = 1\text{m}$. For the MUSIC based MeMBer filter proposed in this chapter: the specifications of the MeMBer filter for image data apply to this filter also, except that beamforming is not required. For the MUSIC based MTT PF of chapter 3: (i) the total number of particles is $I = 5000$, (ii) the number of clusters is $C = 8$, (iii) the maximum number of particles in a cluster is $\beta = 500$, and (iv) the cluster looseness threshold to initiate and remove tracks is $\exp(-T_c) = 0.75$.

4.6. SIMULATION RESULTS

with decreasing SNR: in the PHD filter this is due to increased number of false alarms and missed detections caused by the thresholding process, and in the MeMBER filter this is due to the spurious peaks in the image. Due to the avoidance of thresholding, the MeMBER filter exhibits better performance than the PHD filter. It can be seen that both the MUSIC based MTT filters display better performance than the RFS filters and give results close to that of the joint filter. This is due to a more direct use of the array data by virtue of using MUSIC. The reduced track accuracy of the MUSIC based MTT PF as compared to that of the MUSIC based MeMBER filter could be attributed to the reduced number of particles assigned to a target — the MUSIC based MeMBER filter uses 1000 particles per target while the MUSIC based MTT PF uses only 500 particles per target.

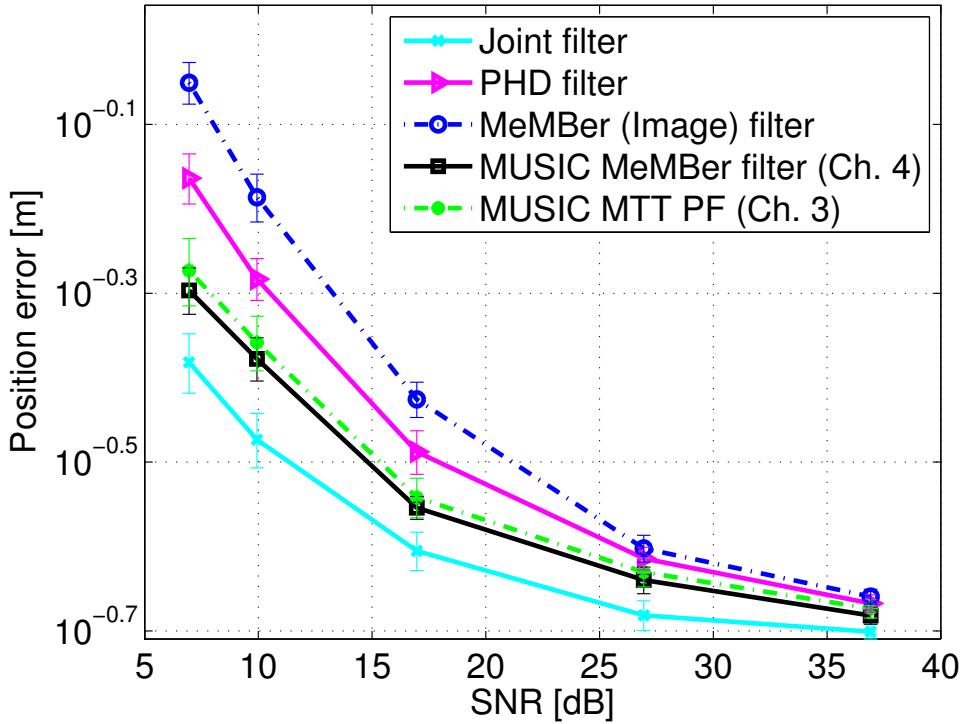


Figure 4.20: RMSE versus SNR for the second MTT scenario. The details mentioned in the caption of Fig. 4.19 apply to this figure also.

Now the ability of the MUSIC based MTT PF of chapter 3 to accurately estimate the number of targets is evaluated. Figs. 4.21 and 4.22 show the MUSIC based MTT PF estimate of the number of targets versus time, for the first and second MTT scenarios shown in Figs. 4.15 and 4.17 respectively. The target number estimate is computed by finding the number of clusters that declare target presence, i.e., the number of clusters whose $\exp(-T_c) \geq 0.75$. The additional MCMC transitions are not considered in this test. It can be seen that the filter provides an accurate estimate of the number of targets even at low SNR. A possible reason for the over-estimation of the target number at low SNR when multiple targets travel in close proximity could be

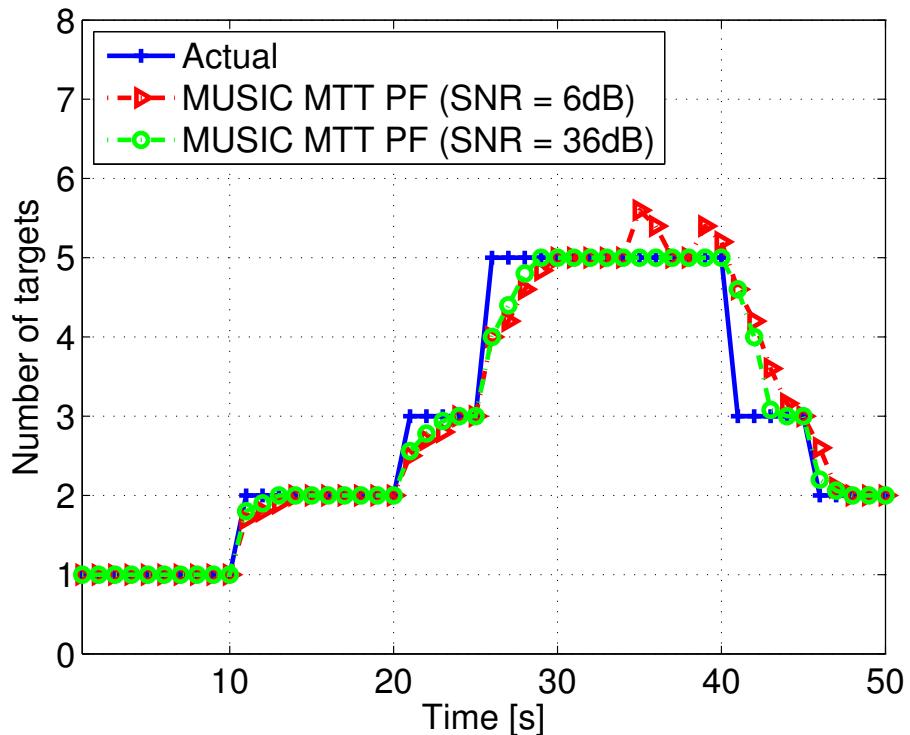


Figure 4.21: The MUSIC based MTT PF estimate of the number of targets, versus time, for the first MTT scenario. The results are averaged over 100 trials, each having 50 time steps. The filter specifications described in the caption of Fig. 4.15 apply to this figure also.

due to (i) the over-estimation by the MMSE based MDL method, or (ii) a free cluster near multiple tight clusters being influenced by the resampling process to declare target presence. Nevertheless, it can be observed that the problem does not persist in time as clustering and regulation are conducted at every time step.

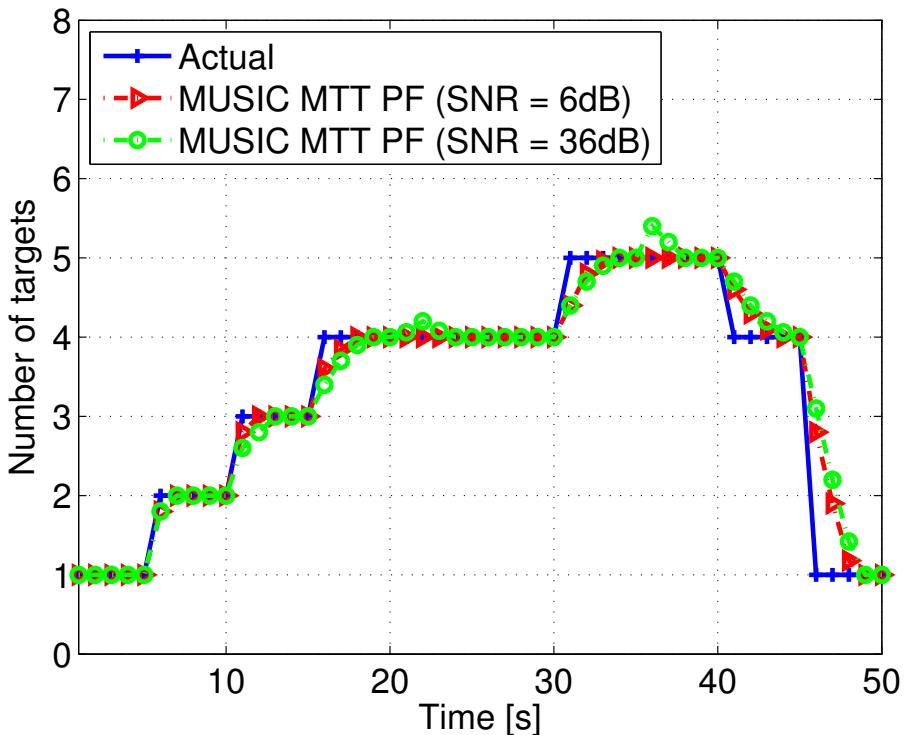


Figure 4.22: The MUSIC based MTT PF estimate of the number of targets, versus time, for the second MTT scenario. The results are averaged over 100 trials, each having 50 time steps. The filter specifications described in the caption of Fig. 4.15 apply to this figure also.

4.7 Conclusion and summary

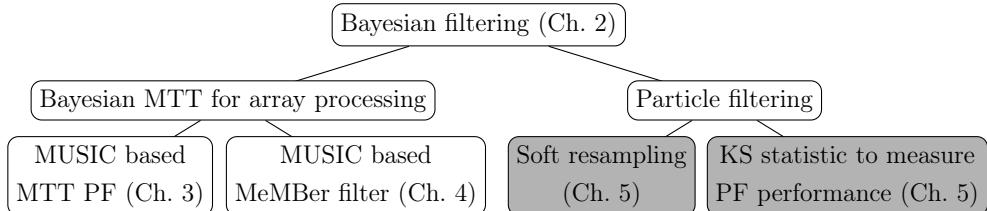
This chapter addressed the problem of Bayesian RFS filtering for phased sensor array observation models. State-of-the-art RFS filters — the PHD

and the MeMBer filters, in spite of overcoming the high computational complexity of the theoretically accurate joint filter, do not possess a suitable framework to operate directly on the phased array data and instead operate on images derived from the data. Converting the phased array data causes substantial information and resolution loss.

This chapter proposed a MUSIC based MeMBer filter, the key innovation of which is to use for the likelihood in the “MeMBer filter for image observations,” the MUSIC pseudo-spectrum — a function which has as its domain the state space of a single target. The advantages of the developed filter are; a) a more direct use of the array data resulting in improved track accuracy at low SNR, b) low computational complexity by replacing the likelihood ratio with the MUSIC value, and c) effective resolution of close targets by the super-resolution property of the MUSIC. The validity and merits of the proposed method were demonstrated using simulations. Referring back to section 1.3.1 of chapter 1, it can be stated that the research goal 2 that aims to overcome problem 2 has been successfully achieved by the proposed MUSIC based MeMBer filter.

5

Soft resampling



The particle filter (PF) [54], described in section 2.5 of chapter 2, is a prominent Bayesian approximation algorithm that is of key interest to this dissertation. Although the PF has attracted great interest from the research community, it is not without limitations. The PF implementation of the multiple signal classification (MUSIC) based multiple target tracking (MTT) proposed in chapter 3 and the MUSIC based multi-target multi-Bernoulli (MeMBer) filtering proposed in chapter 4 for array signal processing applications provided further motivation to address several limitations of PF operation.

The Bayes' filter [10, 17, 35] aims to recursively compute the posterior

probability distribution function (pdf) of the state of the target(s) [44]. However, it is known that Bayes' filtering is intractable [56] due to high dimensional integration and this intractability is overcome using approximate solutions [32]. The PF [21] approximates the Bayes' posterior pdf by a set of weighted independent point target representations, which in this context are called *particles*. Particle filtering involves an intermediary *resampling* step [118–121] in which particles with lower weights are replaced by copies of those with large weights. The major drawback in PF resampling, as identified in problem 3 of section 1.3.2 in chapter 1, is that the process results in loss of the information contained in those particles having low weights and consequently the effectiveness of the PF is affected if these low weights contain potential target information. The research goal 3 in section 1.3.2 of chapter 1 — to develop a resampling scheme that preserves more of the information contained in low weight particles and ensure that the accuracy of the posterior approximation of the developed resampler is not affected — aims to overcome this drawback. This chapter presents the development of this novel resampling approach and illustrates how the goal is achieved.

The resampling step in the PF involves rejection and replacement of particles based on their weights. This causes major adjustments to the information contained in the final particle set representation. Hence, as identified in problem 4 of section 1.3.2 in chapter 1, a numerical measure of the resampler performance is required. The research goal 4 in section 1.3.2 of chapter 1 that deals with this requirement aims to propose a quantifiable measure for assessing the faithfulness of the PF in accurately representing the posterior. A numerical test that achieves this goal is presented and used in this chapter.

The chapter is organised as follows: The related work in state-of-the-art resamplers with regard to information retention is outlined in section 5.1. After stating the problem in section 5.2, the proposed resampling method and its modification are presented in section 5.3 and section 5.4 respectively.

In section 5.5, the numerical test for assessing the resampler performance is described. The evaluation results are presented in section 5.6 and concluding remarks are given in section 5.7.

5.1 Related work

The PF [110], described in sections 2.5.3 and 2.5.4 of chapter 2, can be regarded as a technique that uses two fundamental approaches in succession. The first, *sequential importance sampling (SIS)*, specifies the process of drawing new particles at each time step based on a state equation [52, 127, 135] and updating their weights. The weight of a particle gives a probabilistic measure of the correctness of a hypothesis (particle). By itself, SIS results in a large variance of the weights and this ultimately results in one particle having all the weight — an inefficiency known as *degeneracy*. This degeneracy is overcome using a second stage, *resampling*, that replaces the lower weight particles by those with larger weights [119, 120].

The focus of this chapter is in PF resampling. Resampling techniques can be classified broadly into; a) stochastic, and b) deterministic. A few popular state-of-the-art resampling strategies are outlined in section 2.5.4 of chapter 2. The *stochastic resamplers*, namely the multinomial [241], stratified [242], systematic [52, 136] and residual [53] resamplers, duplicate the particles in proportion to their weights. This could result in the dismissal of moderate/low weight particles along with their weights. That is to say, stochastic resampling could result in loss of the information contained in a *potential low weight particle* — a particle that is gradually gaining weight either because the target is making a turn towards it or it is in the process of declaring a covert target. This problem is more evident in residual resampling because the method *always* discards the low weight particles (refer to its operation in section 2.5.4 of chapter 2). Moreover, the stochastic resamplers reset all the weights to the same number. The implication that every resampled particle is equally valid in representing the posterior pdf of the

target state is only asymptotically true. Deterministic resampling schemes [120, 122], on the other hand, are threshold-based. Partial resampling [120] proposes to keep the set of moderately weighted particles unchanged and resample the set of dominant and negligible particles. However, the reweighting is now based on the size of the sets but not on the individual weights. This causes loss of potential information contained in the negligible particles.

The detection of very covert targets using a PF relies on the accumulation of information over many consecutive observations, i.e., a covert target should be tracked for some time before declaring its presence. The contemporary resampling procedures discussed above impede this accumulation. In scenarios where particles are searching for new (very covert) targets, state-of-the-art resamplers quickly create fairly tight clusters so that there are not enough particles to span the entire observation space by the time a target appears, that is to say, the standard PF operating on state-of-the-art resamplers believes hastily: a characteristic that is unsuitable for the detection of covert targets. Hence the contemporary resamplers can be termed “hard resamplers.” To overcome this lack of information retention in a standard PF, the track-before-detect (TBD) PF [55, 56, 133], described section 2.5.7 of chapter 2, has been proposed. The target state birth and death modelling in the TBD PF aids in information retention. When the target is absent, most particles in the TBD PF are non-existent (dead) and take the maximum weight, i.e., if the existence parameter of the i th particle is $E_k^i = 0$, then its weight is $w_k^i = 1$. Hence using any contemporary resampler will result in *many* replications of these non-existent particles. The *few* existing particles whose $E_k^i = 1$ (the proportion of particles whose $E_k^i = 1$ is defined in the transitional probability matrix) remain in a loose cloud. When the target appears and start manoeuvring, the dead particles gradually come into existence and their weights will now be their likelihood of representing the true target, i.e., if $E_k^i = 1$, then w_k^i will be the weight update equation. That is to say, the higher the likelihood of the particle x_k^i in hypothesising the target correctly, the longer its existence parameter $E_k^i = 1$ and hence the stronger

the belief that x_k^i is tracking a target. The ability of the TBD PF in detecting new targets is dependent on the total number of particles I ; using fewer particles results in an insufficient number of particles to span the observation space for new targets, but using more particles increases computational load.

5.2 Motivation

It is noteworthy that when the number of particles I is large, the posterior distribution represented by the PF using any state-of-the-art resampler discussed above will be an accurate representation of the true distribution [116, 117, 222]. However it is usually the case that I is limited. The first objective in this chapter is to develop a “soft resampling” scheme that; a) aids in improved information accumulation over time, b) preserves the information contained in low weight particles, and c) thereby enhances the posterior representation with fewer particles. The motivating idea to achieve this objective is to reduce the variance between the original and the resampled weights by the process of proportional redistribution (also termed replication) and reweighting of particles, i.e., the particles are replicated in proportion to their weights, and the weights are reweighted in proportion to the number of replications. The aim is to achieve this without affecting the accuracy in representing the true posterior.

The second objective in this chapter is to propose a numerically calculable measure to assess the PF approximation performance and test the faithfulness of the proposed resampler in accurately representing the posterior. The motivating idea to achieve this objective is to use the well-known Kolmogorov-Smirnov (KS) statistic [273] which is a standard test to compare distributions. Using the Kalman filter [20] as a reference, the KS misfit between the *optimal* Kalman filter distribution and the PF representation could be computed.

5.3 Soft resampling

In this section, the soft resampling algorithm [61] is proposed. The conventional resamplers replicate the particles in proportion to their weights, i.e., the number of replications of a particle x_k^i will be¹ $n_k^i = \lfloor Iw_k^i \rfloor$. Then the total number of replications is $K = \sum_{i=1}^I n_k^i$ and generally $K < I$. Hence the information contained in the $I - K$ low weight particles whose $n_k^i = 0$ for $i = K + 1, \dots, I$, and their weights, does not appear in the resampled particle set. Losing this information could be problematic if the target is covert or is making a sharp manoeuvre towards a low weight particle.

Soft resampling addresses this problem. The key idea is to reduce the variance between the actual and the resampled particle weights, i.e., for unbiased resampling, we want

$$\mathbb{E} \left[\sum_{\forall j, 1 \leq j \leq I : x_k^{*j} = x_k^i} w_k^{*j} \right] = w_k^i \quad (5.1)$$

where $\{x_k^i, w_k^i\}_{i=1}^I$ and $\{x_k^{*j}, w_k^{*j}\}_{j=1}^I$ respectively is the weighted particle set before and after resampling. The method to achieve (5.1) is as follows: After the prediction and weight update steps, all the I weights are normalised and sorted in descending order. Then for each index $i = 1, \dots, I$, the number of replications of the i th particle x_k^i is given by

$$\Gamma_{x_k^i} = \max\{1, \lfloor Iw_k^i \rfloor\} \quad (5.2)$$

This indicates that a particle whose $\lfloor Iw_k^i \rfloor \geq 1$ will be replicated according to its weight (i.e., a multinomial selection) and a particle whose $\lfloor Iw_k^i \rfloor < 1$ will appear once (i.e., retention of low weight particles). Based on this rule, the new resampled particles with indices $j = \ell, \dots, \ell + \Gamma_{x_k^i} - 1$ are

$$\check{x}_k^j = x_k^i \quad (5.3)$$

¹This multinomial selection of the number of replications of a particle has been discussed in section 2.5.4 of chapter 2.

and their weights are

$$\check{w}_k^j = \frac{w_k^i}{\Gamma_{x_k^i}} \quad (5.4)$$

where

$$\ell = \begin{cases} 1 & \text{if } i = 1 \\ \ell + \Gamma_{x_k^{i-1}} & \text{if } i > 1 \end{cases} \quad (5.5)$$

In (5.3), it can be observed that particles having high importance, i.e., particles whose $\lfloor Iw_k^i \rfloor \geq 1$, are replicated in accordance to their weights and particles having low importance are retained. In (5.4), it can be observed that high importance weights are reweighted in proportion to the number of replications and low importance weights are retained (essentially the process does not change the original particle weight). Overall, every particle whose weight is less than $2/I$ will appear only once with its weight unchanged, while every particle whose weight is greater than $2/I$ will appear $\lfloor Iw_k^i \rfloor$ times, with its weight equally divided amongst them. Since the original weight of a particle is retained in the duplicated copies of that particle in the resampled particle set, the deviation of the new weights $\{\check{w}_k^j\}_{j=1}^{\Gamma_{x_k^1} + \dots + \Gamma_{x_k^I}}$ from the actual weights $\{w_k^i\}_{i=1}^I$ is zero, that is to say, the low importance weights are also considered important.

The number of reweighted particles after this process is

$$\check{I} = \sum_{i=1}^I \Gamma_{x_k^i} \quad (5.6)$$

and usually $\check{I} > I$. Since only I reweighted particles are eventually required (and it is not necessary to treat *all* low weights to be important), the algorithm is terminated once I particles are created, and the others are discarded. The final weighted particle set is then the subset $\{x_k^i, a_k^i\}_{i=1}^I$ of the set $\{\check{x}_k^i, \check{w}_k^i\}_{i=1}^{\check{I}}$. The procedure is illustrated in Fig. 5.1.

It is now important to satisfy the condition $\sum_{i=1}^I a_k^i = 1$, i.e., to preserve the weights of the discarded particles. These discarded weights $\{\check{w}_k^i\}_{i=I+1}^{\check{I}}$ can be

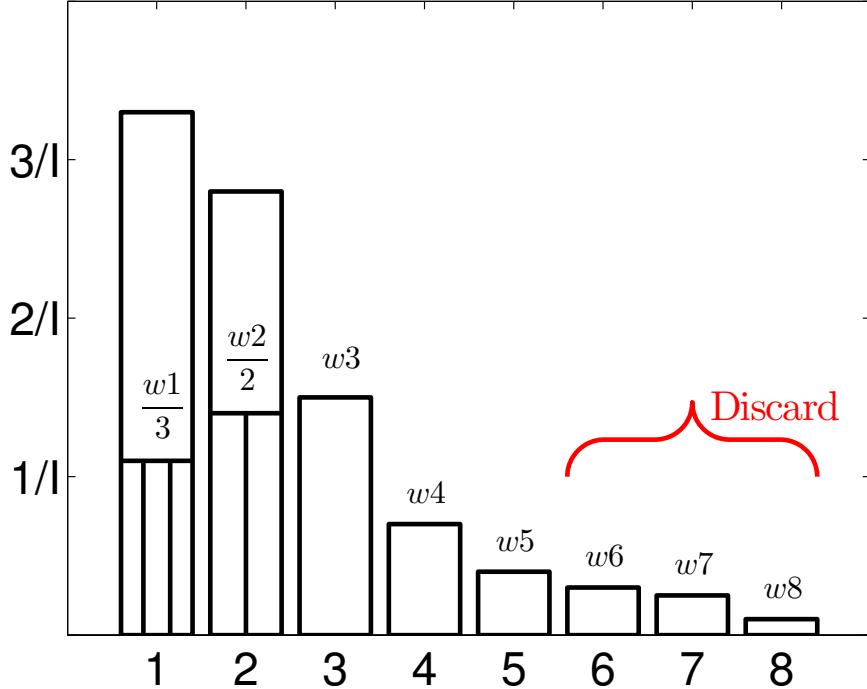


Figure 5.1: Illustration of soft resampling. The figure shows weights of 8 particles; the weight vector $w = [0.3529, 0.2995, 0.1604, 0.0749, 0.0428, 0.0321, 0.0267, 0.0107]$. Among the 8 particles, the first particle has a weight of $3.3/I$ and hence gets 3 copies each weighted at $\frac{w_1}{3} = \frac{1.1}{I}$. The second particle has a weight of $2.8/I$ and hence gets 2 copies each weighted at $\frac{w_2}{2} = \frac{1.4}{I}$. The other particles' weights are left unchanged since they are all less than $2/I$. The last 3 particles are discarded since only 8 particles are eventually required.

reallocated to the retained weights $\{a_k^i\}_{i=1}^I$ in one of the two following ways;

Renormalisation: The retained weights are renormalised as

$$b_k^i = \frac{a_k^i}{\sum_{i=1}^I a_k^i} \quad (5.7)$$

such that the total weight of the discarded particles is shared among all the remaining weights.

Redistribution: The total discarded weight is

$$a_{\text{spare}} = 1 - \sum_{i=1}^I a_k^i \quad (5.8)$$

This weight is redistributed (or shared) only among the lower weight particles as follows: find the particle index y such that

$$a_{\text{spare}} + a_k^I + a_k^{I-1} + \dots + a_k^{I-y} \leq a_k^{I-y-1} \quad (5.9)$$

Then the new weights are

$$b_k^i = \begin{cases} a_k^i & \text{for } i = 1, \dots, y-1 \\ \frac{a_{\text{spare}} + \sum_{s=y}^I a_k^s}{y} & \text{for } i = y, \dots, I \end{cases} \quad (5.10)$$

This method further reduces the weight variance because it allocates the discarded weight only to low importance weights and does not modify the high importance weights.

The soft resampling presented in this section was found to introduce a bias in (5.1). This motivates the work of the next section.

5.4 Soft systematic resampling

In this section, a modification to the previously presented soft resampler is proposed. The rationale behind this proposal is discussed in section 5.4.1 and the algorithm is then presented in section 5.4.2.

5.4.1 Motivation for soft systematic resampling

Retaining the information contained in the particles with low weights is crucial in the detection of very covert targets [55] and in the tracking of highly manoeuvring targets, especially when the number of particles is limited. Soft resampling [61] proposed in the previous section aids in this retention. In

soft resampling, the particles having weights greater than $2/I$ are replicated using (5.3) and their weights are revised using (5.4). Once the particle limit I is reached, the redundant particles and their weights are discarded. The information contained in these weights is retained by renormalisation or re-distribution. However the redundant particles — particles located near the tails of the cumulative distribution function (cdf) of the posterior density, i.e., the low weight particles that lie near the boundary of the particle set — are *discarded all the time*. This perpetual elimination causes a bias in (5.1). Discarding these particles could prove detrimental especially when the weights of the particles close to the target are very large, e.g., the MUSIC based MTT PF in chapter 3 or [256]. These large weight particles are replicated more times causing many lower weight particles that lie near the edges of the particle set to be eliminated. As a result, the PF approximation is truncated near the tails of the cdf thereby leading to an *inaccurate posterior pdf*. This problem is less evident in the stochastic resamplers because they resample all the particles with non-zero probability thus ensuring that not every low weight particle is lost perpetually.

The soft systematic resampling approach [278] presented in this section is an extension and correction to the soft resampling technique and presents a scheme to overcome the problem of discarding the low weight particles all the time. The main idea is to first use a modified variant of soft resampling without discarding any particles and then stochastically resample the lower weights using the systematic resampler [52, 136]. Hence, the technique is termed *soft systematic resampling*.

5.4.2 Soft systematic resampling algorithm

To overcome the problem of perpetually discarding low weight particles in soft resampling, this thesis proposes to *resample a few lower weight particles with non-zero probability*. This can be done by using a systematic resampler [52]. The systematic resampler is used to *stochastically* resample a few lower

weight particles so that not every low weight particle is eliminated all the time. The consequential soft systematic resampling [278] is presented here.

After the SIS step, the weights are normalised and sorted in descending order. For a particle x_k^i , if its weight $w_k^i > 2/I$, x_k^i is replicated

$$\Gamma_{x_k^i} = \max\{1, \lfloor \alpha I w_k^i \rfloor\} \quad (5.11)$$

times, where $0 < \alpha \leq 1$. Low values for α correspond to less replications of the larger weights. α can be used to regulate the spread of the large weights, particularly when the likelihood of the particle closest to the target is very large. Based on this principle the new resampled-reweighted particle set $\{\check{x}_k^i, \check{w}_k^i\}_{i=1}^{\check{I}}$ is obtained using (5.3) and (5.4), where \check{I} is specified in (5.6). Note that the combined weight of the resampled particles for a particular index i is the same before and after resampling, and is independent of α .

Then N_{low} lower weight particles are stochastically resampled. Here,

$$N_{\text{low}} = \min\left\{\check{I}, \lfloor \beta(\check{I} - I) \rfloor\right\} \quad (5.12)$$

The term $\beta \geq 1$ regulates the value of N_{low} such that a large β results in more low weight particles being stochastically resampled. The weighted particle set $\{\check{x}_k^i, \check{w}_k^i\}_{i=\check{I}-N_{\text{low}}+1}^{\check{I}}$ is then fed to the systematic resampler after normalising its weights. Since only I particles are eventually required, the number of particles the resampler generates is limited to

$$N_{\text{resamp}} = N_{\text{low}} - (\check{I} - I) \quad (5.13)$$

These N_{resamp} particles will then replace the particles $\{\check{x}_k^i\}_{i=\check{I}-N_{\text{low}}+1}^{\check{I}}$. This process ensures that not all the $\check{I} - I$ particles (these are the low weight particles that lie near the edge of the particle cloud) are eliminated all the time. The resampled particles are then reweighted as

$$\check{w}_k^i = \frac{\sum_{j=\check{I}-N_{\text{low}}+1}^{\check{I}} \check{w}_k^j}{N_{\text{resamp}}}, \quad i = \check{I} - N_{\text{low}} + 1, \dots, I \quad (5.14)$$

The remaining $\check{I} - I$ weighted particles are then discarded. The final weighted particle set is then the subset $\{\mathbf{x}_k^i, a_k^i\}_{i=1}^I$ of the set $\{\check{\mathbf{x}}_k^i, \check{w}_k^i\}_{i=1}^{\check{I}}$. The soft systematic resampling is illustrated in Fig. 5.2.

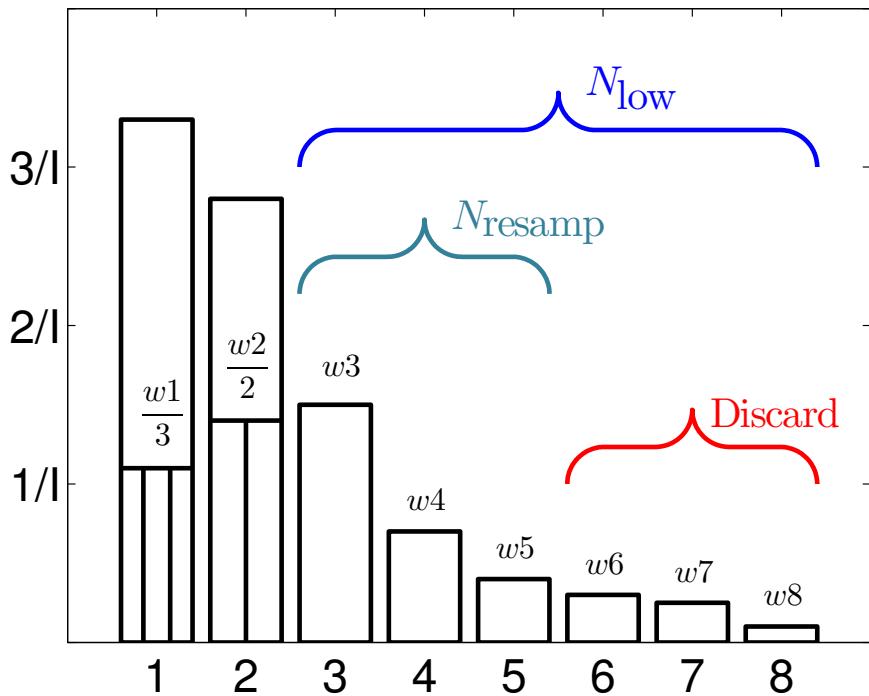


Figure 5.2: Illustration of soft systematic resampling with $\alpha = 1$ and $\beta = 2$, for the same weight vector given in the caption of Fig. 5.1. The original sample size is $I = 8$. The replication and reweighting process generates $\check{I} = 11$ particles. From (5.12), the number of low weights particles to be stochastically resampled is $N_{\text{low}} = 6$, i.e., particles with indices $i = \check{I} - N_{\text{low}} + 1, \dots, \check{I} = 6, \dots, 11$, are stochastically resampled to generate $N_{\text{resamp}} = 3$ particles. These N_{resamp} particles will replace particles with indices $i = \check{I} - N_{\text{low}} + 1, \dots, I = 6, \dots, 8$. The weights of particles with indices $i = \check{I} - N_{\text{low}} + 1, \dots, I = 6, \dots, 8$, are modified using (5.14). Finally, the last $\check{I} - I = 3$ weighted particles are discarded.

5.4.3 Analysis of properties of the soft resampler

Here, the unbiased property of soft (systematic) resampling is analysed using a simple example. As defined in (5.1), the soft resampler is said to be unbiased if the weight of a particle before resampling is completely preserved in each of its duplicated copies, i.e., if

$$w_k^i - E \left[\sum_{\forall j, 1 \leq j \leq I : x_k^{*j} = x_k^i} w_k^{*j} \right] = 0 \quad (5.15)$$

where $\{x_k^i, w_k^i\}_{i=1}^I$ and $\{x_k^{*j}, w_k^{*j}\}_{j=1}^I$ respectively is the weighted particle set before and after resampling. Fig. 5.3 shows the bias analysis for a set of 100 samples having uniformly random weights. These (sorted) weights are shown in Fig. 5.3(a). Figs. 5.3(b) and (d) show the error between the weight of a particle before resampling and the expected value of the total weight in each of the duplicated copies of that particle, i.e., the computation

$$w_k^i - E \left[\sum_{\forall j, 1 \leq j \leq I : x_k^{*j} = x_k^i} w_k^{*j} \right]$$

Fig. 5.3(c) shows the variance of the total weight contained in all the duplicated copies of a particle, i.e., the computation

$$\text{Var} \left[\sum_{\forall j, 1 \leq j \leq I : x_k^{*j} = x_k^i} w_k^{*j} \right]$$

It can be observed from the figure that the weight error and weight variance in the soft resamplers is much smaller when compared to that of the conventional stochastic resamplers. This is because the former preserve the complete weight of a particle in each of its duplicated copies. Also note the difference in the order of magnitude of the weight error and weight variance for the soft and stochastic resamplers. The low weight error in soft (systematic) resampling indicates that the actual weight of a particle is retained in the new particle set. This leads to improved information retention over time.

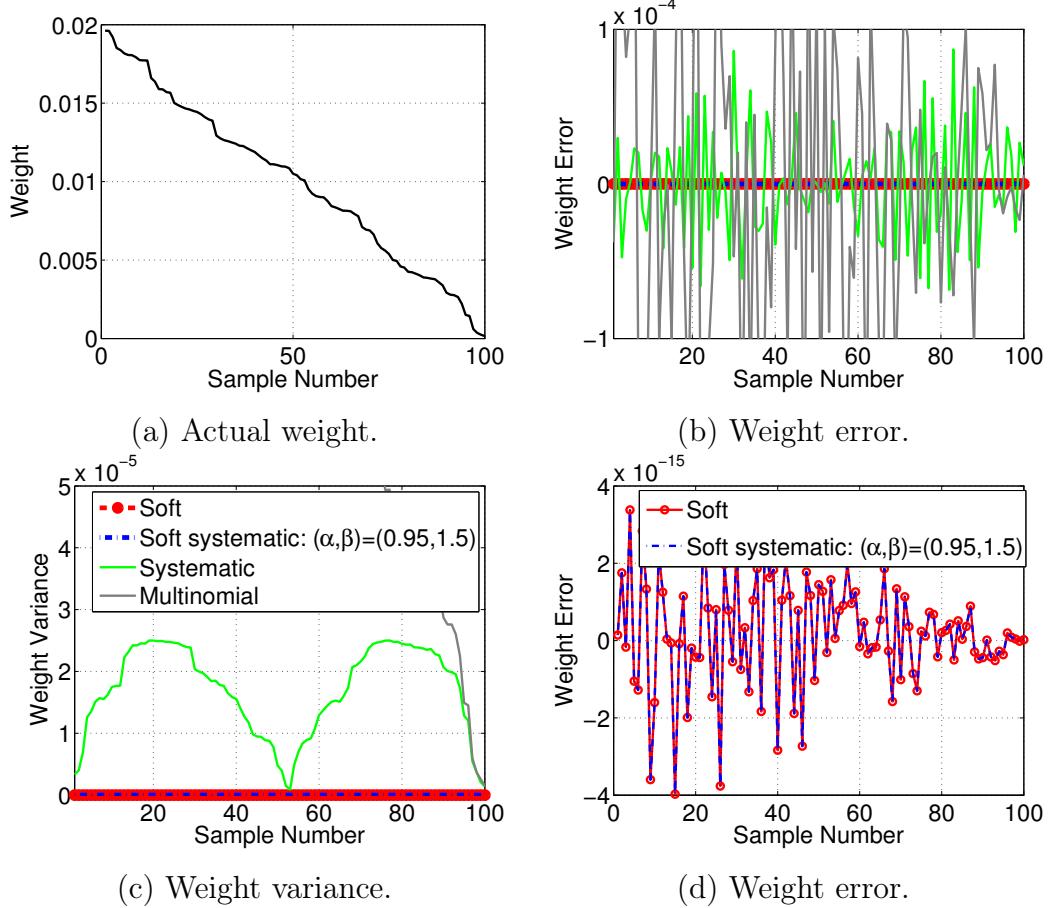


Figure 5.3: Soft resampler bias analysis. The number of particles (samples) is $I = 100$. The results are computed over 1000 realisations. The legend of (c) applies to (b) also. The soft and soft systematic resamplers are plotted separately in (d) for better visualisation.

Since the weights are uniform, the number of duplications of a particle is zero (in this example). Hence the soft resampler does not suffer from bias. The bias analysis when only a few particles have large weights is shown in Fig. 5.4. In this example, the (sorted) weights of 100 samples are shown in Fig. 5.4(a). It can be observed from Fig. 5.4(b) that the soft resampler suffers a bias, albeit having a very low weight error and variance. This is because the low weights that lie near the tail of the distribution are completely discarded. This bias is successfully overcome in the soft systematic resampler by virtue

of stochastically resampling a few low weight particles. It can be noted that as β increases, the performance of the soft systematic resampler approaches that of the systematic resampler.

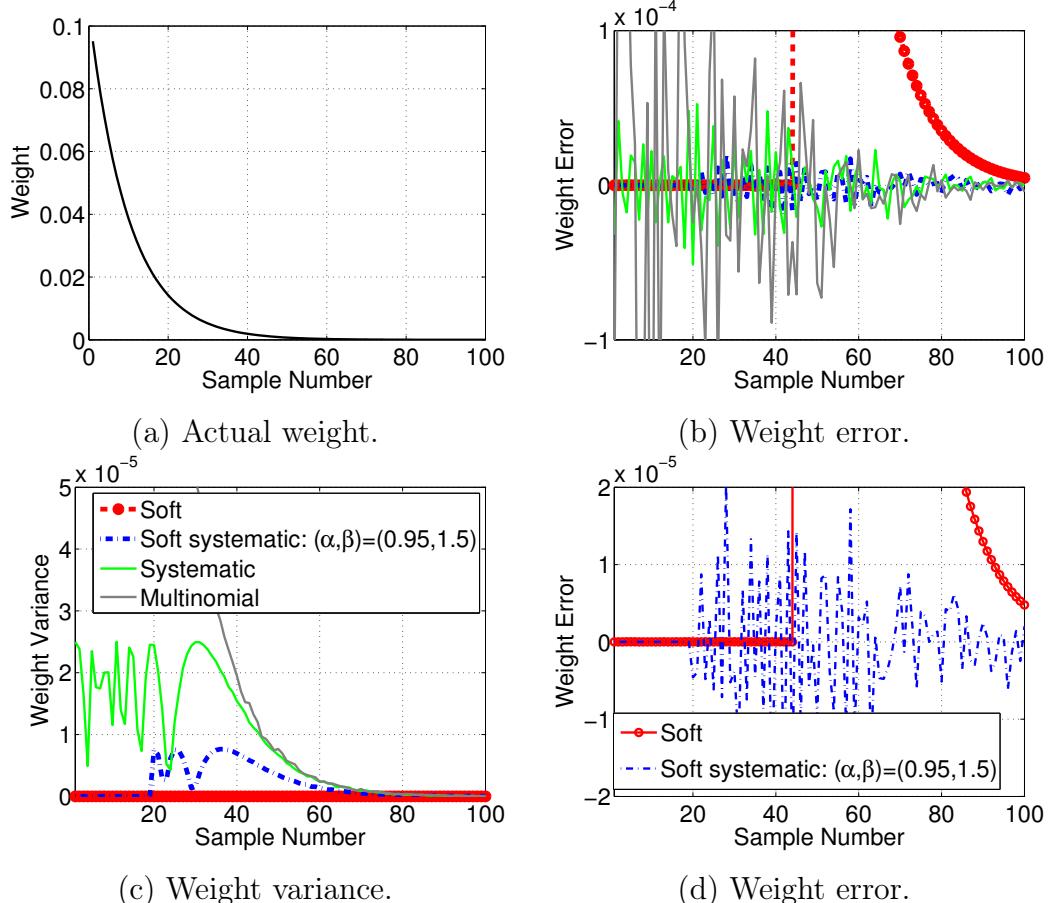


Figure 5.4: Soft resampler bias analysis. The number of particles (samples) is $I = 100$. The results are computed over 1000 realisations. The legend of (c) applies to (b) also. The soft and soft systematic resamplers are plotted separately in (d) for better visualisation.

5.5 The numerical measure of PF performance

Here, a quantitative measure to evaluate the accuracy of the PF in representing the posterior is introduced. The numerical test presented here can

be used in a generic PF framework.

It is known that for linear Gaussian models, the Kalman filter [20], described in section 2.4 of chapter 2, provides a optimal tracking estimate [32, 81, 90]. This thesis proposes to test the faithfulness of the PF in accordance to its KS statistic agreement [62] with the *theoretically optimal* Kalman filter. The KS test² has the advantage of providing a reliable measure of the accuracy [279] of the estimate of the posterior. However, the test has been under-utilised in the PF literature and has never been used to measure resampler performance. The classical one-sample KS test evaluates the misfit between the cdfs of two uni-dimensional ordered datasets [273] by computing the largest absolute difference. If the empirical distribution X_n for a set of I independent and identical distributed (i.i.d) uni-dimensional particles $\{x^i\}_{i=1}^I$ is defined as

$$X_n(y) = \frac{\text{number of elements in the sample } \leq y}{I} = \frac{|\{x^i : x^i \leq y\}|}{I} \quad (5.16)$$

Then the KS statistic for a given cdf X is

$$\kappa_{\text{KS}} = \sup_y |X_n(y) - X(y)| \quad (5.17)$$

This procedure however, cannot be applied to multi-dimensional distributions because there is no unique way to order the data. KS testing for multi-dimensional Gaussian models [274] can be conducted using the two-sample KS statistic [275, 276] by comparing the cdfs from two multi-dimensional datasets which are ordered in some fashion. One method is to compare the cdfs of the two samples with all possible combinations of sorting, and take the largest of the set of resulting KS statistics. Multi-dimensional KS testing has been used previously for other PF problems [277, 280]. A few other performance evaluation schemes that may also be adaptable to PF performance metrics can be found in [281].

In this thesis, for KS testing in linear Gaussian and multi-dimensional

²The KS test was first used in this dissertation in chapter 4.

models, a modified KS testing approach is proposed. The rationale for this proposal is to *exploit the availability of the mean and covariance estimate provided by the theoretically optimal Kalman filter*. The procedure will now be presented.

5.5.1 The modified KS testing approach

Mathematically, the proposed KS statistic testing is conducted as follows: At time k , let the Kalman mean estimate be μ_k and the covariance estimate be P_k . Let G_k be the Cholesky decomposition of P_k . Consider that the D -dimensional resampled particle set $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^I$ is available and that the particle states along the d th dimension are denoted by $\{\mathbf{x}_{d,k}^i\}_{i=1}^I$. Firstly, each dimension of the particle state as represented by the particles are de-meaned and de-correlated using the theoretically optimal Kalman mean and covariance. This is done according to

$$y_{d,k}^i = G_k^{-1} (\mathbf{x}_{d,k}^i - \mu_k) \quad (5.18)$$

for $i = 1, \dots, I$ and $d = 1, \dots, D$. The new particles $\{y_{d,k}^i\}_{i=1}^I$ are then sorted as

$$\{\Psi_{d,k}^i, \theta_{d,k}^i\}_{i=1}^I = \text{sort}(\{y_{d,k}^i\}_{i=1}^I); d = 1, \dots, D \quad (5.19)$$

where $\{\Psi_{d,k}^i\}_{i=1}^I$ contains the sorted particle states along the d th dimension and $\{\theta_{d,k}^i\}_{i=1}^I$ contains the indices of the particle states (in the d th dimension) in the sorted order. Then for $i = 1, \dots, I$ and $d = 1, \dots, D$, the cdf of the particle states is evaluated according to

$$\hat{c}_k(d, i) = \sum_{j=1}^i w_k^{\theta_{d,k}^j} \quad (5.20)$$

The emperical cdf is the error function related to the integral of the standard normal distribution and is measured according to

$$\breve{c}_k(d, i) = \Phi(\Psi_{d,k}^i) = \frac{1 + \text{erf}(\Psi_{d,k}^i / \sqrt{2})}{2} \quad (5.21)$$

where the error function is described as

$$\text{erf}(s) = \frac{2}{\sqrt{\pi}} \int_0^s \exp(-t^2) dt \quad (5.22)$$

Finally, the KS statistic is measured by conducting the one-sample KS test along each dimension separately and then taking the maximum KS deviation value amongst all the dimensions. That is,

$$\kappa_{\text{KS}} = \sup_d \sup_i |\hat{c}_k(d, i) - \check{c}_k(d, i)| \quad (5.23)$$

In words, the procedure is as follows: After resampling, the particle states are de-meanned and de-correlated using the theoretically optimal Kalman mean and covariance. Then the particle states along each dimension are sorted and the cdf (along that dimension) is constructed by evaluating the cumulative sum of the weights taken in the sorted order. Finally, the one-sample KS test is conducted along each dimension separately by comparing the cdf of the particle states and the error function, and maximum KS deviation value amongst all the dimensions is taken.

If the PF representation is identical to that of the Kalman filter distribution, then (5.18) will result in a set of uncorrelated particles which are normally distributed with zero mean and unit variance. The cdf of the states of these particles (after sorting) will be close to that of the error function, i.e., the KS deviation will be small. Conversely, if the PF representation is not identical to that of the Kalman filter distribution, then the cdf of the particle states will differ from that of the error function, i.e., the KS deviation will be large. Overall, the proposed method tests the *representation misfit* of the PF against a reference distribution — the theoretically optimal Kalman filter. The procedure is now illustrated using a simple example.

Illustration of the proposed KS test:

Consider a two dimensional state space. Let the Kalman filter mean be $\mu = [6, 8]^T$ and its covariance be

$$P = \begin{bmatrix} 0.85 & 0.2 \\ 0.2 & 0.7 \end{bmatrix}$$

The first scenario is shown in Fig. 5.5. Here, 1000 particles are drawn from a 2D normal distribution with mean $\mu_{\text{PF}} = \mu$ and covariance $P_{\text{PF}} = P$. These particles are plotted (as red dots) over the contour describing the Kalman filter distribution. The de-mean and de-correlation of these particles will result in a new set of particles (plotted as black dots). The contour of a standard normal distribution is also shown for reference. Since the PF representation is identical to that of the Kalman filter distribution, the deviation of the cdf of particles from the error function of $\mathcal{N}(0, 1)$ is small.

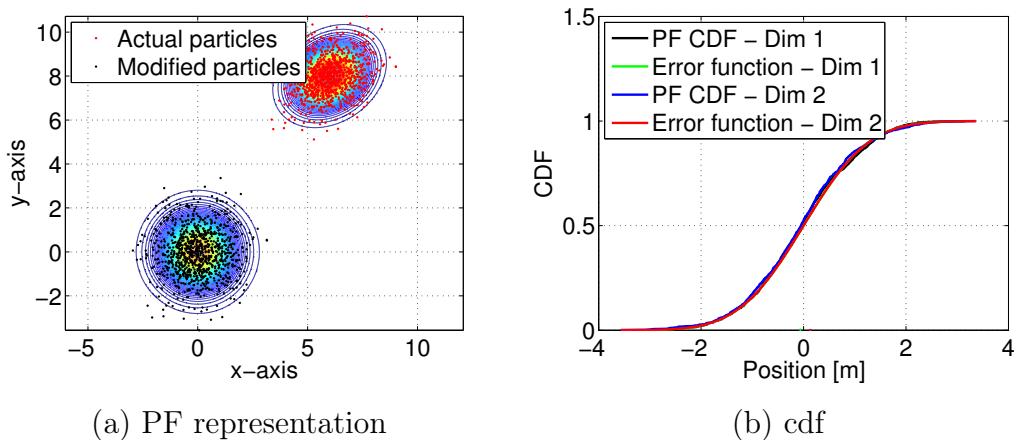


Figure 5.5: KS test illustration when the PF representation is close to the optimal Kalman filter distribution.

The second scenario is shown in Fig. 5.6. Here, 100 particles are drawn from a 2D normal distribution with mean $\mu_{\text{PF}} = \mu$ and covariance $P_{\text{PF}} = P$. Since the PF representation is identical to that of the Kalman filter distribution, the deviation of the cdf of particles from the error function of $\mathcal{N}(0, 1)$ is small. However, the figure shows that the accuracy of approximation depends on the number of particles I , and insufficient I leads to poor representation of the distribution $\mathcal{N}(0, 1)$. Hence the deviation of the cdf of particle states

from the error function of $\mathcal{N}(0, 1)$ is larger when compared to the case of $I = 1000$.

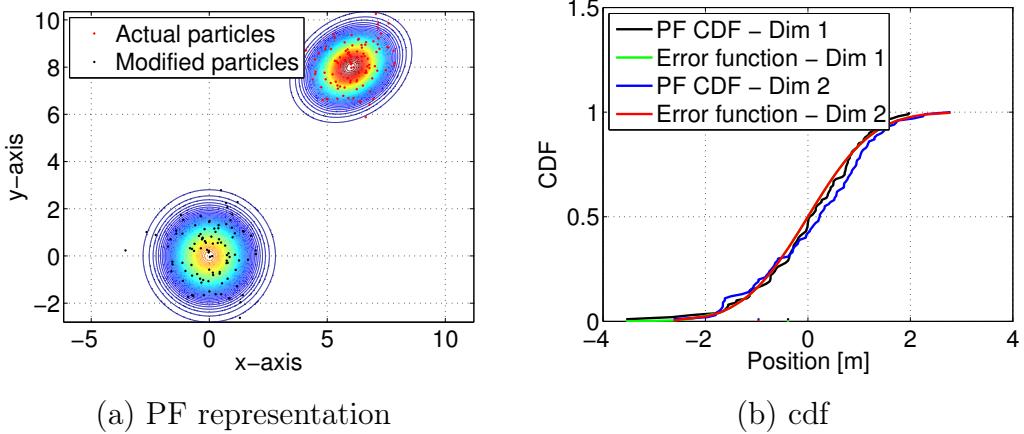


Figure 5.6: KS test illustration when the PF representation is close to the optimal Kalman filter distribution, but the number of particles I is low.

The third scenario is shown in Fig. 5.7. Here, 1000 particles are drawn from a 2D normal distribution with mean and covariance different from μ and P respectively. Since the PF representation is not identical to that of the Kalman filter distribution, the deviation of the cdf of particles from the error function of $\mathcal{N}(0, 1)$ is large.

5.6 Evaluation

The validity and efficacy of the proposed soft and soft systematic resampling methods is demonstrated here. Although the primary interest of this thesis is on phased sensor arrays, other sensor models are used here in the context of a) being able to test the KS performance of the resamplers, and b) availability of real image data. The proposed methods are first evaluated using image observations in section 5.6.1. Secondly, the methods are tested by comparing with the optimal Kalman filter in section 5.6.2 using a 2D linear Gaussian model.

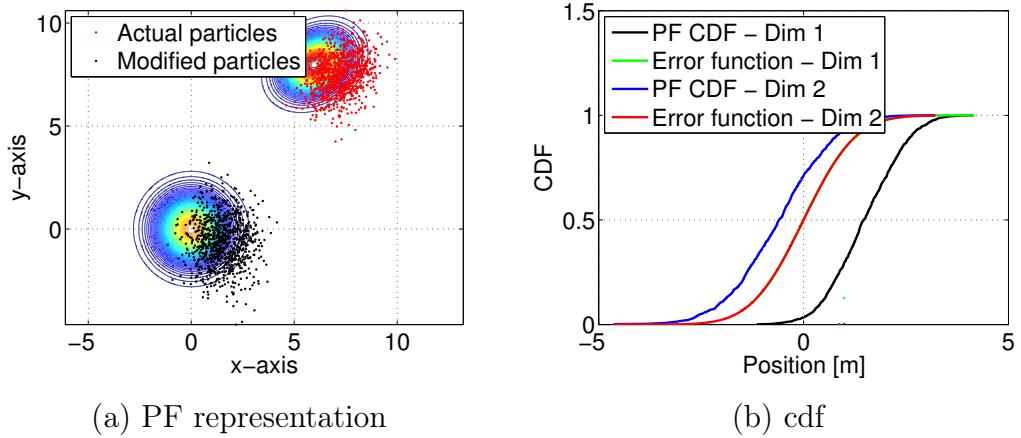


Figure 5.7: KS test illustration when the PF representation is not identical to the optimal Kalman filter distribution.

5.6.1 The proposal analysed using a TBD example

Here, the efficacy of the proposed methods is tested using a TBD application with single target tracking using Gaussian image models. The model specifications can be found in Ch. 11 of [56] and also in [55]. The state of the target is defined as $\mathbf{x}_k = (x, v_x, y, v_y, B)^T$ where x, y are the positions and v_x, v_y are the velocities along the x, y -axes. B is the intensity contribution of the target to the image. $(\cdot)^T$ denotes matrix transpose. The time evolution of \mathbf{x}_k is described as

$$x_k = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & T & 0 & 0 & 1 \end{bmatrix} x_{k-1} + q_{k-1} \quad (5.24)$$

where the additive process noise $q_{k-1} \sim \mathcal{N}(0, Q)$ with

$$Q = \begin{bmatrix} \frac{0.001}{3} T^3 & \frac{0.001}{2} T^2 & 0 & 0 & 0 \\ \frac{0.001}{2} T^2 & 0.001 T^2 & 0 & 0 & 0 \\ 0 & 0 & \frac{0.001}{3} T^3 & \frac{0.001}{2} T^2 & 0 \\ 0 & 0 & \frac{0.001}{2} T^2 & 0.001 T^2 & 0 \\ 0 & 0 & 0 & 0 & 0.01 T \end{bmatrix} \quad (5.25)$$

The duration between two observations is $T = 1\text{s}$. The observations are 20×20 staring camera snapshots. The snapshot intensity at the (n, m) th pixel at time k is

$$z_k^{(n,m)} = \begin{cases} \mathcal{N}(\gamma_k^{(n,m)}(\mathbf{x}_k), \sigma^2) & \text{if the target is present} \\ \mathcal{N}(0, \sigma^2) & \text{otherwise} \end{cases} \quad (5.26)$$

where the intensity contribution $\gamma_k^{(n,m)}(\mathbf{x}_k)$ by a target with state $\mathbf{x}_k = (x, v_x, y, v_y, B)^T$ to the (n, m) th pixel is

$$\gamma_k^{(n,m)}(\mathbf{x}_k) = \frac{\Delta_n \Delta_m B}{2\pi \Sigma^2} \exp\left(-\frac{(n\Delta_n - x)^2 + (m\Delta_m - y)^2}{2\Sigma^2}\right) \quad (5.27)$$

The blurring parameter is $\Sigma = 0.7$. $(\Delta_n, \Delta_m) = (1, 1)$ is the center of the (n, m) th pixel. σ is the camera background noise standard deviation at each pixel.

The test runs for 30 frames (time steps) as follows: (i) the target is absent in the first 7 frames — the softness of the particles during this period is tested, (ii) the target appears in the 8th frame — the time to lock onto the target starting from the 8th frame is tested, and (iii) the target disappears from the 25th frame — the time to release target lock is tested. All the results are averaged over 50 iterations.

Softness

Firstly, the *softness* of the proposed methods is tested during the first 7 frames without the presence of a target. Fig. 5.8 shows the softness (a.k.a

looseness or variance) of 128 particles measured according to

$$C_{\text{loose}} = \frac{\sum ||\mathbf{x}_k^i - \sum_{i=1}^I \frac{\mathbf{x}_k^i}{I}||^2}{I} \quad (5.28)$$

for varying sensor noise standard deviation σ at each pixel in the snapshot. All the resamplers are tested on a standard PF (i.e., not on the TBD PF framework). It can be observed that by the 7th frame, the proposed soft and soft systematic resamplers leave a more loosely-fitted cluster than that of the conventional resamplers. The renormalised and redistributed variants of soft resampling exhibit almost similar performance. It can be seen that the soft systematic resampler with low α regulates the spread of the large weights, i.e.,

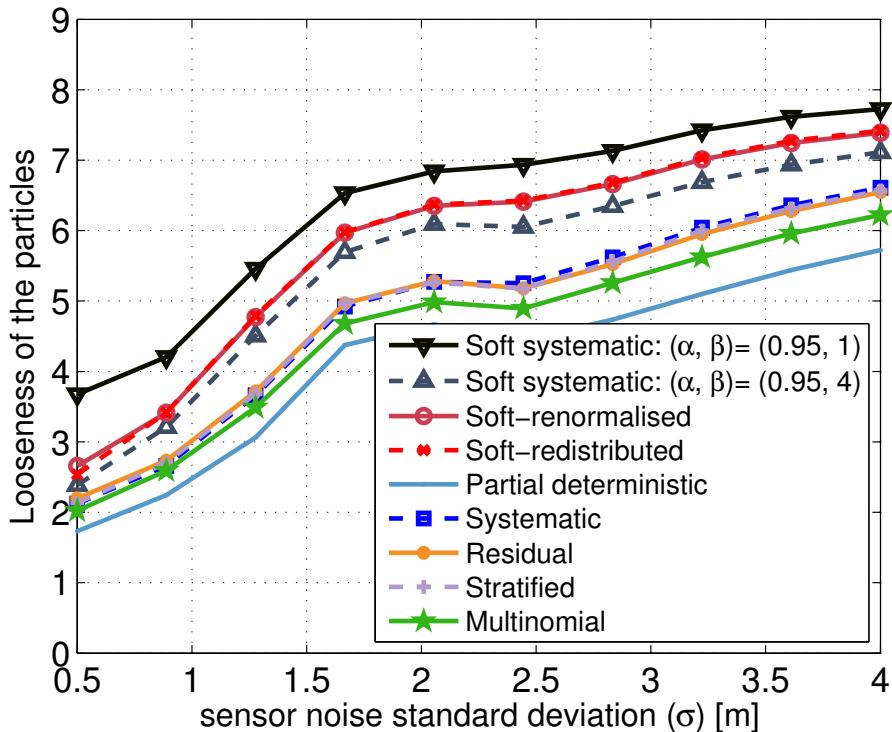


Figure 5.8: Looseness of 128 particles versus pixel noise standard deviation. The particles are propagated in a standard PF operating with the various resampling schemes shown in the figure. For partial deterministic resampling [120], the thresholds to find the dominant and negligible particles are $1/I$ and $1/I^2$ respectively.

low α generates fewer inter-particle replacements and this results in a more looser cluster. It can also be observed that the particle cloud gets tighter with increasing β and will approach the performance of the systematic resampler. Overall, the proposed resamplers leave a *soft* particle cluster and this feature aids in a faster lock (when compared to that of standard non-TBD PFs) when the target appears.

Time to lock

Secondly, the locking ability of the various filters is tested wherein the target appears in the 8th frame and starts manoeuvring. Fig. 5.9 shows the number of frames taken, beginning from the 8th frame, to lock the target. The proposed resamplers are tested on a standard PF. The conventional resamplers do not effectively aid in successive accumulation of the information sufficient for the detection of covert targets. This problem is overcome using the TBD PF. Hence the conventional resamplers are tested on both the standard PF and TBD PF frameworks. It can be observed in Fig. 5.9 that since the soft resampler leaves a loose particle cluster at the 7th frame, there are enough particles in the vicinity of target appearance to detect and quickly lock onto the target. The figure also indicates that the speed of lock in the soft systematic resampler decreases with increase in β (due to high stochasticity in particle replacements) and decrease in α (due to fewer particle replacements). It can also be observed that the traditional “hard” resamplers operating on the standard PF take more frames to lock the target than that of the TBD PF counterpart (because in the TBD PF, target detection is aided by the transition probability assigned to the particles). The stratified and residual resamplers exhibit almost similar performance to that of the systematic resampler and are hence not shown, both here in Fig. 5.9 and also in Fig. 5.10.

Time to release target lock

Thirdly, the time taken to release the target lock is tested: the target appeared at the 8th frame, manoeuvred until the 24th frame and then disap-

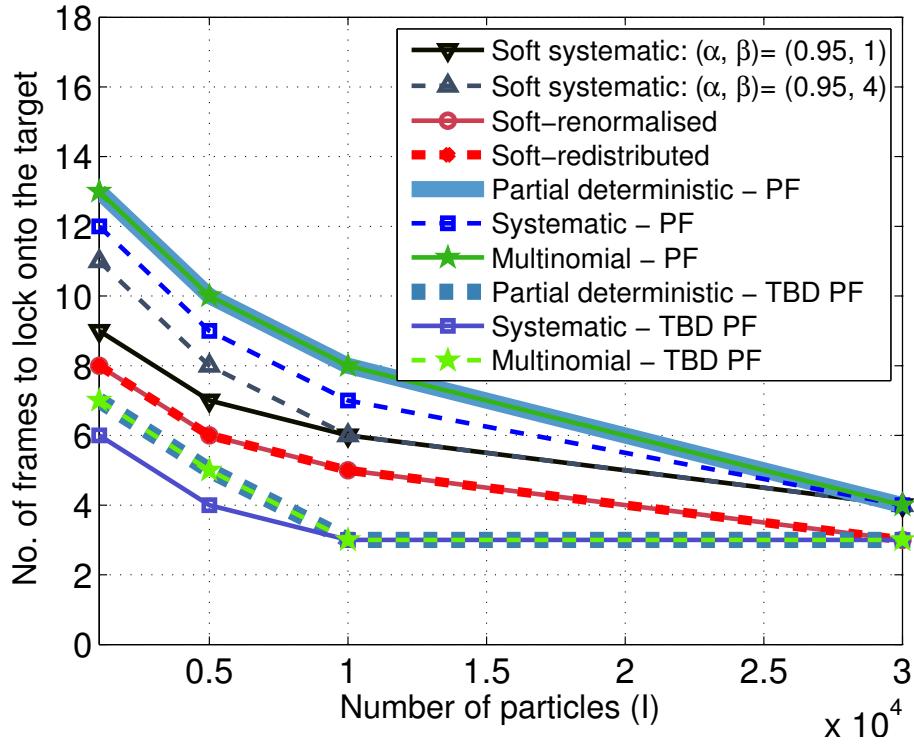


Figure 5.9: Number of frames taken from the appearance of a target in the 8th frame to lock onto the target, versus the number of particles. The target is considered to be locked; a) in the standard PF when the root mean square error decreases below 0.15m (In standard PFs, there is no direct measure to detect the presence of a target. Hence for comparison, we conveniently chose the position error), and b) in the TBD PF [56] when the target existence probability is $\frac{\sum_{i=1}^I E_k^i}{I} \geq [6I/10]$, i.e., when at least 60% of particles come to life. For TBD PF; a) the target birth probability is $p_b = 0.05$, b) the target death probability is $p_d = 0.05$, c) the positions of the new-born particles is $x/y \sim \mathcal{U}[0, 20]$, d) the velocities of the new-born particles is $v_x/v_y \sim \mathcal{U}[-1, 1]$, and e) the intensity of the new-born particles is $B \sim \mathcal{U}[10, 50]$. For partial deterministic resampling, the thresholds are $1/I$ and $1/I^2$. The noise standard deviation in the filters is $\sigma = 3$.

pears in the 25th frame. Fig. 5.10 shows the number of frames taken by the various filters to release the target lock. In the conventional resamplers operating on standard PF, once the target disappears, the divergence of the particle cloud is no longer weight dependent because all particles are consid-

ered to be equally probable in representing the posterior. The divergence of the particles then depends on the process noise sequence q_{k-1} , i.e., for highly observable targets, the matrix norm $\|Q\| = \sum_{i,j} Q_{i,j}^2$ is smaller and hence the particle cloud divergence is slower, and vice versa. That is to say, the release of the target in a standard PF operating on conventional resamplers is not dependent on the resampler. Hence for the conventional resamplers, only their TBD PF implementations are shown in the figure. Fig. 5.10 indicates that the TBD PF (using any resampler) releases the lock immediately once the target disappears. Releasing the target lock as soon as the target dis-

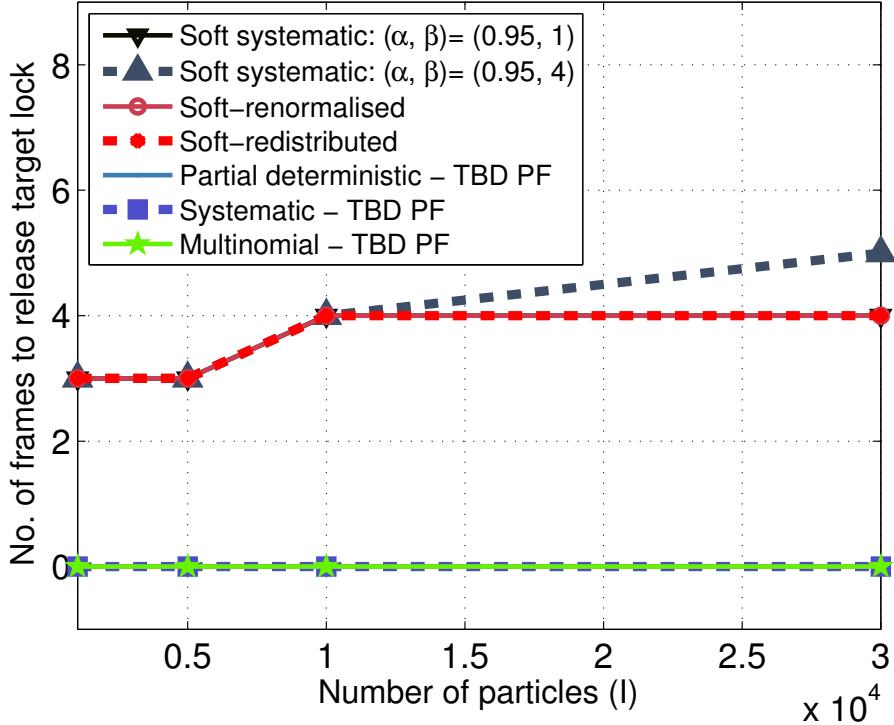


Figure 5.10: Number of frames taken from the disappearance of a target to release the target lock. Target is considered to be released; a) in the proposed method when $C_{\text{loose}} > -\log(0.3)$, and b) in the TBD PF when the target existence probability is $\frac{\sum_{i=1}^I E_k^i}{I} < \lfloor 6I/10 \rfloor$ [56]. The TBD PF system and partial deterministic resampling threshold specifications described in the caption of Fig. 5.9 apply to this figure also.

pears could be problematic in cases where the target momentarily shuts off, or is obstructed by non-target objects like buildings, etc. The proposed soft resamplers preserve the lower weights, and hence the standard PF operating on these resamplers waits for some time (anticipating target reappearance) before releasing the target lock. Note that these results are very sensitive to the parameters used for each filter type, and so these results of TBD and non-TBD are not directly comparable.

Accuracy

Fourthly, the accuracy of the proposed soft resampling methods is tested for the image observation models. Fig. 5.11 shows the root mean square error

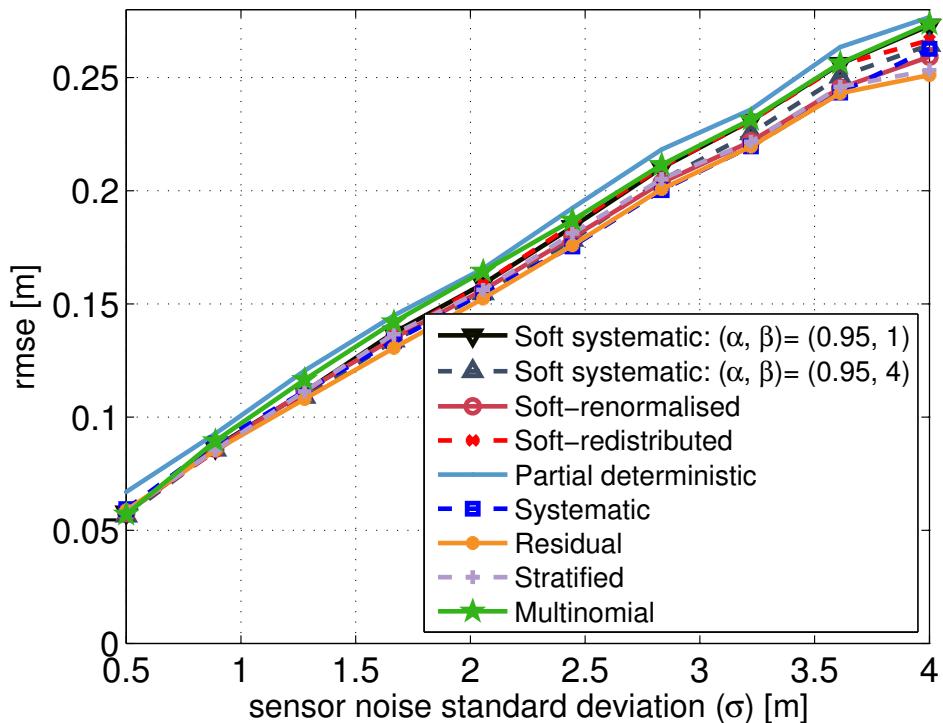


Figure 5.11: rmse versus the noise standard deviation σ . The number of particles is $I = 128$. The target is present all the time. The partial deterministic resampling threshold specifications described in the caption of Fig. 5.8 apply to this figure also.

(rmse) in position estimates for varying sensor noise standard deviation σ for 128 particles. Fig. 5.12 shows the rmse in position estimates for varying threshold levels for the effective sample size I_{eff} for 128 particles — the particles are resampled only when I_{eff} falls below the threshold. All the resamplers in Fig. 5.11 and Fig. 5.12 are operated on the standard PF and the target is present all the time for 20 time steps. It can be observed that the proposed techniques are comparable to state-of-the-art resamplers.

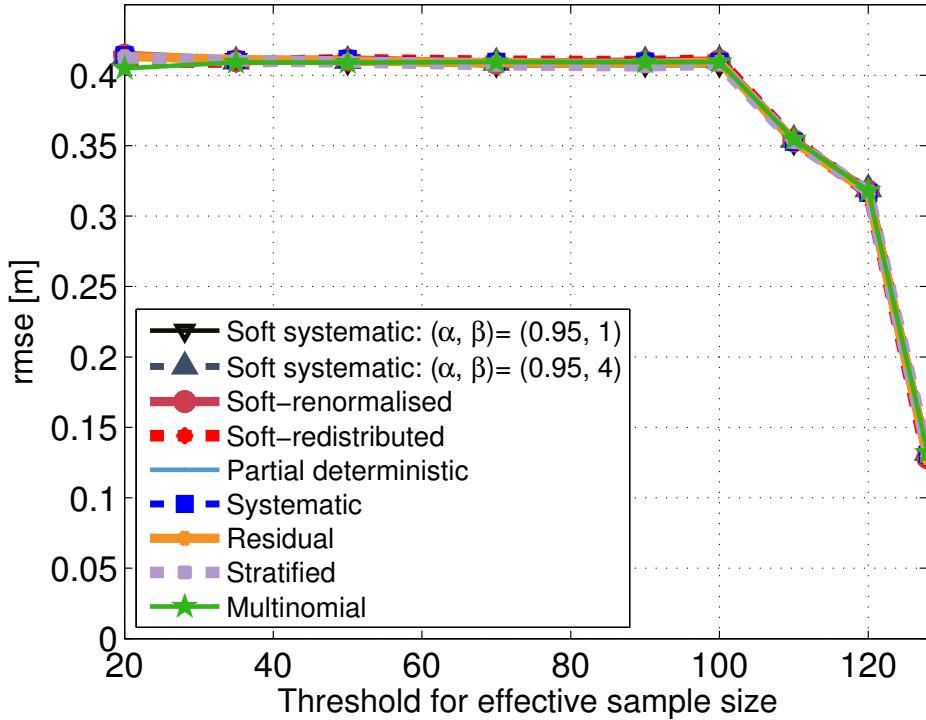


Figure 5.12: rmse versus the threshold for effective sample size I_{eff} . The number of particles is $I = 128$. The target is present all the time. For partial deterministic resampling, its specifications described in the caption of Fig. 5.8 apply here also.

5.6.2 The proposal compared with a Kalman filter

Here, the ability of the proposed soft and soft systematic resamplers to accurately represent the posterior and to retain more information is demonstrated. For this testing, a 2D linear Gaussian model is used so that comparison can be made with the Kalman filter [20] which is known to give the optimal posterior. The target state is characterised by the position and velocity of the target as $\mathbf{x}_k = (x, v_x)^T$. The process model is

$$\mathbf{x}_k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + q_{k-1} \quad (5.29)$$

with $T = 1\text{s}$ and the process noise $q_{k-1} \sim \mathcal{N}(0, Q)$ where

$$Q = \begin{bmatrix} 0.95 & 0.2 \\ 0.2 & 0.75 \end{bmatrix}$$

The observation model

$$z_k = \mathbf{x}_k + r_k \quad (5.30)$$

generates a clean though noisy point target measurement. The term “clean measurement” implies there are no false alarms, clutter detections, missed observations and/or out-of-sequence (OOS) observations³. The sensor noise is $r_k \sim \mathcal{N}(0, R)$ with $R = \sigma^2 \mathbf{I}$ having the variance $\sigma^2 = 0.5$ along each dimension.

Accuracy of posterior representation

Fig. 5.13 shows the KS statistic of various filters for varying numbers of particles. It can be observed that the PF operating with the systematic resampler exhibits the best consensus with the theoretically optimal posterior representation obtained using the Kalman filter. It can also be observed that

³A OOS observation is a target detection at time k but received at a later time sample $k + \tau$.

the performance of the soft resampler is slightly lower than that of the systematic resampler because of continually discarding the low weight particles. The figure also indicates that with increasing β the soft systematic resampler exhibits better agreement with the theoretically optimal posterior distribution. The soft redistribution exhibits similar performance to that of the soft renormalisation and hence is not shown.

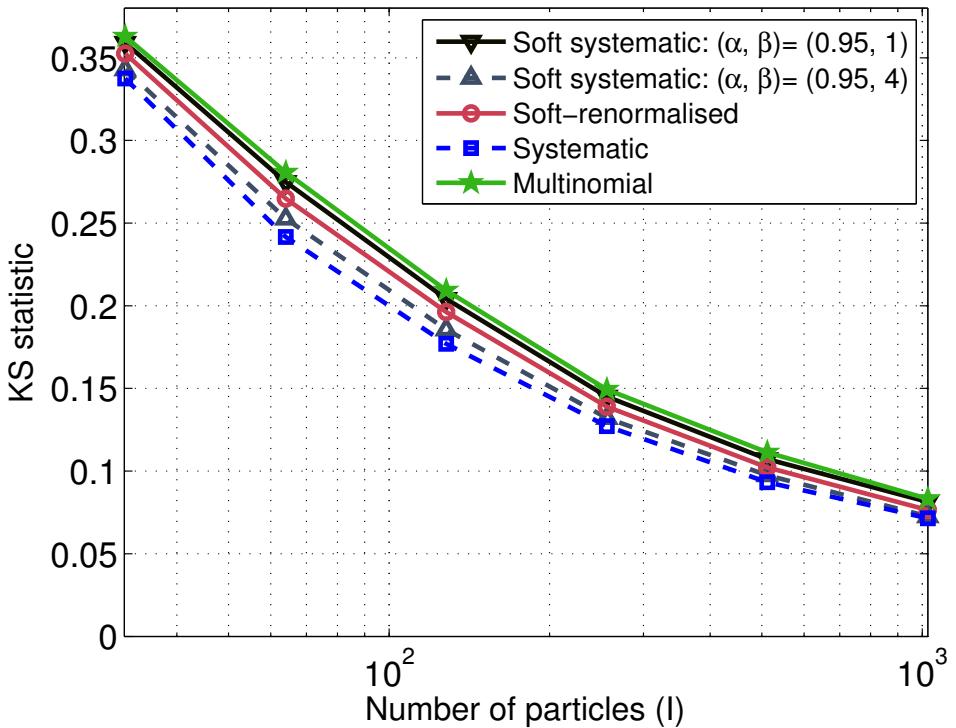


Figure 5.13: KS statistic versus the number of the particles I . The results are averaged over 50 instances, each having 50 time steps.

Fig. 5.14 shows the cdfs of the standard PFs (in the position dimension) operating on soft, soft systematic and systematic resamplers. The figure corresponds to a single time sample when the target is making a sharp manoeuvre. The proposed resamplers are compared with the systematic resampler because the latter has been found to exhibit the best performance

in terms of accurately representing the posterior in Fig. 5.13. In Fig. 5.14, the cdf in blue is obtained from (5.20) and the cdf in red is obtained from (5.21). It can be observed from the figure that while the soft resampler in Fig. 5.14(d) does not properly treat the tails of the cdf because the method discards the low weight particles all the time, the soft systematic resampler in Fig. 5.14(b) overcomes this problem by stochastically resampling the particles located near the tails of the cdf. For increasing β , the performance of

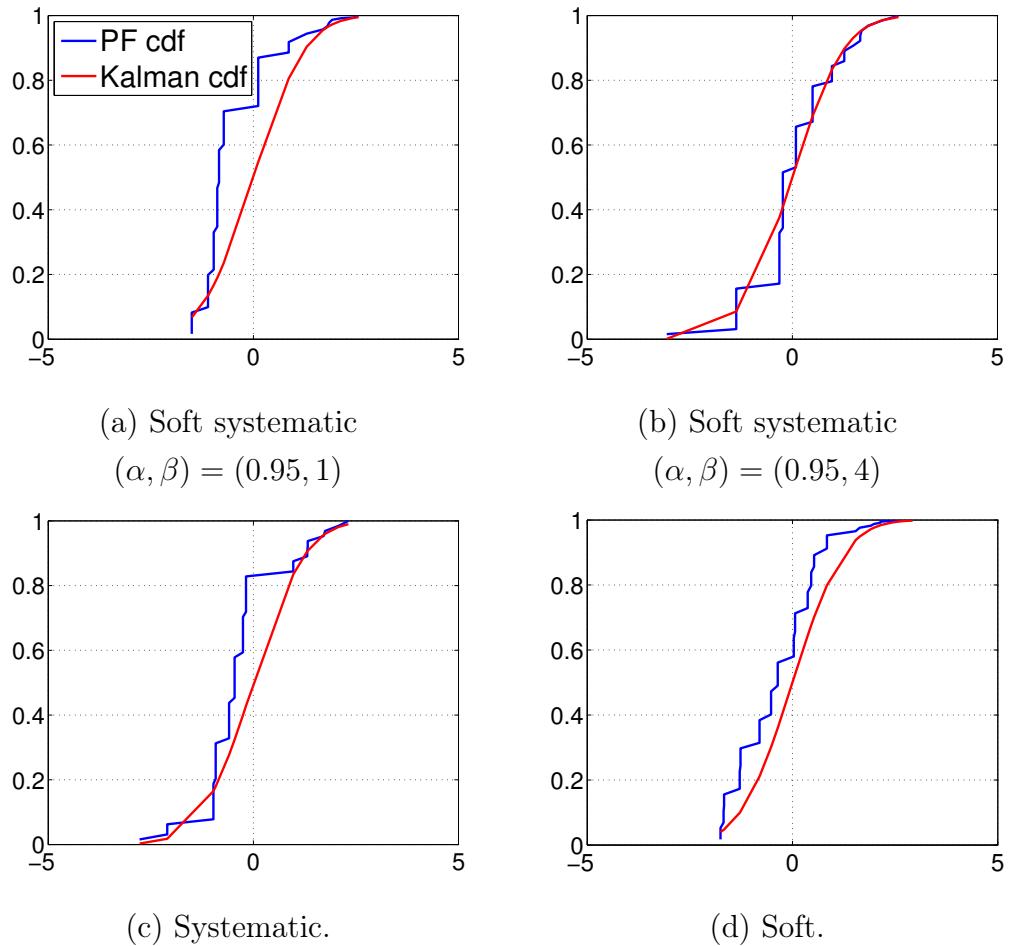


Figure 5.14: cdfs for PFs operated using soft, soft systematic and systematic resamplers. These cdfs correspond to the position of a target. The legend of Fig. 5.14(a) applies to Fig. 5.14(b,c,d) also.

the proposed soft systematic resampler will approach that of the systematic resampler. Fig. 5.14(c) indicates that the systematic resampler based PF approximation exhibits agreeable representation of the posterior described by the theoretically optimal Kalman filter. The soft resampling (with renormalisation) can be considered as soft systematic resampling with $(\alpha, \beta) = (1, 0)$. Therefore as observed in Fig. 5.14(a), decreasing the number of replications by reducing α will cause inaccuracies in posterior representation but the feature can be used to address other PF problems, e.g., to increase the diversity of the particles when the weights of the particles near a target are very large [256].

Information retention

The proposed soft and soft systematic resamplers are analysed in terms of their ability to retain more information over successive time steps and thereby aid in a faster lock. For this test, the 2D linear Gaussian model described in section 5.6.2 is again used. Here, the particles are initialised with unit covariance and a mean that is 8m away from the actual target. This arrangement results in a particle set that is *just* near the actual target by the time filtering starts. This arrangement requires retention of the weight information over time for a faster lock. Fig. 5.15 illustrates how fast the target is detected by showing the drop in the rmse of the PF estimates for 10 time samples. It can be inferred that by appropriate reweighting and retaining more information contained in the lower weights, the soft and soft systematic resamplers lock onto the targets faster than the systematic resampler that regards all the particles as equally weighted. The target in this test is allowed to make sharp manoeuvres, hence it is highly probable that the target could move towards a low weight particle. By ensuring that not all low weight particles are lost all the time, the soft systematic resampler aids in faster lock than its soft counterpart. The soft redistributed variant exhibits almost similar performance to that of soft renormalised variant and hence is not shown.

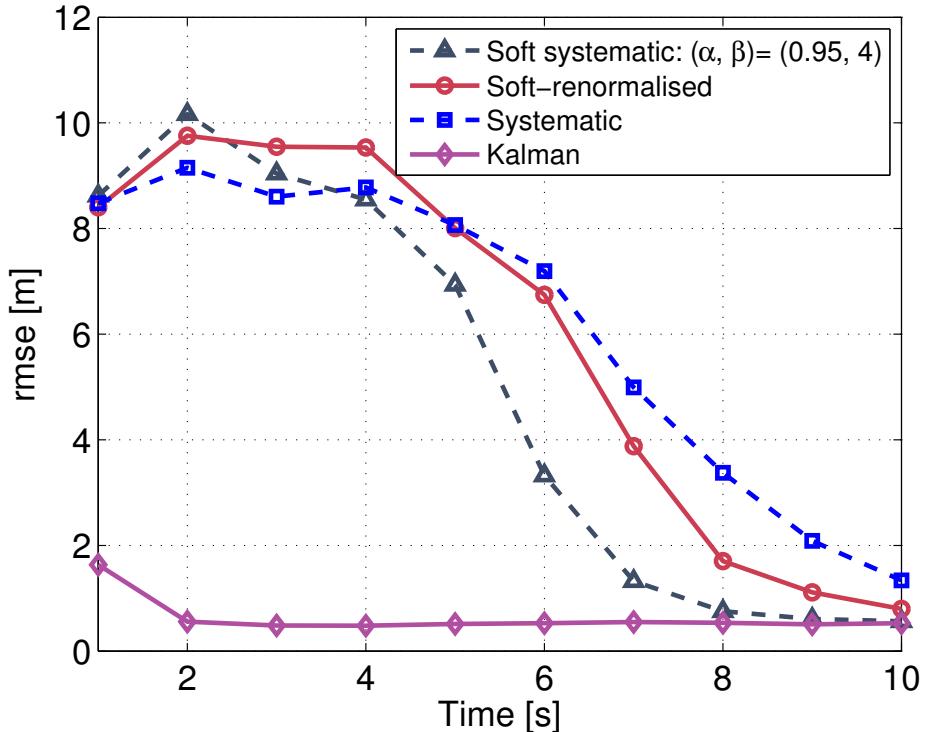


Figure 5.15: rmse versus time samples. The number of particles $I = 1024$. The results are averaged over 100 instances, each having 10 time steps.

Accuracy

The accuracy of the various resamplers is tested for the same 2D linear Gaussian model. Fig. 5.16 shows the rmse in position for varying sensor noise variance σ^2 for 1024 particles. Again, the soft redistributed variant exhibits almost similar performance to that of the soft renormalised variant and hence not shown. It can be observed that the proposed techniques are comparable to state-of-the-art resamplers.

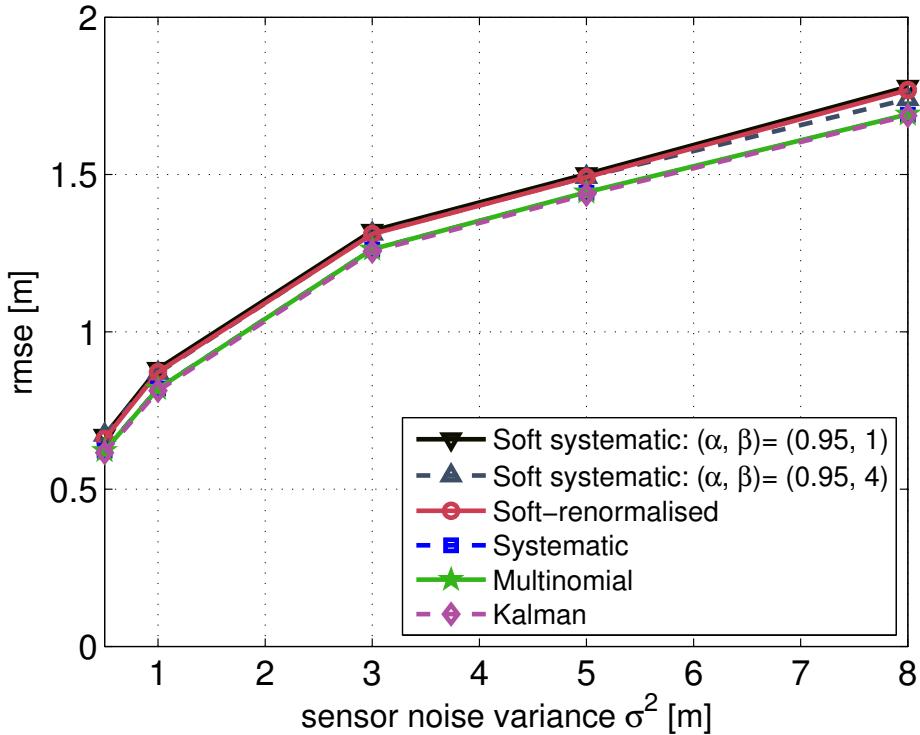


Figure 5.16: rmse versus the sensor noise variance σ^2 . The number of particles is $I = 1024$. The results are averaged over 20 instances, each having 50 time steps.

5.6.3 Offline implementation on real data

Here, the implementation of the proposed resamplers and their comparison with state-of-the-art PF methods is illustrated using real data tested offline. The observations are a set of 24 consecutive frames with a single target (a ball) appearing in the 4th frame and manoeuvring until the last frame. The original frames contain a black background with slight illumination at the bottom right corner. This region acts as a *bias region* for the particles when the target is absent. The original clean frames are manually corrupted with additive Gaussian noise. These noisy images can be seen in Fig. 5.17. It can be observed that it is difficult to locate the target by visualising a single frame, i.e., the target can be located after observing a

few frames in succession — a replica of a covert target tracking problem. These noisy images are then fed to the PFs operating on 500 particles. The evolution of the particles in these PFs is shown in Fig. 5.18 and Fig. 5.19. The particles are plotted on the ground truth. In each of these figures, the PFs from left to right are: (i) standard PF with soft resampling, (ii) standard PF with soft systematic resampling with $(\alpha, \beta) = (0.95, 10)$, (iii) standard PF with systematic resampling, and (iv) TBD PF with systematic resampling. The particle evolution with time in these figures is shown from top to bottom. For the TBD PF, the system specifications described in the caption of Fig. 5.9 are used here. For the soft systematic resampler, $(\alpha, \beta) = (0.95, 10)$. Simulation videos for this scenario are available at <http://ecs.victoria.ac.nz/Groups/CSP/PublicationExtras>.

Observation 1 - When target is absent:

It can be observed from Fig. 5.17 and Fig. 5.18 that the target is not present during frames 1 – 4. The standard PF operating with the traditional “hard” systematic resampler (the third column in Fig. 5.18) rapidly converges its particles towards the bias region (the bottom right corner of the image). That is to say, conventional resamplers do not effectively retain information over successive observations, a feature required to detect covert targets. The TBD PF was introduced to overcome this information retention problem. It can be seen in the TBD PF (the fourth column in Fig. 5.18), that most particles are non-existent (dead) when the target is not present and only a small proportion of persisting (alive) particles span the surveillance region for the new target. It can be understood that the TBD PF requires a *large number* of particles to span the frame in order to effectively detect the new target, but using more particles increases computational complexity. On the other hand, the standard PF operating on the proposed soft resampler (the first column in Fig. 5.18) leaves a more loosely-fitted particle cluster so there are enough particles spanning the image to detect the target presence. Obviously, over the course of time, the particles will converge around the bias

region but this convergence rate is much slower than that of the conventional resamplers.

Observation 2 - When target appears:

The soft resampler leaves a loose particle cloud by the 5th frame and hence the filter is able to quickly detect the target presence and lock the target, as seen in frames 6 – 8 of Fig. 5.18. The standard PF with the conventional systematic resampler cannot detect the ball until it is in the vicinity of the particle cloud (frame 9 in Fig. 5.19). In the TBD PF, it can be observed during frames 6–9 of Fig. 5.18 and Fig. 5.19 that the dead particles progressively come alive and start tracking the target. The soft systematic resampler acts as a hybrid between soft and systematic resamplers. Its advantage becomes evident when analysing the PF posterior representation.

Observation 3 - Accuracy of posterior approximation:

It can be observed in Fig. 5.19 that the particles obtain a good lock from frame 9. During the remaining frames in Fig. 5.19, a difference can be observed between the particle representation of the standard PF using the proposed soft resampler and the particle representation of the standard PF using the proposed soft systematic and the conventional systematic resamplers. It can be seen that alike the systematic resampler output, the particle cloud of the soft systematic resampler covers a good portion of the target (the ball). The low weight particles around the edges of this particle cloud are stochastically resampled. In contrast, the soft resampler continually discards these low weight particles. As a result, it can be observed that the particles located near the edge of the soft resampled particle cloud are always truncated as when compared to the particle cloud of the soft systematic and systematic resamplers. It can therefore be inferred that the soft systematic resampler represents the posterior more accurately than the soft resampler.

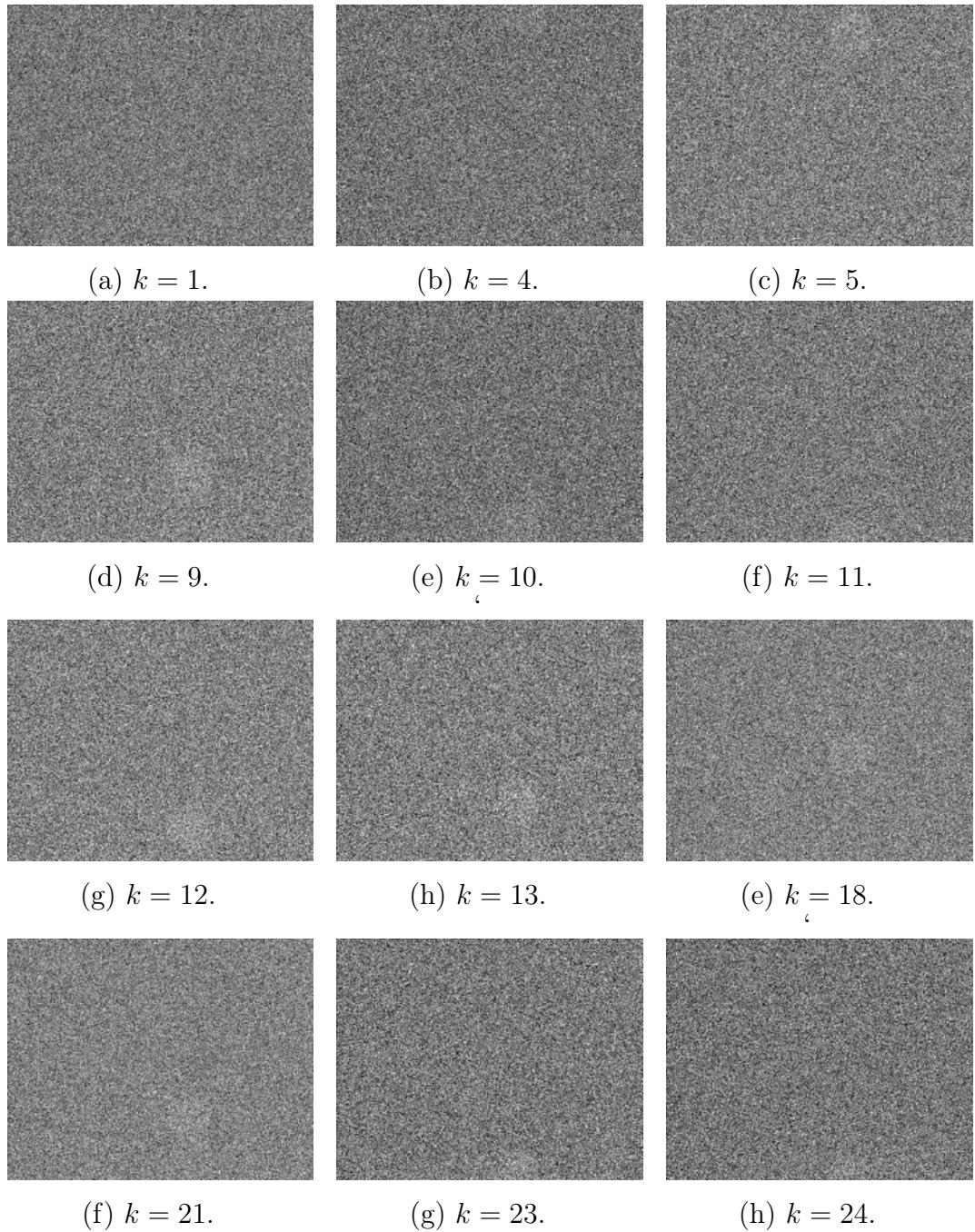


Figure 5.17: The 24 consecutive noisy snapshots fed to the PFs.

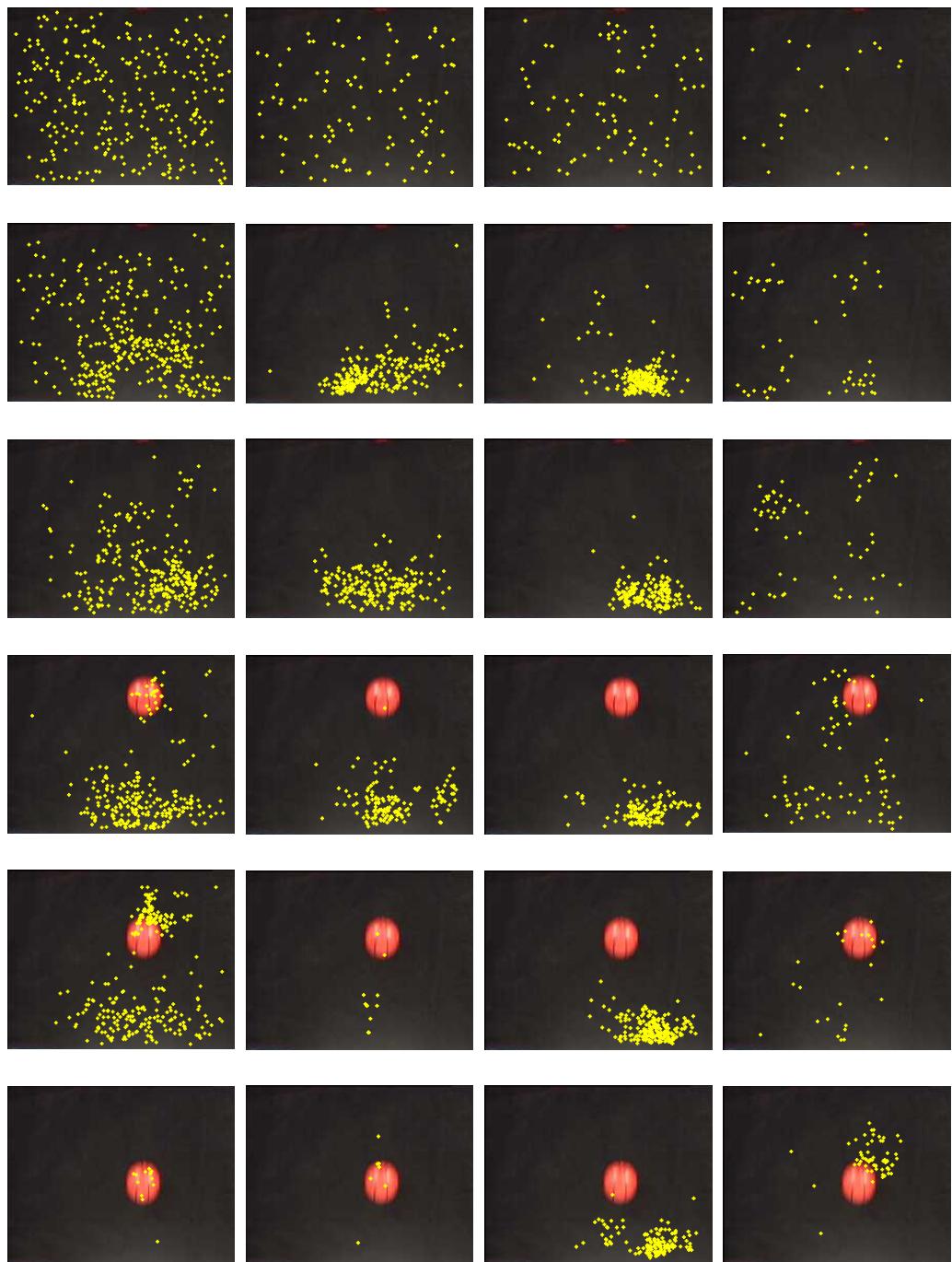


Figure 5.18: From top to bottom: time steps $k = 1, 2, 3, 6, 7, 8$. From left to right: (i) standard PF with soft resampling, (ii) standard PF with soft systematic resampling, (iii) standard PF with systematic resampling, and (iv) TBD PF with systematic resampling.

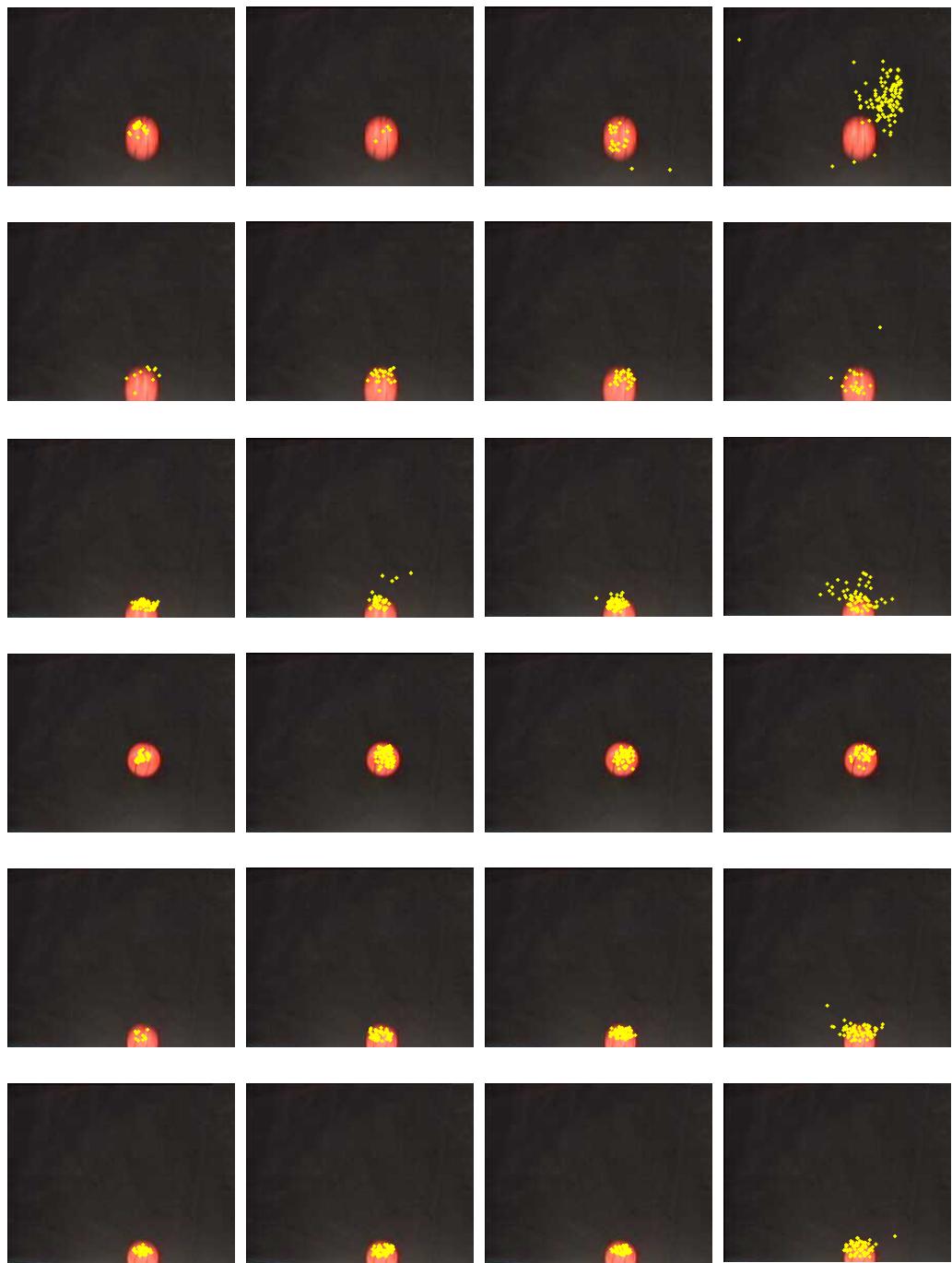


Figure 5.19: From top to bottom: time steps $k = 9, 10, 11, 19, 23, 24$. From left to right: (i) standard PF with soft resampling, (ii) standard PF with soft systematic resampling, (iii) standard PF with systematic resampling, and (iv) TBD PF with systematic resampling.

5.7 Conclusion and summary

This chapter addressed the role of PF resampling in retaining information over successive observations and in accurately representing the posterior. State-of-the-art PF resampling schemes do not effectively aid in information retention over time. Moreover, the standard PF using a conventional resampling scheme is not adequate for the detection of covert targets. The TBD PF overcomes this inadequacy but requires an excessive number of particles to effectively detect new targets. This chapter proposed soft resampling that proportionally redistributes and reweights the particles so as to preserve the information contained in low weights. This chapter also proposed soft systematic resampling that provides means to accurately represent the posterior and still preserve the low weights. Furthermore, a numerical mechanism that uses the optimal Kalman filter state estimates in a KS statistic framework was proposed for testing the resampler performance. The key advantages of the proposed resampling schemes are: a) improved information retention over time, b) preservation of low weights over time, and b) enhanced posterior approximation with fewer particles. The efficacy of the proposed methods was evaluated using the proposed KS test and in scenarios where retention of low weights is crucial. Referring back to section 1.3.2 of chapter 1, it can be stated that research goal 3 that aims to overcome problem 3 has been successfully achieved by the proposed resamplers, and the research goal 4 that aims to overcome problem 4 has been successfully achieved by the proposed KS testing scheme.

6

Conclusion and future extensions

This dissertation addressed an important research topic in science and engineering — the Bayesian multiple target tracking (MTT) [10, 17, 32, 35, 41]. Bayesian MTT is useful in many applications. Examples include aircraft and missile tracking using RADAR [1, 152, 154, 158], vessel and fish tracking using SONAR [2, 99, 100], acoustic and seismic activity tracking [155–157, 192, 214], space debris and spacecraft tracking [3, 106, 107], heart-beat tracking from ECG [4, 9, 147], and many others.

This thesis added to the body of knowledge in Bayesian MTT by proposing several original innovative ideas that overcome problems with state-of-the-art tracking systems. The central focus of this dissertation is on Bayesian MTT from phased sensor array signals [38] using the popular Bayesian approximation tool — the particle filter (PF) [21, 52, 54, 109]. The research conducted during the course of this thesis investigated and proposed novel solutions (presented in chapters 3 and 4) that overcame several challenges of the focus area — the Bayesian MTT for array signal processing applications

[37]. This research provided further motivation to address and propose leading edge solutions (presented in chapter 5) that overcame challenges of the PF resampling operation.

In this chapter, the thesis contributions and their merits are first summarised in section 6.1. Secondly, the research extensions and suggestions that could be addressed in the future are presented in section 6.2. Finally, concluding remarks are presented in section 6.3.

6.1 Contributions of this thesis

In this section, the contributions made by this thesis are summarised.

6.1.1 The MUSIC based MTT PF

Chapter 3 developed the *MUltiple SIgnal Classification (MUSIC) based MTT PF*, a Bayesian MTT filter for array processing applications.

Motivating problems in contemporary methods: The Bayesian MTT method of using a single target filter for each target separately [215, 216] suffers from the filter hijack problem [80, 126]. This problem is avoided in the joint filter [44, 124, 282]. However, the computational complexity in the joint filter increases significantly with the number of targets [45, 232]. The sub-optimal random finite set (RFS) filters [233] — the probability hypothesis density (PHD) filter [26, 32] and the multi-target multi-Bernoulli (MeMBer) filter [31–33, 49, 196], overcome this complexity problem. However, for tracking targets using phased sensor arrays, they require the sensor signals to be first converted to an image; this conversion leads to information and resolution loss.

Proposed solution: This thesis proposed the MUSIC based MTT PF, the key innovation of which is to use for the likelihood function, a proxy which has as its domain the domain of a single target. The MUSIC pseudo-spectrum is

such a function. Using MUSIC values as pseudo-weights allow the particles to operate in the space of a single target. However, tracking multiple targets in the single target state space results in the eventual convergence of all the particles around a single target. This problem was overcome by proportionally grouping the particles that track each individual target. This grouping was accomplished by first clustering [59] the particles and then imposing a cluster regulation scheme in the subsequent soft resampling [61] stage.

Merits of the proposal: The advantages of the proposed method as compared to state-of-the-art methods are as follows; *(i)* the computational complexity is lower than the joint filter by virtue of operating in the single target space, *(ii)* track accuracy is higher than that of the RFS filters by virtue of MUSIC that allows a more direct use of the sensor array data, *(iii)* close targets are effectively resolved by virtue of the super resolution property of MUSIC, and *(iv)* particle hijack is avoided by virtue of regulated clustering.

6.1.2 The MUSIC based MeMBer filter

Chapter 4 developed the *MUSIC based MeMBer filter*, a Bayesian RFS filter for array processing applications.

Motivating problems in contemporary methods: The RFS filters [26, 31, 32, 196] operate in the space of a single target and hence are more suitable for the tracking of many targets. However, they are sub-optimal and their performance could be poor at low signal-to-noise ratio (SNR) levels [161, 266]. Moreover, to date, the RFS filters do not possess a suitable mechanism to operate directly on the phased sensor array signals. This problem can be avoided using image models and image pre-processing techniques [50, 189, 191] which lead to information and resolution loss. Furthermore, in the “MeMBer filter for beamformer images,” the computation of the beamformer values for all the pixels that fall within a region of influence for each particle is computationally expensive.

The proposed solution: This thesis proposed the MUSIC based MeMBer filter, the key innovation of which is in adapting the MeMBer filter for image observations to phased sensor array observations. This is achieved by using for the likelihood in the MeMBer filter for image observations, the MUSIC pseudo-spectrum — a function which has as its domain the state space of a single target. The remainder of the algorithm is unchanged.

Merits of the proposal: The simple modification of using MUSIC as a proxy to the likelihood in the MeMBer filter adds considerable merit in its application to array signal processing. The advantages of the method as compared to state-of-the-art methods are as follows; *(i)* improved track accuracy by virtue of using MUSIC which allows the replacement of the image observation with array signals, *(ii)* lower computational complexity by virtue of using MUSIC, i.e., the region of influence is no longer relevant in the proposed technique and the filter requires only a single MUSIC value computation for each particle, and *(iii)* effective separation of spatially close targets by virtue of the super resolution property of MUSIC.

6.1.3 Soft resampling

Chapter 5 developed the *soft* and the *soft systematic* resampling, a PF subroutine that aids in the replenishment of large weight particles.

Motivating problems in contemporary methods: The conventional resamplers [52, 54, 118–120, 122] do not effectively aid in information accumulation over successive observations. As a result, these resamplers could easily eliminate a low weight particle that is gradually acquiring target information. Moreover, the standard PF using a conventional resampler is not suitable for covert target tracking as such a target need to be tracked for some duration (i.e., its information need to be retained) before declaring its presence. This insufficiency is overcome using the track-before-detect (TBD) PF that requires the use of many particles to effectively detect new targets. This requirement increases the computational burden.

The proposed solution: This thesis proposed soft resampling, the key innovation of which is the proportional redistribution of particle information among all the other particles and the preservation of information contained in the low weight particles. The scheme retains particle information for a longer duration. This thesis also proposed soft systematic resampling (an improvement to soft resampling) that properly treats the tails of the posterior. The scheme prevents certain loss of those particles that lie near the boundary of the particle set by stochastically resampling a few lower weight particles.

Merits of the proposal: The advantages of the proposed methods as compared to state-of-the-art methods are as follows; *(i)* improved information retention over time, *(ii)* faster target lock when the target is covert or when the prior is not correct, *(iii)* a more accurate posterior approximation in soft systematic resampling, and *(iv)* favourable track accuracy.

6.1.4 The KS statistic

Chapter 5 developed a method of practically evaluating the PF approximation error using the well-known *Kolmogorov-Smirnov (KS)* statistic [273].

Motivating problem: It is known that the PF provides only an approximate representation of the true posterior. A numerical measure to test PF representation accuracy is not available in practice. The PF resampling subroutine contributes significantly to inaccuracies in PF approximation due to unavoidable particle rejection and replacement operations. Therefore it is important to numerically assess the faithfulness of PF resampling in approximating the posterior.

The proposed solution: This thesis proposed that the PF accuracy of posterior representation can be quantitatively evaluated for linear Gaussian models in accordance to its Kolmogorov-Smirnov (KS) statistic agreement [273] with the optimal Kalman filter. The KS test is under-utilised in the PF

literature and has never been used to measure PF resampling performance. Since KS testing [62, 275, 276] cannot be directly used, this thesis proposed to de-mean and de-correlate the resampled particles using the Kalman mean and covariance. Then the one-sample KS test [62] is conducted along each dimension separately by comparing the cumulative sum of the (sorted) particle weights and the error function of $\mathcal{N}(0, 1)$. Finally, the maximum KS deviation over all the dimensions is taken. This method was used to test the performance of the proposed resampler in chapter 5.

Merits of the proposal: If the PF representation is identical to the Kalman filter distribution, then the de-mean and de-correlation of particles will result in a set of uncorrelated particles normally distributed with zero mean and unit variance. Hence the KS deviation of the cumulative distribution function (cdf) of the (sorted) particle weights from the error function of $\mathcal{N}(0, 1)$ will be small. Conversely, if the PF representation is not identical to the Kalman filter distribution, then the KS deviation will be large. A numerical measure to test the PF representation accuracy is now available in practice.

6.2 Future extensions of this thesis

In this section, the future research extensions and suggestions of this thesis are outlined.

6.2.1 Bayesian filtering for array processing

Some future research ideas identified in the MUSIC based MTT PF proposed in chapter 3 and in the MUSIC based MeMBER filter proposed in chapter 4 are now presented.

Hierarchical clustering

This idea relates to the MUSIC based MTT PF presented in chapter 3. The proposed filter used k-means clustering to group (or separate) the particles so

as to avoid particle hijack — a general problem that becomes evident when particles in the same space try to localise multiple targets. Since the actual number of targets is not known, the number of clusters formed is chosen to be the maximum number of targets the filter can track. A regulation scheme is then imposed to restrict any one cluster from containing more than a fixed proportion of the total number of particles.

An alternative to k-means clustering is the well known hierarchical clustering [283] that is known to provide a linkage criterion which specifies the dissimilarity of the particles as a function of their pairwise Euclidean distances. By observing the dendrogram formed by this clustering algorithm (or finding that instance where a certain number of clusters take a long time to merge), one is able to *approximately* infer the number of tight clusters, i.e., the actual number of targets in the scene. The number of clusters formed would then be approximately equal to the number of actual targets. A cluster regulation scheme can then be imposed on these hierarchically clustered particles.

Weight based clustering

This idea relates to the MUSIC based MTT PF presented in chapter 3. The proposed filter used regulated k-means clustering through soft resampling to avoid particle hijack and to sustain weak targets — targets that generate/reflect weak signals. However, in scenarios where there are only a few particles near a weak target, the soft resampler would not replenish sufficient particles near that target. As a result, the cluster near the target will contain an inadequate number of particles. Although these particles (and hence the target) can be sustained by the filter, the track accuracy could be affected. This problem could be solved by leveraging the clustering procedure on the weights of the particles. This can be achieved by replacing the k-means clustering with a method like gravitational clustering [284] wherein the mass (MUSIC pseudo-weight) of each particle exerts a gravitational at-

traction on the nearby masses (weights) such that the particle with a lower mass, although further away, will be eventually attracted to the particle with a higher mass. This will aid in having a sufficient number of particles near a weak target given that the particles further from the target are not tracking another target. Additionally, this procedure could improve the detection of covert targets, especially when the number of targets appearing over time is large.

Statistical robust signal processing

The ability of the MUSIC based MTT PF of chapter 3 and the MUSIC based MeMBer filter of chapter 4 to withstand deviations in noise and reverberation model assumptions could be improved using statistical robust signal processing [285]. Robust estimation is known to successfully handle situations where the actual system parameters deviate widely from their assumed distributions. Using the robust MUSIC algorithm presented in [286, 287], the possibility of developing a robust-MUSIC based MTT PF and a robust-MUSIC based MeMBer filter could be investigated.

MUSIC as observation in RFS filters

The MTT filters in chapters 3 and 4 that were designed to operate in single target space used MUSIC as a proxy for the likelihood function. This approach facilitated in accelerating the filtering process and in resolving close targets more effectively. An alternative approach is to use a MUSIC image as the observation feed to the RFS filters. The applicability of this idea could be tested comprehensively.

6.2.2 ESPRIT based measurement feed for PHD filter

Another research suggestion relating to Bayesian MTT using phased sensor arrays is now presented. It is known that the observation feed to the PHD

filter can only be a finite set of point target detections. This impedes the direct use of the filter on array data. This impediment is usually circumvented by a two-stage conversion process; a) converting the array signals into images [50, 189, 191], and b) pre-processing [192] the images to obtain the observation feed. This procedure causes a two-stage information loss.

An alternative approach to convert the array data (that has not been explored in the literature) is as follows. For array processing localisation applications, the well known estimation of signal parameters via rotational invariant techniques (ESPRIT) [288, 289] algorithm provides point estimates like positions, angles of arrival (AOA), etc., directly from the array data. These ESPRIT estimates can be used as the observation feed to the PHD filter. The advantage of this proposition is that there is a more direct use of the array data because the conversion is now single-stage. This could improve the filter accuracy at low SNR. However, the ESPRIT algorithm requires the knowledge about the actual number of targets to obtain the eigenvectors that lie in the signal space. One approach to address this requirement is to use the minimum-mean-square-error (MMSE) based minimum description length (MDL) algorithm [258] (described and used in section 3.5.2 of chapter 3). A second approach is as follows: the ESPRIT algorithm can localise at most $M - 1$ targets, where M is the number of sensors. If the algorithm is directed to generate $M - 1$ target estimates, then $(M - 1) - N$ (let N be the actual number of targets) of these estimates would be incorrect (false) detections. The PHD filter will eliminate these false detection peaks as time progresses, provided that the false detections are independent from one time to the next. The possibility of operating the PHD filter with its observation feed being the ESPRIT output estimates could be investigated. The idea could be extended to the well known root-MUSIC [246] algorithm which also generates point estimates about the target states. Moreover, the time independency of false detections generated by ESPRIT could also be investigated.

6.3 Conclusion

The central focus of this dissertation is on Bayesian MTT using PFs for array processing applications. This dissertation addressed several challenges of state-of-the-art methods in; a) Bayesian filtering for array processing, and b) PF resampling. This concluding chapter first summarised the contributions made by this thesis and then outlined research extensions and ideas which could be investigated in the future.

Bibliography

- [1] M. V. Finn, “Unified data fusion applied to monopulse tracking,” DTIC Document, Tech. Rep., 1999.
- [2] A. Xydes, M. Moline, C. Lowe, T. J. Farrugia, and C. Clark, “Behavioral characterization and particle filter localization to improve temporal resolution and accuracy while tracking acoustically tagged fishes,” *J. Elsevier Ocean Engineering*, vol. 61, pp. 1–11, 2013.
- [3] J. Huang and W. Hu, “MCMC-Particle-based group tracking of space objects within Bayesian framework,” *J. Elsevier Adv. in Space Research*, vol. 53, no. 2, pp. 280–294, 2014.
- [4] S. H. Salleh, H. S. Hussain, T. T. Swee, C. M. Ting, A. M. Noor, S. Pipatsart, J. Ali, and P. P. Yupapin, “Acoustic cardiac signals analysis: a Kalman filter-based approach.” *J. Nanomedicine*, vol. 7, pp. 2873–2881, 2011.
- [5] Y. Bar Shalom, *Tracking and data association*. Academic Press Professional, Inc., 1987.
- [6] E. Ozkan, C. Lundquist, and F. Gustafsson, “A Bayesian approach to jointly estimate tire radii and vehicle trajectory,” in *Proc. IEEE Intell. Transp. Syst.*, 2011, pp. 1622–1627.
- [7] C. Fernández-Prades, P. Closas, and J. Fernández-Rubio, “Rao-Blackwellized variable rate particle filtering for handset tracking in

- communication and sensor networks,” in *Proc. European Signal Process.*, 2007, pp. 111–115.
- [8] A. Rahni, K. Wells, E. Lewis, M. Guy, and B. Goswami, “An iterative particle filter approach for respiratory motion estimation in nuclear medicine imaging,” in *Proc. SPIE Medical Imaging*, 2011, pp. 79 624C–79 624C.
 - [9] B. Balasingam, M. Forouzanfar, M. Bolić, H. Dajani, V. Groza, and S. Rajan, “Arterial blood pressure parameter estimation and tracking using particle filters,” in *Proc. IEEE Med. Meas. and Appl.*, 2011, pp. 473–476.
 - [10] Y. B. Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
 - [11] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House Norwood, 1999, vol. 685.
 - [12] A. F. M. Smith and A. E. Gelfand, “Bayesian statistics without tears: A sampling–resampling perspective,” *Am. Statist.*, vol. 46, no. 2, pp. 84–88, 1992.
 - [13] W. Bolstad, *Introduction to Bayesian statistics*. WILEY-INTERSCIENCE, 2004.
 - [14] S. R. Eddy, “What is Bayesian statistics?” *J. Nature Biotechnology*, vol. 22, no. 9, pp. 1177–1178, 2004.
 - [15] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.
 - [16] T. Bayes and Price, “An essay towards solving a problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, FRS communicated by

- Mr. price, in a letter to John Canton, AMFRS,” *Trans. Royal Society Philosophical*, vol. 53, pp. 370–418, 1763.
- [17] T. Bayes, D. Bunn, H. Raiffa, R. Schlaifer, and D. Winterfeldt, “An essay toward solving a problem in the doctrine of chances,” *Philosophical Trans. Royal Society London*, vol. 53, 1984.
 - [18] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.
 - [19] Y. C. Ho and R. Lee, “A Bayesian approach to problems in stochastic estimation and control,” *IEEE Trans. Automatic Control*, vol. 9, no. 4, pp. 333–339, 1964.
 - [20] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
 - [21] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to non-linear/non-Gaussian Bayesian state estimation,” in *Proc. Inst. Elect. Eng.*, vol. 140, no. 2, 1993, pp. 107–113.
 - [22] N. Metropolis and S. Ulam, “The Monte Carlo method,” *J. Am. Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949.
 - [23] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *J. Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
 - [24] J. E. Handschin and D. Q. Mayne, “Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering,” *J. Control*, vol. 9, no. 5, pp. 547–559, 1969.
 - [25] J. E. Handschin, “Monte Carlo techniques for prediction and filtering of non-linear stochastic processes,” *J. Elsevier Automatica*, vol. 6, no. 4, pp. 555–563, 1970.

BIBLIOGRAPHY

- [26] R. P. S. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [27] S. Mori, C. Y. Chong, E. Tse, and R. P. Wishner, “Tracking and classifying multiple targets without identification,” *IEEE Trans. Automatic Control*, vol. 31, no. 5, pp. 401–409, 1986.
- [28] I. R. Goodman, *Mathematics of data fusion*. Springer, 1997, vol. 37.
- [29] J. Goutsias, R. P. S. Mahler, and H. T. Nguyen, *Random Sets: Theory and Applications*. Springer New York, 1997.
- [30] H. Fei and I. Reid, “Joint Bayes filter: A hybrid tracker for non-rigid hand motion recognition,” in *J. Springer Computer Vision*, 2004, pp. 497–508.
- [31] B. T. Vo, B. N. Vo, and A. Cantoni, “The cardinality balanced multi-target multi-Bernoulli filter and its implementations,” *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, 2009.
- [32] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House Publishers, 2007, vol. 685.
- [33] B. T. Vo, B. N. Vo, and A. Cantoni, “On multi-Bernoulli approximations to the Bayes multi-target filter,” in *Proc. Inf. Fusion*, 2007.
- [34] R. P. S. Mahler, ““Statistics 101” for multisensor, multitarget data fusion,” *IEEE Aerosp. and Electron. Syst. Mag.*, vol. 19, no. 1, pp. 53–64, 2004.
- [35] L. D. Stone, C. A. Barlow, and T. L. Corwin, *Bayesian multiple target tracking*. Artech House, 1999.
- [36] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*. Artech House, 2013.

- [37] D. H. Johnson and D. E. Dudgeon, *Array signal processing: concepts and techniques*. Simon & Schuster Pub., 1992.
- [38] M. Viberg and B. Ottersten, “Sensor array processing based on subspace fitting,” *IEEE Trans. on Signal Process.*, vol. 39, no. 5, pp. 1110–1121, 1991.
- [39] M. R. Bai, A. Agarwal, C. C. Chen, and Y. C. Wang, “Bayesian approach of nearfield acoustic reconstruction with particle filters,” *J. Acoust. Soc. of America*, vol. 133, no. 6, pp. 4032–4043, 2013.
- [40] B. A. Yocom, J. M. Aughenbaugh, and B. R. L. Cour, “Range-sensitive Bayesian passive sonar tracking,” in *Proc. Inf. Fusion*, 2010, pp. 1–10.
- [41] V. Trees and L. Harry, *Detection, Estimation, and Modulation theory*. John Wiley & Sons, 2004.
- [42] D. Avitzour, “Stochastic simulation Bayesian approach to multitarget tracking,” in *Proc. IEE Radar, Sonar and Navig.*, vol. 142, no. 2, 1995, pp. 41–44.
- [43] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, “Multi-target tracking using joint probabilistic data association,” in *Proc. IEEE Decision and Control Adaptive Process.*, vol. 19, 1980, pp. 807–812.
- [44] C. Kreucher, K. Kastella, and A. O. Hero III, “Multitarget tracking using the joint multitarget probability density,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1396–1414, 2005.
- [45] F. Daum and J. Huang, “Curse of dimensionality and particle filters,” in *Proc. IEEE Aerosp. Conf.*, 2003, pp. 1979–1993.
- [46] T. Bengtsson, P. Bickel, and B. Li, “Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems,” in *J. IMS Probability and Statistics*, 2008, vol. 2, pp. 316–334.

BIBLIOGRAPHY

- [47] W. Yi, M. Morelande, L. Kong, and J. Yang, “A computationally efficient particle filter for multi-target tracking using an independence approximation,” *IEEE Trans. on Signal Process.*, vol. 61, no. 4, pp. 843–856, 2013.
- [48] J. Vermaak, S. J. Godsill, and P. Pérez, “Monte Carlo filtering for multi target tracking and data association,” *IEEE Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 309–332, 2005.
- [49] B. N. Vo, B. T. Vo, N. T. Pham, and D. Suter, “Bayesian multi-object estimation from image observations,” in *Proc. Inf. Fusion*, 2009, pp. 890–898.
- [50] D. E. Clark, I. T. Ruiz, Y. Petillot, and J. Bell, “Particle PHD filter multiple target tracking in sonar image,” *IEEE Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 409–416, 2007.
- [51] M. Tobias and A. D. Lanterman, “Techniques for birth-particle placement in the probability hypothesis density particle filter applied to passive radar,” *Proc. IET Radar, Sonar & Navig.*, vol. 2, no. 5, pp. 351–365, 2008.
- [52] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
- [53] J. Liu and R. Chen, “Sequential Monte Carlo Methods for Dynamic Systems,” *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.
- [54] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. M. Bugallo, and J. Miguez, “Particle filtering,” *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, 2003.
- [55] D. J. Salmond and H. Birch, “A particle filter for track-before-detect,” in *Proc. IEEE American Control Conf.*, vol. 5, 2001, pp. 3755–3760.

- [56] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.
- [57] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986.
- [58] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. Berkeley Symp. Mathematical Statist. and Probability*, vol. 1, no. 281-297, 1967, p. 14.
- [59] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, Inc., 1975.
- [60] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer Series in Statistics, 2001, vol. 2, no. 1.
- [61] P. B. Choppala, P. D. Teal, and M. R. Frean, “Soft resampling for improved information retention in particle filtering,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 13, 2013, pp. 4036–4040.
- [62] J. A. Peacock, “Two-dimensional goodness-of-fit testing in astronomy,” *Royal Astronomical Society*, vol. 202, pp. 615–627, 1983.
- [63] S. M. Kay, “Fundamentals of statistical signal processing: Detection theory,” 1998.
- [64] A. H. Jazwinski, “Stochastic processes and filtering theory,” *Mathematics in Science and Eng.*, no. 64, 1970.
- [65] B. P. Carlin, N. G. Polson, and D. S. Stoffer, “A Monte Carlo approach to nonnormal and nonlinear state-space modeling,” *J. Am. Statistical Asso.*, vol. 87, no. 418, pp. 493–500, 1992.
- [66] W. Zhang, *State-space search: Algorithms, complexity, extensions, and applications*. Springer, 1999.

- [67] C. W. Gardiner *et al.*, *Handbook of stochastic methods*. Springer Berlin, 1985, vol. 3.
- [68] Z. Ghahramani, “An introduction to hidden Markov models and Bayesian networks,” *J. World Scientific Pattern Recog. Artificial Intell.*, vol. 15, no. 01, pp. 9–42, 2001.
- [69] J. Burlet, T. D. Vu, and O. Aycard, “Grid-based localization and online mapping with moving object detection and tracking,” 2007.
- [70] K. Kastella, “Event-averaged maximum likelihood estimation and mean-field theory in multitarget tracking,” *IEEE Trans. Automatic Control*, vol. 40, no. 6, pp. 1070–1074, 1995.
- [71] C. M. Kreucher, K. D. Kastella, and A. O. Hero III, “Tracking multiple targets using a particle filter representation of the joint multitarget probability density,” in *Proc. SPIE, Optical Science and Technology*, 2003, pp. 258–269.
- [72] K. D. Kastella, “Joint multitarget probabilities for detection and tracking,” in *SPIE Proc. AeroSense*. International Society for Optics and Photonics, 1997, pp. 122–128.
- [73] D. Avitzour, “A maximum likelihood approach to data association,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 28, no. 2, pp. 560–566, 1992.
- [74] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Trans. on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [75] R. E. Bellman, “Dynamic programming and Lagrange multipliers,” *Proc. National Academy Sciences*, vol. 42, no. 10, p. 767, 1956.
- [76] ——, *Adaptive control processes: A guided tour*. Princeton University Press, 1961, vol. 4.

- [77] N. Weiner, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*, vol. 8.
- [78] G. Welch and G. Bishop, “An introduction to the Kalman filter,” Tech. Rep., 1995.
- [79] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and practice*. Prentice Hall, 1993.
- [80] A. Bhatia, “Hierarchical Charged Particle Filter for Multiple Target Tracking,” Ph.D. dissertation, North Carolina State University, 2012.
- [81] R. E. Kalman and R. S. Bucy, “New results in linear filtering and prediction theory,” *J. Basic Eng.*, vol. 83, no. 1, pp. 95–108, 1961.
- [82] J. George, “Robust Kalman-Bucy filter,” *IEEE Trans. Automatic Control*, vol. 58, no. 1, pp. 174–180, 2013.
- [83] R. Lefferts, “Altitude estimation using asynchronous α - β tracking filters,” *IEEE Trans. Aerosp. and Electron. Syst.*, no. 4, pp. 469–477, 1982.
- [84] P. R. Kalata, “The tracking index: A generalized parameter for Alpha-Beta and Alpha-Beta-Gamma target trackers,” *IEEE Tran. Aerosp. Electron. Syst.*, no. 2, pp. 174–182, 1984.
- [85] S. F. Schmidt, “Application of state-space methods to navigation problems,” *J. Adv. Control Syst.*, vol. 3, pp. 293–340, 1966.
- [86] V. E. Beneš, “New exact nonlinear filters with large Lie algebras,” *J. Elsevier Systems & Control Lett.*, vol. 5, no. 4, pp. 217–221, 1985.
- [87] F. E. Daum, “New exact nonlinear filters,” *Proc. Bayesian Analysis of Time Series and Dynamic Models*, vol. 94, pp. 199–226, 1988.
- [88] L. A. Zadeh, “Optimum nonlinear filters,” *J. Applied Physics*, vol. 24, no. 4, pp. 396–404, 2004.

- [89] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
- [90] F. Gustafsson and G. Hendeby, “Some relations between extended and unscented Kalman filters,” *IEEE Trans. on Signal Process.*, no. 99, pp. 545–555, 2012.
- [91] H. Tanizaki and R. S. Mariano, “Nonlinear filters based on Taylor series expansions,” *Comm. Stat. Theory and Methods*, vol. 25, no. 6, pp. 1261–1282, 1996.
- [92] S. J. Julier and J. K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Proc. Int. Symp. Aerospace Defense Sensing, Simul. and Controls*, vol. 3, no. 26, 1997, pp. 3–2.
- [93] ——, “A general method for approximating nonlinear transformations of probability distributions,” Tech. Rep., 1996.
- [94] ——, “Unscented filtering and nonlinear estimation,” *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [95] E. Wan and R. V. D. Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proc. IEEE Adaptive Systems for Signal Process., Comm., and Control*, 2000, pp. 153–158.
- [96] F. Gustafsson and G. Hendeby, “On nonlinear transformations of stochastic variables and its application to nonlinear filtering,” in *Proc. IEEE Acoustic., Speech and Signal Proc.*, 2008, pp. 3617–3620.
- [97] T. Kailath, A. H. Sayed, and B. Hassibi, “Linear estimation,” 2000.
- [98] T. I. Fossen and T. Perez, “Kalman filtering for positioning and heading control of ships and offshore rigs,” *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 32–46, 2009.

- [99] D. B. Cousins, “Underwater model-based processing for SLAM and environment characterization,” in *Proc. IEEE Aerosp.*, 2008, pp. 1–10.
- [100] K. S. Rao, “Pseudo linear Kalman filter for underwater target location using intercept sonar measurements,” in *Proc. IEEE Position, location and navig.*, 2006, pp. 1036–1039.
- [101] S. Chumchean, A. Seed, and A. Sharma, “Correcting of real-time radar rainfall bias using a Kalman filtering approach,” *J. Elsevier Hydrology*, vol. 317, no. 1, pp. 123–137, 2006.
- [102] D. Carevic, “Kalman filter-based approach to target detection and target-background separation in ground-penetrating radar data,” in *Proc. ISOP AeroSense*, 1999, pp. 1284–1288.
- [103] C. Yardim, P. Gerstoft, and W. S. Hodgkiss, “Tracking of geoacoustic parameters using Kalman and particle filters,” *J. Acoust. Soc. Am.*, vol. 125, no. 2, pp. 746–760, 2009.
- [104] X. Deng and J. Hu, “A new Kalman filter for tracking event of seismic record,” in *Proc. IEEE Image Signal Process.*, no. 13, 2013, pp. 1143–1147.
- [105] J. Traa and P. Smaragdis, “A wrapped Kalman filter for azimuthal speaker tracking,” *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1257–1260, 2013.
- [106] E. J. Lefferts, F. L. Markley, and M. D. Shuster, “Kalman filtering for spacecraft attitude estimation,” *J. Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982.
- [107] M. T. Nasri and W. Kinsner, “Extended Kalman filtering for pico-satellites attitude determination,” in *Proc. IEEE Elec. Comp. Eng.*, 2013, pp. 1–5.

- [108] G. Kitagawa, “Non-Gaussian state-space modeling of non-stationary time series,” *J. Am. Statist. Asso.*, vol. 82, no. 400, pp. 1032–1041, 1987.
- [109] A. Doucet, N. D. Freitas, and N. Gordon, *An introduction to Sequential Monte Carlo methods in practice*. Springer New York, 2001, vol. 1.
- [110] A. Doucet, S. J. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *J. Springer Statistic. Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [111] S. J. Godsill, A. Doucet, and M. West, “Monte Carlo smoothing for nonlinear time series,” *J. Am. Statist. Asso.*, vol. 99, no. 465, 2004.
- [112] M. Carter and B. V. Brunt, *The Lebesgue-Stieltjes integral*. Springer.
- [113] J. R. Blum, D. L. Hanson, and L. H. Koopmans, “On the strong law of large numbers for a class of stochastic processes,” *J. Springer Probability Theory and Related Fields*, vol. 2, no. 1, pp. 1–11, 1963.
- [114] Z. Chen, “Bayesian filtering: From Kalman filters to particle filters, and beyond,” *J. Statistics*, vol. 182, no. 1, pp. 1–69, 2003.
- [115] T. K. Chandra, “Uniform integrability in the Cesaro sense and the weak law of large numbers,” *J. Indian Statistics*, pp. 309–317, 1989.
- [116] D. Crisan and A. Doucet, “A survey of convergence results on particle filtering methods for practitioners,” *IEEE Tran. Signal Process.*, vol. 50, no. 3, pp. 736–746, 2002.
- [117] X. L. Hu, T. Schön, and L. Ljung, “A basic convergence result for particle filtering,” *IEEE Tran. Signal Process.*, vol. 56, no. 4, pp. 1337–1348, 2008.
- [118] J. D. Hol, T. B. Schön, and F. Gustafsson, “On Resampling Algorithms for Particle Filters,” in *Proc. Nonlinear Stat. Signal Process. Workshop*, 2006.

- [119] R. Douc and O. Cappé, “Comparison of resampling schemes for particle filtering,” in *Proceedings ISPA 4th International Symposium on Image and Signal Processing and Analysis*, 2005, pp. 64–69.
- [120] M. Bolić, P. M. Djurić, and S. Hong, “Resampling algorithms for particle filters: A computational complexity perspective,” *J. EURASIP Applied Signal Process.*, vol. 2004, pp. 2267–2277, 2004.
- [121] J. D. Hol, “Resampling in particle filters,” Tech. Rep., 2004.
- [122] M. Bolić, P. M. Djurić, and S. Hong, “Resampling algorithms and architectures for distributed particle filters,” *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2442–2450, 2005.
- [123] K. Pearson, “The problem of the random walk,” *J. Nature*, vol. 72, no. 1865, p. 294, 1905.
- [124] M. R. Morelande, C. M. Kreucher, and K. Kastella, “A Bayesian approach to multiple target detection and tracking,” *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1589–1604, 2007.
- [125] W. Du and J. Piater, “Tracking by cluster analysis of feature points and multiple particle filters,” *Pattern Recognition and Image Analysis*, pp. 701–710, 2005.
- [126] J. Vermaak, A. Doucet, and P. Pérez, “Maintaining multimodality through mixture tracking,” in *Proc. IEEE 9th Int. Conf. Computer Vision*, 2003, pp. 1110–1116.
- [127] M. Orton and W. Fitzgerald, “A Bayesian approach to tracking multiple targets using sensor arrays and particle filters,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 216–223, 2002.
- [128] A. Giremus, J. Tournieret, and P. Djuric, “An improved regularized particle filter for gps/ins integration,” in *Proc. IEEE Signal Process. Adv. Wireless Comm.*, 2005, pp. 1013–1017.

BIBLIOGRAPHY

- [129] N. Vaswani and D. Das, “Particle filter with efficient importance sampling and mode tracking (pf-eis-mt) and its application to landmark shape tracking,” in *Proc. IEEE Signals, Syst. and Computers*, 2007, pp. 720–724.
- [130] N. Yang, B. Ding, and J. Zhang, “Improved sampling importance resampling particle filter optimized by Tabu search,” in *Proc. IEEE Artificial Intell. and Comput. Intell.*, vol. 3, 2010, pp. 359–363.
- [131] C. Shen, D. A. H. Hengel, B. Anthony, and M. J. Brooks, “Enhanced importance sampling: Unscented auxiliary particle filtering for visual tracking,” in *J. Springer Adv. in Artificial Intell.*, 2005, pp. 180–191.
- [132] P. Torma and C. Szepesvári, “Local importance sampling: A novel technique to enhance particle filtering,” *J. Multimedia*, vol. 1, no. 1, pp. 32–43, 2006.
- [133] M. Rollason and D. Salmond, “A particle filter for track-before-detect of a target with unknown amplitude,” in *Proc. IEE Target Tracking: Algorithms and Applications*, vol. 1, 2001, pp. 14/1–14/4.
- [134] M. Fallon and S. J. Godsill, “Multi target acoustic source tracking using track before detect,” in *Proc. IEEE Workshop Appl. of Signal Process. to Audio and Acoust.*, 2007, pp. 102–105.
- [135] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *J. of the American Statistical Association*, pp. 590–599, 1999.
- [136] J. Carpenter, P. Clifford, and P. Fearnhead, “An improved particle filter for nonlinear problems,” *Proc. IEE Radar, Sonar and Navigation*, vol. 146, pp. 2–7, 1999.
- [137] D. Ustebay, M. Coates, and M. Rabbat, “Distributed auxiliary particle filters using selective gossip,” in *Proc. IEEE Acoust., Speech and Signal Process.*, 2011, pp. 3296–3299.

- [138] G. Johannesson, B. Hanley, and J. Nitao, “Dynamic Bayesian models via Monte Carlo—An introduction with examples,” *Lawrence Livermore National Laboratory, UCRL-TR-207173*, 2004.
- [139] R. V. D. Merwe, A. Doucet, N. D. Freitas, and E. Wan, “The unscented particle filter,” *Adv. Neural Inf. Process. Systems*, pp. 584–590, 2001.
- [140] L. Xue, S. Gao, and J. Wang, “Research on robust unscented regularized particle filtering,” in *Proc. IEEE Inf. Theory Inf. Security*, 2010, pp. 790–793.
- [141] C. Andrieu, N. D. Freitas, A. Doucet, and M. Jordan, “An introduction to MCMC for machine learning,” *J. Machine Learning*, vol. 50, no. 1, 2003.
- [142] M. R. Morelande, “A marginalised particle filter for bearings-only tracking,” in *Proc. IEEE Acoust., Speech and Signal Process.*, 2013, pp. 6298–6302.
- [143] T. Schön, F. Gustafsson, and P. J. Nordlund, “Marginalized particle filters for mixed linear/nonlinear state-space models,” *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2279–2289, 2005.
- [144] N. D. Freitas, “Rao-Blackwellised particle filtering for fault diagnosis,” in *Proc. IEEE Aerosp.*, vol. 4, 2002, pp. 4/1767–4/1772.
- [145] P. Nordlund and F. Gustafsson, “Marginalized particle filter for accurate and reliable terrain-aided navigation,” *IEEE Trans. Aerosp. and Electron. Systems*, vol. 45, no. 4, pp. 1385–1399, 2009.
- [146] C. Fritzsche, T. Schon, and A. Klein, “The marginalized auxiliary particle filter,” in *Proc. IEEE Comput. Adv. Multi-Sensor Adaptive Process.*, 2009, pp. 289–292.
- [147] P. D. Teal, “Bayesian phase tracking for multiple pulse signals,” *J. Elsevier Signal Process.*, vol. 90, no. 6, pp. 2050–2059, 2010.

- [148] Y. Boers and J. N. Driesssen, “Interacting multiple model particle filter,” *Proc. IEE Radar, Sonar and Navig.*, vol. 150, no. 5, pp. 344–349, 2003.
- [149] L. Hong, N. Cui, M. Bakich, and J. R. Layne, “Multirate interacting multiple model particle filter for terrain-based ground target tracking,” in *Proc. IEE Control Theory and Appl.*, vol. 153, no. 6, 2006, pp. 721–731.
- [150] F. Maurelli, Y. Petillot, A. Mallios, P. Ridao, and S. Krupinski, “Sonar-based AUV localization using an improved particle filter approach,” in *Proc. IEEE Oceans*, 2009, pp. 1–9.
- [151] C. V. S. Rodrigues and W. S. Filho, “Particle filter applied in the real-time bearings-only tracking problem of a sonar target,” in *Proc. Acoust.*, 2008, pp. 2453–2458.
- [152] Y. Boers and J. N. Driesssen, “Multitarget particle filter track before detect application,” in *Proc. IEE Radar, Sonar and Navig.*, vol. 151, no. 6, 2004, pp. 351–357.
- [153] X. Deng, Y. Pi, M. R. Morelande, and B. Moran, “Track-before-detect procedures for low pulse repetition frequency surveillance radars,” *Proc. IET Radar, Sonar & Navig.*, vol. 5, no. 1, pp. 65–73, 2011.
- [154] G. Li, R. Ebelt, and M. Vossiek, “Particle filter based synthetic aperture reconstruction approach for real-time 3D wireless local positioning,” in *Proc. IEEE Microwave*, 2012, pp. 1–4.
- [155] A. M. Johansson, E. A. Lehmann, and S. Nordholm, “Real-time implementation of a particle filter with integrated voice activity detector for acoustic speaker tracking,” in *Proc. IEEE Circuits and Systems*, 2006, pp. 1004–1007.
- [156] C. Andrieu and S. J. Godsill, “A particle filter for model based audio source separation,” in *Proc. Workshop Independent Component Analysis*, 2000.

- [157] E. Baziw, “Real-time seismic signal enhancement utilizing a hybrid Rao-Blackwellized particle filter and hidden Markov model filter,” *IEEE Geoscience and Remote Sensing Lett.*, vol. 2, no. 4, pp. 418–422, 2005.
- [158] A. Benavoli, L. Chisci, and A. Farina, “Tracking of a ballistic missile with a-priori information,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 43, no. 3, pp. 1000–1016, 2007.
- [159] Y. Zhou, E. Yeniaras, P. Tsiamyrtzis, N. Tsekos, and I. Pavlidis, “Collaborative tracking for MRI-guided robotic intervention on the beating heart,” in *J. Springer Medical Image Computing and Computer-Assisted Intervention*, 2010, pp. 351–358.
- [160] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, “Particle filters for positioning, navigation, and tracking,” *IEEE Tran. Signal Process.*, vol. 50, no. 2, pp. 425–437, 2002.
- [161] H. Sidenbladh, “Multi-target particle filtering for the probability hypothesis density,” in *Proc. Inf. Fusion*, vol. 2, 2003, pp. 800–806.
- [162] T. Zajic and R. P. S. Mahler, “Particle-systems implementation of the PHD multitarget-tracking filter,” in *Proc. SPIE*, vol. 5096, 2003, pp. 291–299.
- [163] B. N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo implementation of the PHD filter for multi-target tracking,” in *Proc. Inf. Fusion*, 2003, pp. 792–799.
- [164] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro, “Particle PHD filtering for multi-target visual tracking,” in *Proc. IEEE Acoust., Speech and Signal Process.*, vol. 1, 2007, pp. I–1101.

- [165] N. Whiteley, S. Singh, and S. J. Godsill, “Auxiliary particle implementation of probability hypothesis density filter,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 46, no. 3, pp. 1437–1454, 2010.
- [166] E. Baser and M. Efe, “A novel auxiliary particle PHD filter,” in *Proc. Inf. Fusion*, 2012, pp. 165–172.
- [167] A. M. Johansen, S. Singh, A. Doucet, and B. N. Vo, “Convergence of the SMC implementation of the PHD filter,” *J. Springer Methodology and Computing in Applied Probability*, vol. 8, no. 2, pp. 265–291, 2006.
- [168] D. E. Clark and J. Bell, “Convergence results for the particle PHD filter,” *IEEE Trans. Signal Process.*, vol. 54, no. 7, pp. 2652–2661, 2006.
- [169] ——, “Data association for the PHD filter,” in *Proc. IEEE Int. Conf. Intelligent Sensors, Sensor Networks and Inf. Process.*, 2005, pp. 217–222.
- [170] L. Zhao, P. Ma, X. Su, and H. Zhang, “A new multi-target state estimation algorithm for PHD particle filter,” in *Proc. Inf. F*, 2010, pp. 1–8.
- [171] B. Ristic, D. E. Clark, and B. N. Vo, “Improved SMC implementation of the PHD filter,” in *Proc. Inf. F*, 2010, pp. 1–8.
- [172] L. Lin, Y. Bar-Shalom, and T. Kirubarajan, “Track labeling and PHD filter for multitarget tracking,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 42, no. 3, pp. 778–795, 2006.
- [173] K. Panta, B. N. Vo, and S. Singh, “Novel data association schemes for the probability hypothesis density filter,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 43, no. 2, pp. 556–570, 2007.

- [174] D. E. Clark and J. Bell, “Multi-target state estimation and track continuity for the particle PHD filter,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 43, no. 4, pp. 1441–1453, 2007.
- [175] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Royal Statistic. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.
- [176] S. Hong, L. Wang, Z. G. Shi, and K. S. Chen, “Simplified Particle PHD Filter for Multiple-Target Tracking: Algorithm and Architecture,” *Progress in Electromagnetics Research*, vol. 120, pp. 481–498, 2011.
- [177] Z. Shi, Y. Liu, S. Hong, J. Chen, and X. Shen, “POSE: Design of hardware friendly particle-based observation selection PHD filter,” *Industrial Electronics, IEEE Transactions on*, vol. 61, no. 4, pp. 1944–1956, 2014.
- [178] J. H. Yoon, D. Y. Kim, and K. J. Yoon, “Efficient importance sampling function design for sequential Monte Carlo PHD filter,” *J. Elsevier Signal Process.*, vol. 92, no. 9, pp. 2315–2321, 2012.
- [179] B. N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo methods for multitarget filtering with random finite sets,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, 2005.
- [180] B. Ristic, D. E. Clark, and B. N. Vo, “Improved SMC implementation of the PHD filter,” in *Proc. Inf. Fusion*, 2010, pp. 1–8.
- [181] B. N. Vo and W. K. Ma, “A closed-form solution for the probability hypothesis density filter,” in *Proc. Inf. Fusion*, vol. 2, 2005, pp. 8–pp.
- [182] ——, “The Gaussian mixture probability hypothesis density filter,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, 2006.

BIBLIOGRAPHY

- [183] D. E. Clark, K. Panta, and B. N. Vo, “The GM-PHD filter multiple target tracker,” in *Proc. Inf. Fusion*, 2006, pp. 1–8.
- [184] D. E. Clark and B. N. Vo, “Convergence analysis of the gaussian mixture PHD filter,” *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1204–1212, 2007.
- [185] K. Granström, C. Lundquist, and O. Orguner, “Extended target tracking using a Gaussian-mixture PHD filter,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 48, no. 4, pp. 3268–3286, 2012.
- [186] D. E. Clark and J. Bell, “GM-PHD filter multi-target tracking in sonar images,” in *Proc. SPIE Symp. Defense and Security*, 2006.
- [187] R. P. Mahler, “PHD filters of higher order in target number,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, 2007.
- [188] M. Ulmke, O. Erdinc, and P. Willett, “Gaussian mixture cardinalized PHD filter for ground moving target tracking,” in *Proc. Inf. Fusion*, 2007, pp. 1–8.
- [189] D. E. Clark and J. Bell, “Bayesian multiple target tracking in forward scan sonar images using the PHD filter,” in *Proc. IEE Radar Sonar Navig.*, vol. 152, no. 5, 2005, pp. 327–334.
- [190] D. E. Clark, J. Bell, Y. Saint Pern, and Y. Petillot, “PHD filter multi-target tracking in 3D sonar,” in *Proc. IEEE Oceans*, vol. 1, 2005, pp. 265–270.
- [191] B. Balakumar, A. Sinha, T. Kirubarajan, and J. P. Reilly, “PHD filtering for tracking an unknown number of sources using an array of sensors,” in *IEEE Workshop Statistical Signal Process.*, 2005, pp. 43–48.

- [192] E. Özkan, M. B. Guldogan, U. Orguner, and F. Gustafsson, “Ground multiple target tracking with a network of acoustic sensor arrays using PHD and CPHD filters,” in *Proc. Inf. Fusion*, 2011, pp. 1–8.
- [193] A. Masnadi-Shirazi and B. D. Rao, “An ICA-SCT-PHD filter approach for tracking and separation of unknown time-varying number of sources,” *IEEE Trans. Audio, Speech, and Language Process.*, vol. 21, no. 4, pp. 828–841, 2013.
- [194] J. D. Glass and A. D. Lanterman, “MIMO radar target tracking using the probability hypothesis density filter,” in *Proc. IEEE Aerosp.*, 2012, pp. 1–8.
- [195] M. Canaud, L. Mihaylova, J. Sau, and N. E. Faouzi, “Probability hypothesis density filtering for real-time traffic state estimation and prediction,” *J. AIMS Networks & Heterogeneous Media*, vol. 8, no. 3, 2013.
- [196] B. N. Vo, B. T. Vo, N. T. Pham, and D. Suter, “Joint detection and estimation of multiple objects from image observations,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5129–5141, 2010.
- [197] R. Hoseinnezhad, B. N. Vo, B. Suter, and B. T. Vo, “Multi-object filtering from image sequence without detection,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 1154–1157.
- [198] B. Ristic, B. T. Vo, B. N. Vo, and A. Farina, “A tutorial on Bernoulli filters: theory, implementation and applications,” *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3406–3430, 2013.
- [199] F. Lian, C. Li, C. Han, and H. Chen, “Convergence analysis for the SMC-MeMBer and SMC-CBMeMBer filters,” *J. Applied Mathematics*, vol. 2012, 2012.
- [200] J. Yin, J. Zhang, and J. Zhao, “The Gaussian particle multi-target multi-Bernoulli filter,” in *Proc. IEEE Adv. Computer Control*, vol. 4, 2010, pp. 556–560.

- [201] J. H. Kotecha and P. M. Djurić, “Gaussian particle filtering,” *IEEE Tran. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, 2003.
- [202] ——, “Gaussian sum particle filtering,” *IEEE Tran. Signal Process.*, vol. 51, no. 10, pp. 2602–2612, 2003.
- [203] D. E. Clark, B. T. Vo, and B. N. Vo, “Gaussian particle implementations of probability hypothesis density filters,” in *Proc. IEEE Aerosp.*, 2007, pp. 1–11.
- [204] L. Chen, M. J. Wainwright, M. Cetin, and A. S. Willsky, “Multitarget-multisensor data association using the tree-reweighted max-product algorithm,” in *Proc. Int. Soc. Optics and Photonics, AeroSense*, 2003, pp. 127–138.
- [205] J. L. Williams, “Experiments with graphical model implementations of multiple target multiple Bernoulli filters,” in *Proc. IEEE Intell. Sensors, Sensor Netw. and Inf. Process.* IEEE, 2011, pp. 532–537.
- [206] ——, “Hybrid Poisson and multi-Bernoulli filters,” in *Proc. Inf. Fusion*, 2012, pp. 1103–1110.
- [207] C. Ouyang, H. Ji, and C. Li, “Improved multi-target multi-Bernoulli filter,” *Proc. IET Radar, Sonar & Navig.*, vol. 6, no. 6, pp. 458–464, 2012.
- [208] B. T. Vo, B. N. Vo, R. Hoseinnezhad, and R. P. S. Mahler, “Multi-Bernoulli filtering with unknown clutter intensity and sensor field-of-view,” in *Proc. IEEE Inf. Sciences and Systems*, 2011, pp. 1–6.
- [209] B. T. Vo, R. Hoseinnezhad, and R. P. S. Mahler, “Robust multi-Bernoulli filtering,” *J IEEE Selected Topics in Signal Process.*, vol. 7, no. 3, pp. 399–409, 2013.

- [210] A. K. Gostar, R. Hoseinnezhad, A. Bab-Hadiashar, and B. T. Vo, “Control of sensor with unknown clutter and detection profile using multi-Bernoulli filter,” in *Proc. Inf. Fusion*, 2013, pp. 1021–1028.
- [211] R. Hoseinnezhad, B. N. Vo, and T. N. Vu, “Visual tracking of multiple targets by multi-Bernoulli filtering of background subtracted image data,” in *J. Springer Adv. in Swarm Intelligence*, 2011, pp. 509–518.
- [212] R. Hoseinnezhad, B. N. Vo, B. T. Vo, and D. Suter, “Visual tracking of numerous targets via multi-Bernoulli filtering of image data,” *J. Elsevier Pattern Recogn.*, vol. 45, no. 10, pp. 3625–3635, 2012.
- [213] J. Wong, B. T. Vo, B. N. Vo, and R. Hoseinnezhad, “Multi-Bernoulli based track-before-detect with road constraints,” in *Proc. Inf. Fusion*, 2012, pp. 840–846.
- [214] R. Hoseinnezhad, B. N. Vo, B. T. Vo, and D. Suter, “Bayesian integration of audio and visual information for multi-target tracking using a CB-MeMBer filter,” in *Proc. IEEE Acoust. Speech and Signal Process.*, 2011, pp. 2300–2303.
- [215] L. Geng, M. F. Bugallo, and P. M. Djurić, “Tracking with RFID asynchronous measurements by particle filtering,” vol. 13, pp. 4051–4055, 2013.
- [216] C. Idler, R. Schweiger, D. Paulus, M. Mahlisch, and W. Ritter, “Real time vision based multi-target-tracking with particle filters in automotive applications,” in *Proc. IEEE Intelligent Vehicles*, 2006, pp. 188–193.
- [217] C. Kreucher and B. Shapo, “Multitarget detection and tracking using multisensor passive acoustic data,” *IEEE J. Ocean. Eng.*, vol. 36, no. 2, pp. 205–218, 2011.

BIBLIOGRAPHY

- [218] W. Ng, J. Li, S. J. Godsill, and J. Vermaak, “A hybrid method for online tracking of a variable number of targets,” *IEEE Trans. Proc. Aerosp. Electron. Syst.*, vol. 1, no. 1, pp. 1–22, 2006.
- [219] J. Vermaak, S. Maskell, M. Briers, and P. Pérez, “Bayesian visual tracking with existence process,” in *Proc. IEEE Image Process.*, vol. 1.
- [220] M. Fallon, “Multi target acoustic source tracking with an unknown and time varying number of targets,” in *HSCMA Proc. Hands-Free Speech Communication and Microphone Arrays*, 2008, pp. 77–80.
- [221] M. Fallon and S. J. Godsill, “Acoustic Source Localization and Tracking of a Time-Varying Number of Speakers,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 4, pp. 1409–1415, 2012.
- [222] W. Ng, J. Li, S. J. Godsill, and S. K. Pang, “Multitarget initiation, tracking and termination using Bayesian Monte Carlo methods,” *The Computer J., British Computer Soc.*, vol. 50, no. 6, pp. 674–693, 2007.
- [223] M. Ekman, “Particle filters and data association for multi-target tracking,” in *Proc. IEEE Information Fusion*, 2008, pp. 1–8.
- [224] G. Casella and E. George, “Explaining the Gibbs sampler,” *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.
- [225] C. Hue, J. P. L. Cadre, and P. Pérez, “Tracking multiple objects with particle filtering,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 791–812, 2002.
- [226] S. Oh, S. Russell, and S. Sastry, “Markov chain monte carlo data association for multi-target tracking,” *IEEE Trans. Automatic Control*, vol. 54, no. 3, pp. 481–497, 2009.
- [227] T. Yang, G. Huang, and P. G. Mehta, “Joint probabilistic data association-feedback particle filter for multiple target tracking applications,” in *IEEE, American Control Conf.*, 2012, pp. 820–826.

- [228] P. Chavali and A. Nehorai, “Concurrent Particle filtering and Data association using Game theory for tracking multiple maneuvering targets,” *IEEE Trans. Signal Process.*, vol. 61, no. 20, pp. 4934–4948, 2013.
- [229] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [230] Y. Ruan, P. Willett, and R. Streit, “The PMHT for maneuvering targets,” in *Proc. IEEE Am. Control Conference*, vol. 4, 1998, pp. 2432–2433.
- [231] P. Willett, Y. Ruan, and R. Streit, “PMHT: problems and some solutions,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 38, no. 3, pp. 738–754, 2002.
- [232] M. R. Morelande and D. Musicki, “Fast multiple target tracking using particle filters,” in *Proc. IEEE Decision and Control*, 2005, pp. 530–535.
- [233] B. T. Vo, B. N. Vo, and A. Cantoni, “Bayesian filtering with random finite set observations,” *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1313–1326, 2008.
- [234] A. L. Swindlehurst and P. Stoica, “Maximum likelihood methods in radar array signal processing,” *Proc. IEEE*, vol. 86, no. 2, pp. 421–441, 1998.
- [235] S. Haykin, “Radar signal processing,” *IEEE Acoustic, Speech and Signal Proc. Mag.*, vol. 2, no. 2, pp. 2–18, 1985.
- [236] R. J. Vaccaro, “The past, present, and the future of underwater acoustic signal processing,” *IEEE Signal Proc. Mag.*, vol. 15, no. 4, pp. 21–51, 1998.

BIBLIOGRAPHY

- [237] J. Liu, J. Reich, and F. Zhao, “Collaborative in-network processing for target tracking,” *EURASIP J. Applied Signal Proc.*, vol. 2003, pp. 378–391, 2003.
- [238] Y.-D. Huang and M. Barkat, “Near-field multiple source localization by passive sensor array,” *IEEE Trans. Antennas and Propagation*, vol. 39, no. 7, pp. 968–975, 1991.
- [239] J. P. Dmochowski, J. Benesty, and S. Affes, “Broadband MUSIC: opportunities and challenges for multiple source localization,” in *Proc. IEEE Workshop on Applications of Signal Process. Audio and Acoust.*, 2007, pp. 18–21.
- [240] J. Böhme, “Estimation of source parameters by maximum likelihood and nonlinear regression,” in *Proc. IEEE Acoustics, Speech, and Signal Proc.*, vol. 9, 1984, pp. 271–274.
- [241] B. Efron and R. J. Tibshirani, “An introduction to the bootstrap,” *Chapman and Hall*, 1994.
- [242] G. Kitagawa, “Monte Carlo filter and smoother for non-Gaussian non-linear state space models,” *J. of Computational and Graphical Statistics*, pp. 1–25, 1996.
- [243] P. Stoica and R. L. Moses, *Spectral analysis of signals*. Pearson/Prentice Hall, 2005.
- [244] H. Sun, E. Mabande, K. Kowalczyk, and W. Kellermann, “Localization of distinct reflections in rooms using spherical microphone array eigenbeam processing,” *J. Acoustical Soc. of America*, vol. 131, no. 4, pp. 2828–2840, 2012.
- [245] N. Shahapurkar and C. S. Ramalingam, “Threshold performance of MUSIC when using the forward-backward data matrix,” *IEEE Signal Process. Lett.*, vol. 13, no. 2, pp. 80–83, 2006.

- [246] D. B. Rao and K. Hari, “Performance analysis of root-MUSIC,” *IEEE Trans. Acoust., Speech and Signal Process.*, vol. 37, no. 12, pp. 1939–1949, 1989.
- [247] L. Blanco and M. Nájar, “Sparse covariance fitting for direction of arrival estimation,” *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, pp. 1–11, 2012.
- [248] Y. Abramovich and N. K. Spencer, “Locally optimal maximum-likelihood completion of a partially specified toeplitz covariance matrix,” in *Proc. IEEE Workshop Statistical Signal Process.*, 2001, pp. 229–232.
- [249] S. Krutchman and B. Nadler, “Non-parametric detection of the number of signals: hypothesis testing and random matrix theory,” *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3930–3941, 2009.
- [250] E. Yazdian, S. Gazor, and M. H. Bastani, “Limiting spectral distribution of the sample covariance matrix of the windowed array data,” *J. EURASIP Adv. Signal Process.*, vol. 2013, no. 1, pp. 1–15, 2013.
- [251] J. Wang, Z. Huang, and Y. Zhou, “Direction-of-arrival estimation based on joint sparsity,” *J. Sensors, Molecular Diversity Preservation Int'l*, vol. 11, no. 9, pp. 9098–9108, 2011.
- [252] N. Vaswani, “Particle filtering for large-dimensional state spaces with multimodal observation likelihoods,” *IEEE Trans. on Signal Process.*, vol. 56, no. 10, pp. 4583–4597, 2008.
- [253] D. B. Ward and R. C. Williamson, “Particle filter beamforming for acoustic source localization in a reverberant environment,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, 2002, pp. 1777–1780.

BIBLIOGRAPHY

- [254] C. Andrieu, A. Doucet, and V. B. Tadic, “On-line parameter estimation in general state-space models,” in *Proc. IEEE Decision and Control*, 2005, pp. 332–337.
- [255] M. Kawamoto, F. Asano, H. Asoh, and K. Yamamoto, “Particle Filtering Algorithms for Tracking Multiple Sound sources using Microphone Arrays,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, 2007, pp. I-129–I-132.
- [256] X. Zhong, A. B. Premkumar, and A. S. Madhukumar, “Particle filtering for acoustic source tracking in Impulsive noise with Alpha-stable process,” *IEEE J. Sensors*, vol. 13, no. 2, pp. 589–600, 2013.
- [257] I. Marković, A. Portello, P. Danes, I. Petrović, and S. Argentieri, “Active speaker localization with circular likelihoods and bootstrap filtering,” in *Proc. IEEE Intell. Robots and Syst.*, 2013, pp. 2914–2920.
- [258] L. Huang, T. Long, E. Mao, and H. C. So, “MMSE-based MDL method for accurate source number estimation,” *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 798–801, 2009.
- [259] L. Huang and S. Wu, “Low-complexity MDL method for accurate source enumeration,” *IEEE Signal Process. Lett.*, vol. 14, no. 9, pp. 581–584, 2007.
- [260] H. Akaike, “A new look at the statistical model identification,” *IEEE Trans. Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.
- [261] B. M. Radich and K. M. Buckley, “The effect of source number underestimation on MUSIC location estimates,” *IEEE Trans. Signal Process.*, vol. 42, no. 1, pp. 233–236, 1994.
- [262] F. Aurenhammer, “Voronoi diagrams — A survey of a fundamental geometric data structure,” *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

- [263] J. C. Dunn, “A fuzzy relative of the ISO DATA process and its use in detecting compact well-separated clusters,” *J. Cybernetics*, pp. 32–57, 1973.
- [264] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Royal Statistical Society*, pp. 1–38, 1977.
- [265] S. C. Johnson, “Hierarchical clustering schemes,” *Springer Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
- [266] H. Zhu, C. Han, and Y. Lin, “Particle labeling PHD filter for multi-target track-valued estimates,” in *Proc. Inf. Fusion*, 2011, pp. 1–8.
- [267] D. Franken, M. Schmidt, and M. Ulmke, ““Spooky action at a distance” in the cardinalized probability hypothesis density filter,” *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 45, no. 4, pp. 1657–1664, 2009.
- [268] F. Thouin, S. Nannuru, and M. Coates, “Multi-target tracking for measurement models with additive contributions,” in *Proc. Inf. Fusion*, 2011, pp. 1–8.
- [269] S. Nannuru and M. Coates, “Multi-Bernoulli filter for superpositional sensors,” in *Proc. Inf. Fusion*, 2013, pp. 1632–1637.
- [270] P. B. Choppala, P. D. Teal, and M. R. Frean, “Adapting the multi-Bernoulli filter to phased array observations using MUSIC as pseudo-likelihood,” in *Proc. Inf. Fusion*, 2014.
- [271] D. B. Ward, E. A. Lehmann, and R. C. Williamson, “Particle filtering algorithms for tracking an acoustic source in a reverberant environment,” *IEEE Trans. Speech and Audio Process.*, vol. 11, no. 6, pp. 826–836, 2003.

- [272] T. Zajic and R. Mahler, “Particle-systems implementation of the PHD multitarget-tracking filter,” in *Proc. SPIE*, vol. 5096, 2003, pp. 291–299.
- [273] N. Smirnov, “Table for estimating the goodness of fit of empirical distributions,” *J. Annals of Mathematical Statistics*, pp. 279–281, 1948.
- [274] G. Fasano and A. Franceschini, “A multidimensional version of the Kolmogorov-Smirnov test,” *Royal Astronomical Society*, vol. 225, pp. 155–170, 1987.
- [275] A. Justel, D. Peña, and R. Zamar, “A multivariate Kolmogorov-Smirnov test of goodness of fit,” *Elsevier Statistics & Probability Letters*, vol. 35, no. 3, pp. 251–259, 1997.
- [276] R. H. C. Lopes, I. Reid, and P. R. Hobson, “The two-dimensional Kolmogorov-Smirnov test,” in *XI Int. Workshop on Adv. Comput. Analysis Techniques in Physics Research*, 2007.
- [277] P. M. Djurić and J. Míguez, “Assessment of Nonlinear Dynamic Models by Kolmogorov-Smirnov Statistics,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5069–5079, 2010.
- [278] P. B. Choppala, P. D. Teal, and M. R. Frean, “Soft systematic resampling for accurate posterior approximation and increased information retention in particle filtering,” in *Workshop IEEE Statistic. Signal Process.*, 2014.
- [279] J. A. Koziol and D. P. Byar, “Percentage points of the asymptotic distributions of one and two sample KS statistics for truncated or censored data,” *Technometrics*, vol. 17, no. 4, pp. 507–510, 1975.
- [280] P. Closas, M. F. Bugallo, and P. M. Djurić, “Assessing robustness of particle filtering by the Kolmogorov-Smirnov statistics,” in *Proc. IEEE Acoust., Speech and Signal Process.*, 2009, pp. 2917–2920.

BIBLIOGRAPHY

- [281] J. Lin, “Divergence measures based on the shannon entropy,” *IEEE Trans Inf. Theory*, vol. 37, no. 1, pp. 145–151, 1991.
- [282] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough, “A joint particle filter for audio-visual speaker tracking,” in *Proc. ACM Multimodal Interfaces*, 2005, pp. 61–68.
- [283] S. C. S. C. Johnson, “Hierarchical clustering schemes,” *J. Springer Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
- [284] W. E. Wright, “Gravitational clustering,” *J. Elsevier Pattern Recognition*, vol. 9, no. 3, pp. 151–166, 1977.
- [285] C. Masreliez and R. Martin, “Robust Bayesian estimation for the linear model and robustifying the Kalman filter,” *IEEE Trans. Automatic Control*, vol. 22, no. 3, pp. 361–371, 1977.
- [286] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, “Robust estimation in signal processing: (a) tutorial-style treatment of fundamental concepts,” *IEEE Signal Process. Mag.*, vol. 29, no. 4, pp. 61–80, 2012.
- [287] R. Couillet, F. Pascal, and J. W. Silverstein, “A joint robust estimation and random matrix framework with application to array processing,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 13, 2013, pp. 6561–6565.
- [288] R. Roy and T. Kailath, “ESPRIT - Estimation of signal parameters via rotational invariance techniques,” *IEEE Trans. Acoust., Speech and Signal Process.*, vol. 37, no. 7, pp. 984–995, 1989.
- [289] R. O. Adeogun, P. D. Teal, and P. A. Dmochowski, “Parametric channel prediction for narrowband mobile MIMO systems using spatio-temporal correlation analysis,” in *Proc. IEEE Vehicular Tech.*, 2013, pp. 1–5.