Track-Before-Detect for Active Sonar

Han Xuan Vu

Thesis submitted for the degree of
Doctor of Philosophy
in
Electrical and Electronic Engineering
at
The University of Adelaide

School of Electrical and Electronic Engineering

THE UNIVERSITY
of ADELAIDE

February 3, 2015

Abstract

The detection and tracking of underwater targets with active sonar is a challenging problem because of high acoustic clutter, fluctuating target returns and a relatively low measurement update rate. In this thesis, a Bayesian framework for the detection and tracking of underwater targets using active sonar is formulated. In general, Bayesian tracking algorithms are built on two statistical models: the target dynamics model and the measurement model. The target dynamics model describes the evolution of the target state with time and is almost always assumed to be a Markov process. The typical measurement model approximates the sensor image with a collection of discrete points at each frame and allows point measurement tracking to be performed. This thesis investigates alternative target and measurement models and considers their application to active sonar tracking.

The Markov process commonly used for target modelling assumes that the state evolves without knowledge of its future destination. Random realisations of a Markov process can also display a large amount of variability and do not, in general, resemble realistic target trajectories. An alternative is the reciprocal process, which assumes conditioning on a known destination state. The first key contribution is the derivation and implementation of a Maximum Likelihood Sequence Estimator (MLSE) for a Hidden Reciprocal Process (HRP). The performance of the proposed algorithm is demonstrated in simulated scenarios and shown to give improved state estimation performance over Markov processes for scenarios featuring reciprocal targets.

In point measurement tracking, reducing the sensor data to point detections results in the loss of valuable information. This method is generally sufficient for tracking high Signal-to-Noise Ratio (SNR) targets but can fail in the case of low SNR targets. The alternative to point measurement tracking is to provide the sensor intensity map, an image, as an input into the tracker. This paradigm is referred to as Track-Before-Detect (TkBD). This thesis will focus on a particular TkBD algorithm based on Expectation-Maximisation (EM) data association called the Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT) as it handles multiple targets with low complexity. In the second key contribution, we demonstrate a Viterbi implementation of the H-PMHT algorithm, and show that it outperforms the Kalman Filter in the linear non-

Gaussian case.

A problem with H-PMHT is that it fails to model fluctuating target amplitude, which can degrade performance in realistic sensing conditions. The third key contribution addresses this by replacing the multinomial measurement model with a Poisson mixture process. The new Poisson mixture is shown to be consistent with the original H-PMHT modelling assumptions but it now allows for a randomly evolving mean target amplitude state with instantaneous fluctuations. This new TkBD algorithm is referred to as the Poisson H-PMHT. The Bayesian prior on the target state is also modified to ensure more robust performance.

The fourth contribution is a novel TkBD algorithm based on the application of EM data association to a new measurement model that directly describes continuous valued intensity maps and avoids using an intermediate quantisation stage like the H-PMHT. This model is referred to as the Interpolated Poisson measurement model and is integrated into the Probabilistic Multi-Hypothesis Tracker (PMHT) framework to derive a TkBD algorithm for continuous data called the Interpolated Poisson-PMHT (IP-PMHT). The performance of the Poisson H-PMHT and IP-PMHT algorithms are verified through simulations and are shown to outperform the standard H-PMHT in terms of SNR estimation, particularly for scenarios featuring targets with highly fluctuating amplitude.

The final key contribution is the application of several TkBD algorithms based on EM data association to the active sonar problem through a comparative study using trial data from an active towed array sonar. The TkBD algorithms are modified to incorporate changes in target appearance with received array bearing, and are shown to give improved SNR and state estimation performance compared with a conventional point measurement tracking algorithm. The thesis concludes by discussing the limitations of the proposed algorithms and possible avenues for future work.

Publications

- H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.C. Lim. A new state prior for the Histogram-PMHT. *IEEE Signal Processing Letters, in preparation*
- H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.C. Lim. Histogram-PMHT with an evolving Poisson prior. In *2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2015
- H. X. Vu, S. J. Davey, F. K. Fletcher, S. Arulampalam, R. Ellem, and C.-C. Lim. Track-before-detect for an active towed array sonar. In *Proceedings of Acoustics 2013*, November 2013
- H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.-C. Lim. H-PMHT with a Poisson measurement model. In *Proceedings of the 2013 International Conference on Radar (Radar)*, pages 446–451, September 2013
- H. X. Vu and S. J. Davey. Track-before-detect using Histogram PMHT and dynamic programming. In *Proceedings of the 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–8, December 2012
- S. J. Davey, H. X. Vu, S. Arulampalam, F. Fletcher, and C.C. Lim. Clutter mapping for Histogram PMHT. In *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, pages 153–156, Gold Coast, Australia, June-July 2014
- S. J. Davey, M. Wieneke, and H. Vu. Histogram-PMHT unfettered. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):435–447, June 2013
- L. B. White and H. X. Vu. Maximum likelihood sequence estimation for hidden reciprocal processes. *IEEE Transactions on Automatic Control*, 58(10):2670–2674, October 2013

Declaration

I, Han Xuan Vu, certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signature..... *Date*.....

Acknowledgements

I would like to acknowledge my phenomenal team of supervisors, Sam Davey, Sanjeev Arulampalam, Fiona Fletcher and Cheng-Chew Lim for their ongoing support and encouragement throughout my candidature. I would also like to express my gratitude for their generousity and willingness to share their valuable time and expertise. Their mentorship, guidance and knowledge have been invaluable. Thank you also to Richard Ellem for the provision of trial data used in this research.

I would like to thank my host institutions the School of Electrical and Electronic Engineering at the University of Adelaide and the Maritime Division of the Defence Science and Technology Organisation for their continued support throughout my candidature.

Finally, a sincere thank you to all my friends and family for their unwavering support and patience through the ups and downs of my candidature. A special mention to my partner Chris for taking the brunt of it, without (too much) complaint.

Contents

List of Figures

List of Tables

List of Acronyms

EM	Expectation-Maximisation
EKF	Extended Kalman Filter
ER	Echo-Repeater
GNN	Global Nearest Neighbour
HMM	Hidden Markov Model
HMC	Hidden Markov Chain
HRC	Hidden Reciprocal Chain
HRP	Hidden Reciprocal Process
H-PMHT	Histogram-Probabilistic Multi-Hypothesis Tracker
iid	independent and identically distributed
IPDA	Integrated Probabilistic Data Association
IP-PMHT	Interpolated Poisson - Probabilistic Multi-Hypothesis Tracker
MAP	Maximum a Posteriori
MLSE	Maximum Likelihood Sequence Estimator
MMSE	Minimum Mean Square Error
ML	Maximum Likelihood
NN	Nearest Neighbour
KF	Kalman Filter
pdf	probability density function

psf	point spread function
PDA	Probabilistic Data Association
PF	Particle Filter
PMHT	Probabilistic Multi-Hypothesis Tracker
rhs	right hand side
RMS	Root Mean Square
RC	Reciprocal Chain
RP	Reciprocal Process
SMC	Sequential Monte Carlo
SNR	Signal-to-Noise Ratio
TkBD	Track-Before-Detect
UKF	Unscented Kalman Filter

List of Principal Symbols

$!$	Factorial operator
$[\cdot]^T$	Transpose operator
$[\cdot]'$	Indicates that the variable is dependent on estimates from the previous EM iteration
η	Forgetting factor
λ	Poisson mixing rate
λ_t^m	Poisson mixing rate for component m at time t
$\widetilde{\lambda}$	Quantised Poisson mixing rate
$\widetilde{\lambda}_t$	Quantised Poisson mixing rate at time t
$\widetilde{\lambda}_t^m$	Quantised Poisson mixing rate for component m at time t
$\widetilde{\lambda}_t^{im}$	Quantised Poisson mixing rate for component m in pixel i at time t
$\mathbf{\Lambda}$	Collection of Poisson mixing rate terms for all times and components
$\mathbf{\Lambda}'$	Collection of Poisson mixing rate terms at the previous EM iteration for all time and components
π	Multinomial mixing proportion term
π_t^m	Multinomial mixing proportion term for component m in pixel i at time t
Π	Collection of multinomial mixing proportion terms for all times and components
Π'	Collection of multinomial mixing proportion terms at the previous EM iteration for all times and components
μ_t^{im}	Proportion of power from component m in pixel i at time t
$\mathfrak{Z}_t^{im}$	Energy from component m in pixel i at time t
$B_{i,j}^k$	Markov bridge transitions

c^2	Quantisation level
f_t	H-PMHT probability density function at time t
f_t^i	H-PMHT per-pixel probability density at time t
$\mathbb{f}_t$	Poisson H-PMHT intensity function at time t
$\mathbb{f}_t^i$	Poisson H-PMHT per-pixel intensity at time t
F	Linear target transition matrix
F_t	Estimated total power in the image at time t as calculated by the H-PMHT
$F_t^{\mathcal{O}}$	Estimated total power in the observed image pixels at time t as calculated by the H-PMHT
$\mathbb{F}_t$	Estimated total power in the observed image pixels at time t as calculated by the Poisson H-PMHT
$G_0(\cdot)$	Clutter distribution
$h(\cdot)$	Point spread function
$h^i\left(\boldsymbol{x}_t^m\right)$	Probability of a shot due to a target $\boldsymbol{x}_t^m$ falls in pixel i
$h^i(\emptyset)$	Probability of a shot due to clutter falls in pixel i
H	Linear measurement function
i	Measurement pixel index
I	Total number observed pixels
K_t^i	Set of components associated with shots in pixel i at time t
K_t^{ir}	Index of the component associated with the r^{th} shot in pixel i at time t
$\mathbf{K}$	Collection of the assignments of measurements to components for all times and components
$\mathbf{L}$	Collection of the precise locations of the shots inside its pixel for all time and components
$\mathbb{m}_t$	Number of point detections at time t
m	Component index
M	Number of components
$\mathbf{n}_t^i$	Quantised measurements (counts) in pixel i at time t
$\mathbf{n}_t^{im}$	Quantised measurements (counts) from component m in pixel i at time t

$\mathbf{n}_t^{ci}$	Quantised unobserved measurements (counts) in pixel i at time t
$\mathbf{N}$	Collection of quantised observed measurements for all times and components
$\mathbf{N}^c$	Collection of quantised unobserved measurements for all times and components
$\|\|\mathbf{N}_t\|\|$	Total number of shots in the observed image at time t
$\|\|\mathbf{N}_t^c\|\|$	Total number of shots in the unobserved image at time t
$\|\|\mathbf{N}_t^{total}\|\|$	Total number of shots in image (from both observed and unobserved pixels) at time t
$\mathbf{O}$	Observer
Q	Process noise covariance
$Q^{(H)}$	H-PMHT auxiliary function
$Q^{(P)}$	Poisson H-PMHT auxiliary function
$Q^{(IP)}$	IP-PMHT auxiliary function
Q_{ijl}	Reciprocal three point transition function
r	Measurement shot index
R	Measurement covariance matrix
S	Total number of image pixels (unobserved and observed)
t	Time index
T	Number of time scans
$\boldsymbol{x}_t^m$	State of component m at time t
X_t	State in Markov Model
$\mathbf{X}$	Collection of components for all times and components
$\mathbf{X}'$	Collection of components at the previous EM iteration for all times and components
Y_t	Observations in the Markov model at time t
$\boldsymbol{z}_t$	An observed measurement at time t
$\boldsymbol{z}_t^i$	Observed energy in pixel i at time t
$\boldsymbol{z}_t^m$	Observed energy from component m at time t
$\mathbf{Z}$	Collection of observed measurements for all times and components
$\mathbf{Z}_t$	Observed measurements at time t
Z_t	Collection of observed measurements up until time t

$\mathbf{Z}^c$	Collection of unobserved measurements for all times
$\mathbf{Z}_t^c$	Collection of unobserved measurements up to time t
$\|\|\mathbf{Z}_t\|\|$	Total observed energy in image at time t

Chapter 1

Introduction

The detection and tracking of underwater targets is a fundamental activity in identifying possible threats to naval operations and other maritime interests. It is typically conducted using passive or active sonar (**so**und **n**avigation **a**nd **r**anging). Active sonar sensors transmit acoustic signals and detect energy that is reflected back from objects in the environment while passive sensors emit no signals and perform detection by 'listening' to sounds waves emitted from the surrounding environment. In the past few decades, improvements in acoustic noise quietening techniques have reduced passive acoustic signature levels making it more difficult to detect underwater targets using passive sonar systems. This has lead to an increased use of active sonar systems for the detection and tracking of underwater targets. Despite this, active sonar tracking remains a challenging problem, particularly in littoral environments where performance can be degraded by high levels of acoustic clutter, fluctuating target returns and a relatively low measurement update rate.

In this thesis, a Bayesian framework for the detection and tracking of underwater targets using active sonar is formulated. In general, Bayesian tracking algorithms are built on two statistical models: the target dynamics model and the measurement model. The target dynamics model describes the evolution of the target state with time and is almost always assumed to be a Markov process. The typical measurement model approximates the sensor image with a collection of discrete points at each frame and point measurement tracking associates these points together over frames. This thesis investigates alternative target and measurement models based on Hidden Reciprocal Processes (HRPs) and Track-Before-Detect (TkBD), respectively, and considers their application to active sonar tracking.

The chapter is arranged as follows: Section 1.1 provides the motivation of this work as it relates to active sonar tracking; Section 1.2 discusses the unique issues associated with active sonar

tracking; Section 1.3 outlines the scope of the thesis and summarises the key contributions.

1.1 Motivation

In the past few decades, the emergence of advanced submarine capability from a number of Asia-Pacific nations has had serious implications for Australian naval operations and its other maritime interests. Due to advancements in technology, modern submarines are equipped with new propulsion systems that enable increasingly stealthy and sustained underwater operations. Combined with other factors such as a highly variable and complex underwater environment, the task of detecting and tracking of submarines using active sonar is a non-trivial problem. The Australian Government Department of Defence states in its 2013 Defence White paper [28] that it envisions "an Australian Defence Force more capable in: undersea warfare; anti-submarine warfare". In support of this strategic initiative, this thesis explores effective and robust solutions for the detection and tracking of underwater targets.

1.2 Active Sonar Tracking Problem

Active sonar systems operate by emitting a pulse of sound energy into the water and detecting energy that is reflected back from objects in the environment [103]. Given a signal return, the location of targets of interest can be easily determined: the distance to objects can be calculated using the difference in time between the initial transmitted pulse and the arrival of any reflected energy back into the system; bearing can be calculated using beamforming by measuring the phase difference of the return signal as it strikes the receiver array elements. However, the use of active sonar to detect and track targets is challenging due to the complex highly variable nature of the acoustic environment. The level of ocean ambient background noise can vary, and at times mask the echoes from targets of interest. Therefore, the Signal-to-Noise Ratio (SNR) of the expected target return must exceed the ambient noise in order for it to be detected.

Sound energy can also be scattered (reflected or diffused) by other sources in the environment such as marine life or by the inhomogeneous nature of the sea surface and sea floor [138]. A special case of scattering is reverberation, which is caused by sound energy being backscattered towards the sensor. In shallow littoral waters, reverberation can often be amplified due to increased signal interactions with the sea surface and sea floor. In this case, the reverberation level can swamp the target return, making target detection difficult. This situation is generally referred to as a reverberation-limited environment [138]. Unsurprisingly, high levels of background ambient noise and/or reverberation are generally undesirable and are the two

most dominant factors in active sonar performance.

The acoustic environment can be often characterised by a highly cluttered and fluctuating channel, analogous to the fading channel phenomena encountered in communications [12]. In general, the degradation of active sonar system performance can attributed to the following:

- high levels of acoustic clutter that result in false alarms,
- poor acoustic propagation resulting in low SNR or fading targets,
- complex time-varying acoustic conditions causing multi-path effects, and finally
- a relatively low data update rate.

Under these conditions, the detection and tracking of a single target is a non-trivial problem. These challenges become even more difficult to mitigate in the multi-target case. Current approaches in the active sonar tracking open literature are unable to identify one single technique that is able to adequately address the problems associated with active sonar.

The main objective in active sonar target tracking is to identify the number of targets and estimate their trajectories over time using a sequence of noisy measurements from an active sonar sensor. Generally, when analysing a dynamic system, two models are required: a target model that describes the target state evolution with time; and a measurement model that relates the noisy measurements to the state. These two models are usually given in probabilistic state space form and information relating to the system is recursively updated with the arrival of new measurements. This process lends itself naturally to Bayesian tracking, in which probabilities are updated whenever new information is received [8].

Under a Bayesian framework, the dynamics of an underwater target is generally assumed to be a Markov process with nearly-constant-velocity motion. However, a Markov process assumes that the state evolves without knowledge of its future destination. Random realisations of a Markov process can also display a large amount of variability and do not, in general, resemble realistic target trajectories.

The typical measurement model approximates the sensor image with a collection of discrete points at each frame to enable point measurement tracking. In conventional active signal processing systems, the intensity map data is approximated with a collection of point measurements via a detection threshold process. The resulting point detections are passed as inputs into the tracker, whose role is to link the measurements with time and provide filtered estimates. This approach is often sufficient for detecting and tracking high SNR targets but can often fail in

the case of low SNR targets, as the process of reducing the intensity map data to point measurements results in the loss of valuable information. A low detection threshold is also often necessary in order to increase the probability of detecting a low SNR target, however this can result in a high false alarm rate. False alarms are undesirable as they can potentially corrupt true tracks or give rise to a false track over time.

1.3 Thesis Scope and Overview

The objective of this research is to develop a robust solution to the problem of underwater tracking using active sonar systems. This thesis will do so by investigating alternative target and measurement models under the Bayesian framework based on Hidden Reciprocal Processes (HRPs) and Track-Before-Detect (TkBD), respectively, and consider their application to active sonar tracking.

In Chapter 2, a summary of the current approaches to the active sonar tracking problem is presented. This survey encompasses techniques based on conventional point measurement tracking as well as the recent application of TkBD and multistatics to the active sonar problem. TkBD is an alternative to point measurement tracking in which the full raw sensor data is supplied to the tracker. It is important to note that the term TkBD has been used in the wider literature to refer to applications in which a sequential likelihood ratio test (SPRT) [139] has been applied to a sequence of point measurements [15]. In this thesis, we will only focus on TkBD methods in which the tracker is supplied with the full intensity map image data. Chapter 2 concludes with a general review of conventional point measurement tracking based on Markov processes, TkBD methods, and HRPs.

Chapter 3 introduces an alternative target model to the Markov process that conditions on a known destination state. This new model is based on HRPs, which impose a joint distribution on the initial and final states. A Maximum Likelihood Sequence Estimate (MLSE) for HRPs based on the Markov bridge approach is derived.

> **The first key contribution is the first formal derivation of the MLSE for a HRP in which the random variables are finite-state**. The details of this contribution have been published in the following journal article:
>
> - L. B. White and H. X. Vu. Maximum likelihood sequence estimation for hidden reciprocal processes. *IEEE Transactions on Automatic Control*, 58(10):2670–2674, October 2013.

The assumption of a destination point can be potentially useful in the identification of two

crossing tracks or the classification of targets and clutter. The problem of joint tracking and classification has been considered in the literature for a single sensor [23,59] and multiple sensors in the radar context [128] but is outside the scope of this thesis.

The remainder of the thesis will focus on an alternative measurement model to point measurement tracking. This paradigm is referred to as Track-Before-Detect (TkBD) and is based on providing the sensor with an intensity map as an input into the tracker. TkBD algorithms eliminate the thresholding process and as a result, all information relating to the target and environment are preserved in the measurement. TkBD can be thought of as an automated tracking algorithm that takes in a sequence of intensity images rather than thresholded detection inputs. This thesis will focus on a particular TkBD algorithm based on Expectation-Maximisation (EM) data association called the Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT) as it handles multiple targets with low complexity.

In Chapter 4, the derivation and implementation of the H-PMHT is reviewed in detail and an alternative implementation via the Viterbi algorithm is proposed. We also discuss its key limitations that arise from its quantisation step and its assumptions of independence through the multinomial measurement model.

The second key contribution is the first implementation of the H-PMHT using a dynamic programming fixed-grid approximation through application of the Viterbi algorithm. The details of this contribution have been published in the following conference and journal articles:

- H. X. Vu and S. J. Davey. Track-before-detect using Histogram PMHT and dynamic programming. In *Proceedings of the 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–8, December 2012.
- S. J. Davey, M. Wieneke, and H. Vu. Histogram-PMHT unfettered. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):435–447, June 2013.

A problem with the H-PMHT is that it fails to model fluctuating target amplitude, which can degrade performance in realistic sensing conditions. In Chapter 5, this limitation is addressed by replacing the H-PMHT multinomial measurement model with a Poisson mixture process. The new Poisson mixture is shown to be consistent with the original H-PMHT modelling assumptions but it now allows for a randomly evolving mean target amplitude state with instantaneous fluctuations. This new TkBD algorithm is referred to as the Poisson H-PMHT. The Bayesian prior on the target state is also modified to ensure more robust performance.

The third key contribution is the first derivation of the H-PMHT assuming a Poisson distribution on the random number of target measurements. This derivation allows for a randomly evolving mean target amplitude and includes the proposal of an alternative state prior density that results in more consistent target tracking. The main contributions of this chapter are summarised in the following conference article and journal submission:

- H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.C. Lim. Histogram-PMHT with an evolving Poisson prior. In *2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2015.
- H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.C. Lim. A new state prior for the Histogram-PMHT. *IEEE Signal Processing Letters, in preparation.*

Chapter 6 proposes a novel TkBD algorithm based on the application of EM data association to a new measurement model that describes continuous valued intensity maps directly. This model is based on an interpolated version of the Poisson distribution, which is shown to be approximately a probability density function (pdf) on the non-negative real line for measurement rate $\lambda > 4$. The interpolated Poisson "distribution" can also be shown to obey an approximate superposition principle for $\lambda > 4$. In this chapter, we will treat the interpolated version of the Poisson distribution as a probability measure even though it is not strictly a pdf, and make use of its associated properties to derive a TkBD that is similar in principle to the H-PMHT but avoids the intermediate quantisation stage inherent in the H-PMHT. The resulting algorithm is referred to as the Interpolated Poisson-PMHT (IP-PMHT). Note that the derivation of the IP-PMHT depends on the validity of the superposition and pdf approximations.

The fourth key contribution is the first derivation of a TkBD algorithm assuming an Interpolated Poisson distribution on the energy generated by an individual target. Under this formulation, direct estimation of the measurement likelihood is possible, eliminating the need for an intermediate quantisation step. The main contributions of this chapter are the topics of the following conference article:

- H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.-C. Lim. H-PMHT with a Poisson measurement model. In *Proceedings of the 2013 International Conference on Radar (Radar)*, pages 446–451, September 2013.

Chapter 7 considers the application of several TkBD algorithms based on EM data association to the active sonar problem through a comparative study using trial data from an active towed array sonar. The TkBD algorithms are modified to incorporate changes in target appearance with received array bearing, and are shown to give improved SNR and state estimation performance compared with a conventional point measurement tracker.

> **The fifth key contribution is a comparison of several TkBD algorithms with a conventional point measurement tracker using trial data from an active towed array sonar system. The TkBD algorithms considered are the standard H-PMHT, the Poisson H-PMHT and the IP-PMHT. The TkBD algorithms also feature a modification to the point spread function to include bearing dependence.** The main contributions of this chapter are summarised in the following conference article:
>
> - H. X. Vu, S. J. Davey, F. K. Fletcher, S. Arulampalam, R. Ellem, and C.-C. Lim. Track-before-detect for an active towed array sonar. In *Proceedings of Acoustics 2013*, November 2013.

An alternative sensing paradigm has also been proposed to address the problems associated with active sonar tracking. This approach is based on changing the mode of operations from monostatic (single co-located transmitter and receiver) or bistatic (single separated transmitter and receiver) to a multistatic architecture consisting of multiple spatially diverse transmitters and receivers. In a multistatic configuration, detection information can be potentially fused from a number of source-receiver combinations to provide increased coverage and probability of detection over traditional configurations. The use of distributed sensors in multistatic framework has been shown to provide significant performance gains in other domains [18, 145]. However the problems inherent in active sonar such as high false alarm rate and multi-path reflections can often be amplified in multistatic systems due to the increased number of source-receiver geometries. Multistatic systems are inherently more complex as issues such as sensor registration and localisation, and a scheme for the optimal fusion of detections and tracks must also considered. Owing to time constraints, the scope of this thesis will be limited to only techniques that improve the tracking aspects of active sonar systems.

In Chapter 8, the thesis is concluded by discussing the limitations of the proposed algorithms and possible avenues for future work.

Chapter 2

Background

This chapter introduces the general tracking problem and discusses the three major causes for uncertainty in object tracking. This chapter then focuses on the Bayesian state estimation problem for active sonar tracking and provides an overview of the various approaches to the problem. A general review of conventional point measurement tracking, Track-Before-Detect (TkBD) methods, and Hidden Reciprocal Processes (HRPs) is also presented.

A survey of the literature reveals that although there have been various approaches to solving the active sonar tracking problem, this area is mostly limited to applications of a single algorithm to simulated sonar data featuring Markov targets. A comprehensive study comparing the performance of different tracking algorithms to trial sonar data is notably absent. Although there have been some applications of TkBD to active sonar, the research has been mostly limited to conventional point measurement tracking.

The thesis will focus on the following areas of research for the active sonar tracking problem:

- the application of alternative target models such as HRPs,
- the application of alternative measurement models based on the TkBD paradigm,
- a comparative study to determine the merits of applying conventional point measurement tracking and TkBD to trial sonar data.

The chapter is arranged as follows: Section 2.1 formulates the generic tracking problem; Section 2.2 establishes the active sonar tracking problem for both conventional point measurement tracking and TkBD; a review of the various approaches to active sonar tracking is also presented; Section 2.3 provides a general review of linear and non-linear point measurement tracking for Markov processes; Section 2.4 introduces an alternative target model based on HRPs and

discusses its application to target tracking; Section 2.5 presents an overview of the various TkBD algorithms available in the literature.

2.1 The General Tracking Problem

In general, the goal of object tracking is to determine the location, trajectory or characteristics of some target of interest using a series of noisy measurements from a sensor. The problem can be challenging owing to the large number of uncertainties that can arise from modelling the target, sensor and environment. These uncertainties can be separated into three main categories. The first category considers the problem of model order, that is, how many targets are assumed to be in the surveillance region. The solution to this problem considers models for track existence and approaches for initialisation. Once the model order is determined, the second problem to consider is data association, that is, which measurements should be associated to which target. The last problem deals with target state estimation and is generally referred to as filtering. In the following subsections, each problem is discussed in more detail.

2.1.1 Model Order

In many practical applications, targets can appear and disappear from the surveillance region at various times and the problem of estimating the number of targets at a specific time is a non-trivial task. In some applications, the model order or number of targets is simply assumed to be known. Alternatively, tracks can be initiated on all detections that appear in every frame. However in some applications, a large number of spurious detections can arise due to random background noise and assuming that each detection is a potential track can be inefficient.

A number of track initialisation schemes have been proposed to link together persistent detections across frames. The most common track initialisation procedure is known as the M/N rule [21], which initialises tracks on any M out of N series of point detections over time that are consistent with the motions of some target of interest.

A more principled approach to track initialisation is to consider the track probability of existence. This method assumes that each detection is a potential track and calculates existence measures to determine whether tracks should be kept or discarded. If a track's probability of existence exceeds a predetermined threshold, then the track is considered a confirmed track. Likewise, a track can be terminated if its existence probability falls below another predetermined threshold.

2.1.2 Data Association

Another complication in tracking is measurement origin uncertainty. In general, measurements from a sensor are assumed to also contain detections that are not from the target of interest. These measurements are regarded as false detections and when passed to the tracker, can result in false tracks. As stated earlier, spurious false detections can arise from random background noise and generally do not result in false tracks, however persistent detections due to environmental boundaries or from targets of non-interest can generate false tracks and also potentially corrupt tracks from targets of interest. One of the main difficulties in tracking is determining which measurements arise due to each target and which measurements are the result of false alarms or due to objects that are not of interest. This problem is referred to as data association.

There are a number of data association schemes available in the literature. One of the simplest schemes is based on the Nearest Neighbour (NN) approach, which associates each target with its closest measurement [14]. A variant of NN is the Global Nearest Neighbour (GNN) method that ensures that a measurement is associated with at most one target. An obvious drawback to these methods is that the NN measurement is not always the true target measurement.

The aforementioned data association schemes are based on 'hard' associations, in which a measurement can only be associated with at most one target. Probabilistic Data Association (PDA) is an alternative paradigm that performs 'soft' associations by expressing the target state probability density function (pdf) as a weighted sum of target state pdfs over data association hypotheses and provides expressions to determine the probabilities of these hypotheses [27]. The resulting pdf is a Gaussian mixture, which is then approximated by a single Gaussian. The use of a single Gaussian component to approximate the Gaussian mixture arising from the sum of pdfs over data association hypotheses can be problematic, especially when the mixture has more than one dominant component. The PDA algorithm also assumes that there is only one target.

A variant of PDA is the Integrated PDA (IPDA) algorithm which combines existence with data association by extending the target state-space with a binary existence variable that is assumed to evolve according to a Markov Chain [27, 94]. The IPDA provides equations for recursively updating the target states and the probability of target existence based on the PDA approach for data association. The probability of existence can then be used to automate track management.

Note that the algorithms discussed up until now assume the existence of only a single target. However, the problem of data association becomes more difficult when there are multiple targets of interest for which data association must be performed. It is possible to extend single

target algorithms to multi-target tracking by running a bank of single target filters in scenarios featuring well-separated targets. Joint Probabilistic Data Association (JPDA) is an alternative scheme developed for multi-target tracking. It is based on PDA and calculates the probability of measurement to target associations jointly across all targets. Each target is represented by a marginalised pdf (by marginalising over all other targets), which are then approximated by a single Gaussian for each target. For a large number of targets or measurements, JPDA quickly becomes infeasible due to the combinatorial complexity of iterating over all possible measurement to target associations. Like the PDA, JPDA can also be extended to include target existence; the resulting algorithm is called the Joint Integrated Probabilistic Data Association (JIPDA) [93]. However, the performance of PDA-based algorithms can suffer in scenarios with closely spaced targets, as measurements from different targets are clustered together and can corrupt target tracks. Another alternative data association scheme for multi-target tracking is the Probabilistic Multi-Hypothesis Tracker (PMHT), which performs data association based on Expectation-Maximisation (EM) [121].

Multi-Hypothesis Tracking (MHT) is widely considered the best algorithm for multi-target tracking. It is based on postponing hard data associations until sufficient evidence becomes available [32]. At each time, the MHT stores all possible track hypotheses regarding past and current associations, which results in a combinatoric explosion of hypotheses. Pruning schemes based on deleting the most unlikely potential tracks are used to reduce the number of hypotheses.

2.1.3 Filtering

The final issue in tracking is the problem of estimating the states of the targets that correspond to existing tracks. This problem is also known as filtering. In this subsection, the generic filtering problem is introduced and a summary of the literature on filtering methods is provided.

Assume a scenario in which we wish to estimate the kinematic state of a target over time using a sequence of noisy measurements. In order to do this, it is necessary to define an appropriate filtering framework. The problem can be formulated generically as follows.

Define the state vector $\boldsymbol{x}_t^m$, which evolves with time $t \in \mathbb{N}$, where $\mathbb{N}$ is the set of all natural numbers, $m = 1, \ldots, M$ denotes the target index in the case of a multi-target scenario, and M is the total number of targets. Assume that the target state cannot be directly observed, and there is some sensor that collects observations $\boldsymbol{z}_t$ of the target at time scan t. Let $\mathsf{Z}_t = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t\}$ denote the collection of measurements observed up until time t.

Generally, when analysing a dynamic system, two models are required: the target dynamics and

measurement models. The target dynamics model describes the evolution of the target state with time and is almost always assumed to be a Markov process. For the most part of this thesis, it is assumed that the target dynamics can be modelled using a 1^{st} order Markov model $p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)$ such that the target state $\boldsymbol{x}_t^m$ at time t only depends on its previous state $\boldsymbol{x}_{t-1}^m$. The measurement model $p(\boldsymbol{z}_t|\boldsymbol{x}_t^m)$ maps the state into the measurement space. The typical measurement model approximates the sensor image with a collection of discrete points, generally by normalising the signal, applying a detection threshold, and performing clustering and centroiding at each frame. These point detections are passed as inputs to a tracker, whose role is to link the detections with time and provide filtered state estimates. This type of tracking is generally referred to as point measurement tracking.

The problem with point measurement tracking is that the act of reducing the sensor image to a collection of points through a detection threshold throws away valuable information and removes the opportunity to accumulate evidence over multiple pings [14]. Also, in conventional point measurement tracking, the clustering stage is required to meet the constraint that a target can only generate one measurement at each frame, however by doing so, it throws away any information regarding the shape of the target. Furthermore, some clustering algorithms such as K-means clustering are N-P hard problems and suboptimal approximations are generally required [70]. Finally, most centroiding algorithms calculate an average between clustered points, which can give rise to large errors, particularly in scenarios featuring closely spaced targets. On the other hand, TkBD methods are able to both accumulate information over time and maintain information about the shape of the target.

The recursive Bayesian approach to dynamic state estimation requires the construction of a posterior pdf $p(\boldsymbol{x}_t^m|\mathsf{Z}_t)$ of the state based on all available information, including information gained from Z_t, the noisy measurements received up to time t. The pdf is assumed to contain all available information of the state and is considered to be a complete solution to the estimation problem. The recursive Bayesian filter used to update the pdf given new information consists of two stages:

Prediction stage: The state is propagated forward in time using the target dynamics model $p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)$. The pdf broadens due to the uncertainty in the system model. Assume that the initial density of the state $p(\boldsymbol{x}_0^m|\boldsymbol{z}_0) = p(\boldsymbol{x}_0^m)$ is known and suppose the posterior pdf at the previous time step $t-1$ is given as $p(\boldsymbol{x}_{t-1}^m|\mathsf{Z}_{t-1})$. The previous estimate state can be predicted to the current time t using the Chapman-Kolmogorov equation to obtain the prediction density (prior pdf of the state), which can be expressed as,

$$p(\boldsymbol{x}_t^m|\mathsf{Z}_{t-1}) = \int p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)p(\boldsymbol{x}_{t-1}^m|\mathsf{Z}_{t-1})d\boldsymbol{x}_{t-1}^m. \tag{2.1}$$

Update stage: The information contained in the new measurement is used to modify (tighten) the pdf. In the update stage of the filter, it is assumed that the measurement $\boldsymbol{z}_t$ becomes available and can be used to modify the prior pdf $p(\boldsymbol{x}_t^m|\mathsf{Z}_{t-1})$. Then the state update is performed using Bayes Theorem:

$$\begin{aligned} p(\boldsymbol{x}_t^m|\mathsf{Z}_t) &= p(\boldsymbol{x}_t^m|\boldsymbol{z}_t, \mathsf{Z}_{t-1}) && (2.2)\\ &= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^m, \mathsf{Z}_{t-1})p(\boldsymbol{x}_t^m|\mathsf{Z}_{t-1})}{p(\boldsymbol{z}_t|\mathsf{Z}_{t-1})} && (2.3)\\ &= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^m)p(\boldsymbol{x}_t^m|\mathsf{Z}_{t-1})}{p(\boldsymbol{z}_t|\mathsf{Z}_{t-1})}, && (2.4) \end{aligned}$$

where $p(\boldsymbol{z}_t|\boldsymbol{x}_t^m)$ is the likelihood function defined by the measurement model and known statistics of the measurement noise sequence $\boldsymbol{\psi}$, and

$$p(\boldsymbol{z}_t|\mathsf{Z}_{t-1}) = \int p(\boldsymbol{z}_t|\boldsymbol{x}_t^m)p(\boldsymbol{x}_t^m|\mathsf{Z}_{t-1})d\boldsymbol{x}_t^m. \quad (2.5)$$

For the remainder of this section, the discussion on tracking solutions will assume that the target is modelled by a Markov process. For further details on conventional point measurement tracking based on Markov processes, see section 2.3.

In the case of a single Markov target scenario in which both the target dynamics and measurement models are assumed to be linear functions with Gaussian noise, the optimal finite dimensional solution to the discrete-time recursive Bayesian state estimation problem is the Kalman Filter (KF). At every time, the KF assumes that the posterior pdf is Gaussian and therefore can be completely characterised by its mean and covariance. See subsection 2.3.1 for more details on the KF. If however, either the target dynamics or the measurement model is non-linear or non-Gaussian, the KF is no longer optimal and non-linear filtering methods based on approximations or suboptimal solutions must be considered. A review of these algorithms is provided in 2.3. These techniques can be separated into four main categories:

1. **Analytic approximations**, e.g. Extended Kalman Filter (EKF) described in subsection 2.3.2.

2. **Grid-Based approximations**, e.g. Forward-Backward and Viterbi algorithms described in subsections 2.5.1 and 2.5.2, respectively.

3. **Sampling Approaches** [112], e.g. Unscented Kalman Filter (UKF) and Particle Filter (PF) described in subsections 2.3.3 and 2.3.4, respectively.

4. **Random Finite Sets**, e.g. Probabilistic Hypothesis Density filter (PHD) and Cardinalised Probabilistic Hypothesis Density (CPHD) filter described in subsection 2.3.5.

The Markov process commonly used for target modelling assumes that the state evolves without knowledge of its future destination. Random realisations of a Markov process can also display a large amount of variability and do not, in general, resemble realistic target trajectories. An alternative to the Markov assumption is the reciprocal process (RP), which assumes conditioning on a known destination state. In subsection 2.4, we introduce the hidden reciprocal model and discuss its application to target tracking.

The algorithms discussed above assume that the measurement model approximates the sensor intensity map, an image, with a collection of point detections. However, reducing the sensor data to point detections results in the loss of valuable information. This method is generally sufficient for tracking high Signal-to-Noise Ratio (SNR) targets, but can fail in the case of low SNR targets. One alternative to point measurement tracking is Track-Before-Detect (TkBD), which provides the sensor image as an input into the tracker. This allows concurrent detection and tracking to be performed. Current TkBD algorithms are based on modifying conventional point measurement trackers to accommodate the sensor image as an input. See section 2.5 for a general review of TkBD algorithms.

In this section, we have formulated the tracking problem in the generic sense. In the next section, we describe the target and measurement models for the problem of underwater tracking using active sonar.

2.2 Active Sonar Tracking Problem

In active sonar tracking, the main objective is to identify the number of targets and estimate their trajectories over time using a sequence of noisy sonar measurements. This is a non-trivial task as the vast majority of these measurements are attributed to clutter objects and measurements from targets of interest have low probability of detection.

In section 2.2.1, we establish the active sonar problem for conventional point measurement tracking and review the literature based on this approach. The requirement for more robust trackers against low SNR targets has led to the development of an alternative measurement model called TkBD. In Section 2.2.2, we motivate the application of TkBD to the active sonar problem and provide an overview of the research in this space. Another method used to track low SNR targets is multistatics, which is based on the fusion of measurements from several sensors. Multistatics exploits detections from a network of sensors to improve target probability of detection. In section 2.2.3, we review multistatics in the context of the active sonar problem.

2.2.1 Conventional Point Measurement Problem

In conventional active signal processing systems, the intensity map data is approximated with a collection of point measurements via a detection threshold process. The resulting point detections are passed as inputs to the tracker, whose role is to link the measurements over time and provide smoothed estimates.

Assume a scenario in which we wish to estimate the kinematic state of M underwater targets over time using a sequence of noisy measurements from a sonar sensor. Again, define the state vector $\boldsymbol{x}_t^m$, which evolves with time t and $m = 0, \ldots, M$ denotes the component index. A component can be attributed to either a clutter or target object, therefore define component $m = 0$ to be the clutter component.

For conventional active sonar target tracking, it is sufficient to describe the target state using position and velocity in two-dimensions. Define x_t^m and $\dot{x}_t^m$ to be the target position and velocity in the x-direction, respectively. Similarly, define y_t^m and $\dot{y}_t^m$ to be the respective target position and velocity in the y-direction. The target state for point measurement tracking is defined as:

$$\boldsymbol{x}_t^m \;=\; \left[\begin{array}{cccc} x_t^m & \dot{x}_t^m & y_t^m & \dot{y}_t^m \end{array}\right]^T, \tag{2.6}$$

where $[\cdot]^T$ denotes the transpose operation. It is assumed that the state vector contains all relevant information about the system.

Target Model: The target model describes the target state evolution with time and can be expressed in terms of a discrete-time stochastic model,

$$\boldsymbol{x}_t^m = f_{t-1}(\boldsymbol{x}_{t-1}^m) + \boldsymbol{v}_{t-1}, \tag{2.7}$$

where f_{t-1} denotes the state transition function from $\boldsymbol{x}_{t-1}^m$ to $\boldsymbol{x}_t^m$ and $\boldsymbol{v}_{t-1}$ is an independent identically distributed (iid) system noise sequence representing uncertainties in the target motion. For the active sonar tracking problem, the target model needs to capture the dynamics of an underwater target. It is assumed that a nearly-constant-velocity model is sufficient. In this case, the function f_{t-1} is set to be a known matrix F_{t-1} describing a linear state transition from $\boldsymbol{x}_{t-1}^m$ to $\boldsymbol{x}_t^m$.

Measurement Model: The measurement model relates the noisy measurements to the state $\boldsymbol{x}_t^m$. In conventional point measurement sonar tracking, measurements are commonly received in range and bearing. Let $\mathbf{Z}_t = \{\boldsymbol{z}_t^j\}$ for $j \in \{1, \ldots, \mathbb{m}_t\}$ denote the set of point measurements received at time t where $\mathbb{m}_t$ denotes the number of measurements received at time t. Note that since the detection probability of each target is, in general, less than unity, there is no guarantee

that every target will produce a point measurement at each time. Furthermore, some of these detections may originate from clutter.

Suppose target m is associated with measurement j_m at time t. Then,

$$\boldsymbol{z}_t^{j_m} = \zeta_t(\boldsymbol{x}_t^m) + \boldsymbol{\psi}_t, \tag{2.8}$$

where $\zeta_t(\boldsymbol{x}_t^m)$ denotes the measurement function that maps the state into the measurement space and $\boldsymbol{\psi}_t$ is an iid measurement noise sequence. Clutter detections are assumed to be uniformly distributed across the surveillance region and the total number of clutter detections at each time is assumed to follow a Poisson distribution. Note that (2.8) is a very simple model and does not accommodate scenarios in which targets can generate more than one detection per frame. Nevertheless, most conventional point measurement trackers in active sonar assume point detections adhere to this model.

In conventional active sonar tracking, the sensor measures the position of the target. Depending on the waveform, it may also measure Doppler (which is related to the velocity) component of the target. Throughout this thesis, it is assumed that the measurement process only observes the position component of the target, and thus the likelihood $p(\boldsymbol{z}_t^{j_m}|\boldsymbol{x}_t^m)$ is independent of the target velocity component.

As the measurements in active sonar are generally polar (range and bearing), and the target state is more naturally modelled in a Cartesian frame, the literature on point measurement tracking for active sonar is generally limited to non-linear filtering techniques. Comparisons of various point measurement trackers for the multi-target active sonar problem can be found in [95,142]. Both articles apply non-linear filtering techniques based on GNN and PDA schemes to a simulated active sonar environment. These association methods are discussed in subsection 2.1.2. A non-linear filtering technique based on an interacting multiple model for a simulated sonar environment has also been proposed [5]. The application of Random Finite Set (RFS) theory to active sonar tracking is considered in [26]. The authors compared a Probability Hypothesis Density (PHD) filter with standard multi-target trackers on both simulated and trials data from an Autonomous Underwater Vehicle (AUV). For more details on RFS theory, refer to subsection 2.3.5. The extension of passive tracking techniques to the active sonar tracking has also been proposed and applied to simulated data [107].

2.2.2 Track-Before-Detect Problem

Conventional tracking techniques reduce sensor data intensity maps to point measurements using a detection thresholding process. The detection of a high SNR target return in a single frame is generally possible. However the task of detection becomes more difficult for low SNR

targets. TkBD algorithms are a natural solution for tracking low SNR targets as the declaration of a target detection can be delayed until after a series of frames have been processed. It does so by supplying raw sensor data to the tracker, rather than point measurements from thresholded detections. This allows potential targets to be detected and tracked simultaneously. In contrast to conventional point measurement tracking, TkBD does not throw away any data, but is inherently more computationally intensive. It is important to note that the term TkBD has been used in the wider literature to refer to applications in which a sequential likelihood ratio test (SPRT) [139] has been applied to a sequence of point measurements. An example of this is the sequential Maximum Likelihood-Probabilistic Data Association (ML-PDA) filter, which assumes a target has deterministic motion [15]. In this thesis, we will only focus on TkBD methods in which the tracker is supplied with the full intensity map image data.

Again, we can define the state vector for the m^{th} target as $\boldsymbol{x}_t^m$, which evolves with time t where $m = 1, \ldots, M$. In the TkBD case, it is common to supplement the state vector with the target amplitude,

$$\boldsymbol{x}_t^m = \left[\begin{array}{ccccc} x_t^m & \dot{x}_t^m & y_t^m & \dot{y}_t^m & A_t^m \end{array} \right]^T, \tag{2.9}$$

where A_t^m denotes the amplitude of the m^{th} target return at time t. As in conventional point measurement tracking, both a target and a measurement model are required when analysing a dynamic system.

Target Model: The target model is assumed to be the same as in the conventional point measurement tracking case.

Measurement Model: For the TkBD problem, let $\mathbf{Z}_t$ now represent the intensity map, an image, that is received from the sonar sensor. The measurement model relates images $\mathbf{Z}_t$ in range and bearing to the target state $\boldsymbol{x}_t^m$. Let $\boldsymbol{z}_t^i$ denote the energy in the ith pixel of the measurement image at time t, and let $\mathbf{Z}_t = \{\boldsymbol{z}_t^i\}$ for $i = 1, \ldots, I$ represent a stacked vector of all the pixels in the image where I is the total number of pixels in the measurement image. For ease of presentation, we have used a stacked vector to represent the image to allow single index referencing. A two dimensional representation could just as easily have been used. We assume a point-scatterer target, such that the target contribution to the measurement image can be described purely in terms of the point spread function (psf), $h(\boldsymbol{x}_t^m)$,

$$\mathbf{Z}_t = \sum_{m=1}^{M} A_t^m h(\boldsymbol{x}_t^m) + \boldsymbol{w}_t, \tag{2.10}$$

where $\boldsymbol{w}_t$ is an iid measurement noise sequence for the image. Note that the psf is a property of the sensor and can vary with different sensors. The sum in (2.10) represents an incoherent

combination of target power. For some systems, it may be appropriate to consider coherent superposition, in which case both A_t^m and $\boldsymbol{w}_t$ are complex.

The recursive Bayesian approach can again be employed to construct the state pdf. However, as the output of the TkBD measurement model generally consists of a two-dimensional image, $p(\mathbf{Z}_t|\boldsymbol{x}_t^m)$ is now defined as the likelihood of seeing the sequence of images given the state of the target. Assuming independent pixel noise, the likelihood for the image $\mathbf{Z}_t$ can be factorised as follows,

$$p(\mathbf{Z}_t|\boldsymbol{x}_t^m) = \prod_{i=1}^{I} p(\boldsymbol{z}_t^i|\boldsymbol{x}_t^m). \tag{2.11}$$

As the psf in (2.10) is highly non-linear, the TkBD measurement model defined in (2.10) is also a highly non-linear function of the target state and only approximations or suboptimal solutions can be considered. We point out however, that techniques based on discrete Hidden Markov Models (HMMs) are not restricted by assumptions of linearity and Gaussian noise, making them naturally suited to TkBD applications. A review of early TkBD algorithms for image sequencing, radar and sonar can be found in [125]. These techniques have the potential to provide significant gains for both detecting and tracking low SNR targets in high clutter scenarios [39].

As a prelude to TkBD in active sonar, techniques based on the inclusion of target and clutter amplitude information within conventional point measurement trackers to improve tracking performance for active sonar have also been considered for simulated data [4, 25, 82] and for active sonar trials data [71, 104, 114].

The first applications of TkBD to active sonar were based on dynamic programming techniques, which employ fixed grids to model the propagation of target states with time [16, 46, 125]. In similar work, a TkBD technique using a HMM for detection in the active sonar problem was proposed in [102]. This work used a new track initiation scheme named a sequential Markov detector that combines a HMM with a sequential detector and delays the decision of a target presence or absence until after a number of time frames have been processed. Results showed improved detection performance on simulated data over conventional trackers.

Most of the TkBD applications have been demonstrated on simulated data and the application to real-world data has been limited; a Generalised Likelihood Ratio Test (GLRT) was developed for a bistatic sonar scenario using sea trials data [97]; the direction of arrival for a series of sonar returns was estimated using a PF adapted to TkBD for the reconstruction of an underwater bathymetric scene [113]. The application of other alternative TkBD algorithms to active sonar data has also been limited. Moreover, previous applications of TkBD have failed to address the issues that are unique to active sonar.

This thesis will focus on the application of a particular TkBD algorithm called the Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT) to the active sonar tracking problem. For a detailed review of the derivation and implementation of the H-PMHT algorithm, the reader is referred to Chapter 4.

A comparison of TkBD methods to conventional point measurement trackers is notably absent in the literature. In Chapter 7, the performance of the H-PMHT is compared with a conventional point measurement tracker based on IPDA [27, 94]. The H-PMHT is also modified to incorporate changes in target appearance with received array bearing. The benefits of TkBD over conventional tracking is analysed in two representative acoustic environments using trials data from an active towed array sonar system.

2.2.3 Multistatics

Multistatic active sonar systems have been proposed as an alternate means for tracking low SNR underwater targets. Traditionally, active sonar systems have been operated in a monostatic (single co-located transmitter and receiver) or bistatic (single separated transmitter and receiver) mode. Multistatic systems extend these systems to multiple spatially diverse transmitters and receivers. In a multistatic configuration, the measurement model assumes detections can be fused from a number of source-receiver combinations, which can provide increased coverage and increased probability of detection over monostatic and bistatic systems. The research in this area has been primarily focused on adapting point measurement trackers and data association algorithms to sonar sensor networks [31, 56, 61–64, 68].

However the problems inherent in active sonar such as high false alarm rate and multi-path reflections can often be amplified in multistatic systems. Furthermore, multistatic systems have additional complexities such as sensor registration, localisation problems, and require fusion of measurements from multiple sensors [29, 30, 33].

Another related area of research is the concept of Multiple-Input Multiple-Output (MIMO) [54]. Originally conceived in communications, and recently adapted to the radar and sonar problem [11, 75], MIMO systems are a specific statistical example of multistatics. Unlike conventional beamforming techniques that transmit correlated signals to form directional beams, MIMO systems capitalise on spatial diversity between sensors to improve system performance.

This thesis focuses on TkBD techniques for monostatic systems but we point out that TkBD can be extended to accommodate multistatic networks. The problem of multistatic TkBD has been considered in the imaging context [123], multistatic radar [57, 74] and for MIMO radar [66, 67, 99, 156]. The first application of TkBD to multistatic active sonar considered the issue

of fusing measurements from a network of passive sonars illuminated by a single active source [90]. This was been followed by the application of dynamic programming techniques [46] and Likelihood Ratio Detection and Tracking (LRDT) [115] to at-sea multistatic sonar data.

2.3 Conventional Point Measurement Filtering based on Markov Processes

In this section, a general overview of linear and non-linear conventional point measurement tracking based on Markov processes is presented. In particular, we focus on the filtering algorithms used for this problem. Subsections 2.3.1-2.3.4 describe single target filters while subsection 2.3.5 discusses multi-target filters.

2.3.1 Kalman Filter

Under linear-Gaussian assumptions, the KF is the optimal solution to the recursive Bayesian state estimation problem. It assumes the posterior density is Gaussian at each time step and therefore can be completely described by its mean and covariance. Recall that for conventional point measurement tracking, the target and measurement models are defined in (2.7) and (2.8), respectively, and are given by,

$$\begin{aligned} \boldsymbol{x}_t &= f_{t-1}(\boldsymbol{x}_{t-1}) + \boldsymbol{v}_{t-1}, \\ \boldsymbol{z}_t &= \zeta_t(\boldsymbol{x}_t) + \boldsymbol{\psi}_t. \end{aligned}$$

where $\boldsymbol{x}_t^m$ and $\boldsymbol{z}_t$ denote the state and its associated measurement, respectively. Let $\mathcal{N}(\boldsymbol{x}; \mu, \mathsf{P})$ be a Gaussian density about $\boldsymbol{x}$ with mean μ and covariance P. Recall that $\mathsf{Z}_t = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t\}$ denotes the set of measurements received up to time t. If the posterior density $p(\boldsymbol{x}_{t-1}|\mathsf{Z}_{t-1})$ is assumed to be Gaussian, then the density at the next time step $p(\boldsymbol{x}_t|\mathsf{Z}_t)$ is also Gaussian if the following assumptions are valid:

- The noise sequences $\boldsymbol{v}_{t-1} \sim N(\boldsymbol{0}, \mathsf{Q}_{t-1})$ and $\boldsymbol{\psi}_t \sim N(\boldsymbol{0}, \mathsf{R}_t)$ are uncorrelated with each other and with time. Note that Q_{t-1} and R_t denote the process and measurement covariance matrices, respectively;
- The process function $f_{t-1}(\boldsymbol{x}_{t-1})$ is known and linear;
- The measurement function $\zeta_t(\boldsymbol{x}_t)$ is known and linear.

Note that the target index m has been suppressed for notational simplicity. Under these assumptions, the state evolution and measurement models can be described as vector difference

equations,

$$\boldsymbol{x}_t = \mathsf{F}_{t-1}\boldsymbol{x}_{t-1} + \boldsymbol{v}_{t-1} \tag{2.12}$$

$$\boldsymbol{z}_t = \mathsf{H}_t\boldsymbol{x}_t + \boldsymbol{\psi}_t, \tag{2.13}$$

where F_{t-1} and H_t are the known linear process and measurement matrices.

Given the posterior pdf at the previous time step $t-1$,

$$p(\boldsymbol{x}_{t-1}|\mathsf{Z}_{t-1}) = \mathcal{N}(\boldsymbol{x}_{t-1}; \hat{\boldsymbol{x}}_{t-1|t-1}, \mathsf{P}_{t-1|t-1}), \tag{2.14}$$

we can predict forwards in time and the prior pdf for the next time step t has the following form,

$$p(\boldsymbol{x}_t|\mathsf{Z}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \hat{\boldsymbol{x}}_{t|t-1}, \mathsf{P}_{t|t-1}). \tag{2.15}$$

The predicted mean $\hat{\boldsymbol{x}}_{t|t-1}$ and covariance $\mathsf{P}_{t|t-1}$ are defined by the standard KF equation [6]. When a new measurement $\boldsymbol{z}_t$ arrives, the posterior pdf for the next time step t has the following distribution,

$$p(\boldsymbol{x}_k|\mathsf{Z}_t) = \mathcal{N}(\boldsymbol{x}_t; \hat{\boldsymbol{x}}_{t|t}, \mathsf{P}_{t|t}). \tag{2.16}$$

The updated mean $\hat{\boldsymbol{x}}_{t|t}$ and covariance $\mathsf{P}_{t|t}$ can also be calculated by the standard KF equations [6]. Given the estimate of the posterior pdf of the state $p(\boldsymbol{x}_t|\mathsf{Z}_t)$ at the current time, the KF recursively calculates the mean and covariance of the state.

2.3.2 Extended Kalman Filter

The KF is the optimal recursive Bayesian filter for the linear-Gaussian case. However in reality, problems are often highly non-linear and non-Gaussian. In these cases, optimal filters cannot be applied and we must look towards approximations or suboptimal filters for solutions. The Extended Kalman Filter (EKF), the higher order EKF and the iterated EKF are examples of analytic approximations applicable to problems with either non-linear state dynamics or non-linear measurement models [108].

Consider the non-linear filtering problem in which both the process function f_{t-1} and measurement function $\zeta_t(\boldsymbol{x}_t)$ are assumed to be non-linear functions with additive white Gaussian noise sequences $\boldsymbol{v}_{t-1} \sim N(\mathbf{0}, \mathsf{Q}_{t-1})$, and $\boldsymbol{\psi}_t \sim N(\mathbf{0}, \mathsf{R}_t)$, respectively.

The EKF approximates the non-linear functions f_{t-1} and ζ_t by the first order terms in their Taylor series expansions. That is, f_{t-1} and ζ_t are approximated by local linearisations through their corresponding Jacobians $\hat{\mathsf{F}}_{t-1}$ and $\hat{\mathsf{H}}_t$, evaluated at $\hat{\boldsymbol{x}}_{t-1|t-1}$ and $\hat{\boldsymbol{x}}_{t|t-1}$, respectively. The

EKF and its cousins are analytic approximations as $\hat{\mathsf{F}}_{t-1}$ and $\hat{\mathsf{H}}_t$ have closed form solutions. As a result, the EKF requires the process and measurement models f_{t-1} and ζ_t to be continuous, differentiable functions of their arguments.

2.3.3 Unscented Kalman Filter

The Unscented Transform (UT) is a method for capturing the mean and covariance of a random variable that undergoes a non-linear transformation. The idea has been applied to the KF framework, resulting in the Unscented Kalman Filter (UKF) [78,141]. Unlike the EKF, the UKF does not attempt to approximate the non-linear functions f_{t-1} and ζ_t with analytic terms, but rather it approximates the posterior distribution of the state $p(\boldsymbol{x}_t|\mathsf{Z}_t)$ by a Gaussian random variable. We can regard the UKF as a statistical linearisation of the state as opposed to the EKF, which is based on an analytic linearisation.

The UKF represents the Gaussian density by a minimal set of deterministically chosen sample points and uses these sample points to completely capture the mean and covariance of the Gaussian density. When propagated through a non-linear system, the sample points capture the true mean and covariance of the posterior density of the state. If the true psf is highly non Gaussian, then there will be discrepancies between the higher order moments of the true pdf and its Gaussian approximations. If the problem is highly non-linear, it is possible to sample non-local effects even if the true mean and covariance are captured. The scaled version of the UT has been proposed that addresses this problem [141]. The main limitation with the UT is that it cannot adequately describe multi-modal pdfs.

2.3.4 Particle Filter

So far, we have considered non-linear filtering algorithms that make analytical or statistical Gaussian approximations about the posterior pdf. In this section, we introduce Particle Filters (PFs) as an alternative approach that does not require these approximations. Like the UKF, the PF also uses the sampling approach to solve the dynamical state estimation problem. The PF is founded upon Sequential Monte Carlo (SMC) estimation and is based on the idea that a probability density can be represented by a set of random samples or 'particles' with associated weights.

Let $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^N$ be a set of samples and associated weights that represents the pdf $p(\boldsymbol{x}_t|\mathsf{Z}_t)$. Then the posterior at time t can be approximated by,

$$p(\boldsymbol{x}_t|\mathsf{Z}_t) = \sum_{i=1}^{N} w_k^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i), \tag{2.17}$$

where $\delta(\cdot)$ denotes the delta function. As the number of samples become very large, the particle representation of the pdf becomes close to the true pdf, and the PF approaches the optimal Bayesian estimator.

To estimate $p(\boldsymbol{x}_t|\mathsf{Z}_t)$, suppose at the previous time step $t-1$, we have a set of samples and associated weights $\{\boldsymbol{x}_{t-1}^i, w_{t-1}^i\}_{i=1}^N$ that represent the posterior pdf $p(\boldsymbol{x}_{t-1}|\mathsf{Z}_{t-1})$. The PF propagates the particles $\boldsymbol{x}_{t-1}^i$ forward in time using a proposal density $q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{z}_t)$, which we refer to as the importance density. When a new measurement $\boldsymbol{z}_t$ arrives at time k, a new set of particles and weights are calculated to approximate $p(\boldsymbol{x}_t|\mathsf{Z}_t)$. The weights for the current time are updated according to

$$w_t^i \propto w_{t-1}^i \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{z}_t)}. \tag{2.18}$$

A problem with the PF is that the weights degenerate over time. That is, after a few recursions all but one particle has a negligible normalised weight [44]. A large amount of computational effort is wasted updating particles that have a negligible impact on the approximation of the posterior. Resampling has been proposed to combat the degeneracy problem and can be implemented whenever severe degeneracy is encountered. Effectively, resampling discards particles of negligible weight and multiplies those particles that contribute most to the approximation of the posterior density. However, resampling can also introduce other problems. Resampling can result in a loss of diversity among particles as at each resampling stage, the same particles are selected due to their high importance weights. Often this phenomenon occurs in problems with small process noise and can cause all particles to collapse into a single point, a situation referred to as sample impoverishment. Solutions to sample impoverishment have been proposed that incorporate a regularisation step in the PF [96].

2.3.5 Random Finite Sets

For most applications, the computation of the posterior distribution of the multi-target state is intractable due the exponential increase in complexity of enumerating over all possible measurement to target associations. Recently, the application of Random Finite Set (RFS) theory, also known as finite point processes, to the multi-target Bayesian tracking problem has been proposed to address this issue.

A RFS is defined as a random variable that takes values in an unordered finite set. Like any random variable, a RFS can be completely described by its probability distribution. As a result, equivalent notions of integration and density can be defined. As a RFS does not follow the normal notions of integration and density inherent in metric probability, the algorithm borrows

ideas of integration and density from FInite Set Statistics (FISST). This has led to the development of the Probability Hypothesis Density (PHD) filter [88], which forms a first moment approximation to the multi-target Bayes filter. The PHD filter proceeds by only propagating the first moment of the filter (posterior intensity function), rather than the normal posterior density itself; the moment is then approximated to close the Bayes recursion Note, however that the PHD filter suffers from the loss of higher order cardinality information. To overcome this, the Cardinalised PHD (CPHD) filter has been proposed as a generalisation to the PHD filter, by jointly propagating the posterior intensity function and posterior cardinality distribution [89]. In addition to these algorithms, the **M**ulti-Target **M**ulti-**B**ernoulli (MeMBer) filter has also been introduced as an approximation to the multi-target posterior density [88]. It does so by propagating the parameters of a multi-Bernoulli RFS, which are used to approximate the multi-target posterior density. Gaussian mixtures and PF implementations of these filters have been proposed [129, 131].

The problem with this approach is that it fails to link state estimates with time. In the multi-target case, this becomes an identification problem and suboptimal procedures can be used to associate new state estimates with existing tracks. Like the JPDA filter, this method also suffers in high clutter and scenarios featuring closely spaced targets.

2.4 Hidden Reciprocal Processes

Current techniques for target tracking predominantly assume a target dynamics model based on a HMM. These models generally assume a causal process such that the target state at any given time is only dependent on its previous states. In some scenarios, it may be more appropriate to assume a non-causal model such as a reciprocal process (RP) to represent the target dynamics. RPs are able to incorporate models that utilise target destination information by assuming a joint probability distribution on the initial and final states. This allows destination aware tracking to be performed [50]. The ability to specify statistically dependent target source and destination points may help to mitigate common problems encountered in multiple target tracking such as resolving targets in a crossing target scenario. This can potentially aid in distinguishing benign targets from targets of interest for classification purposes. RPs can also be used to detect anomalous behaviour in target trajectories by modelling the intent of a target through its destination point [49].

Jamison [77] first showed that an RP could be generated by pinning a Markov process at its endpoints and assigning a probability density to the endpoints. He also showed that two RPs can be generated by pinning the same Markov process at different endpoints. The RPs are con-

sidered equivalent as they are assumed to have the same dynamics. The application of RPs to the surveillance and tracking of merchant ships was first presented in [20] in which the authors show how predictive information can be incorporated into estimation for the continuous-time Gaussian case. In other work, [83] proposed a discrete-time, Gaussian RP for modelling and estimation. The authors derived an optimal fixed-interval smoother based on a Forward-Backward procedure, similar to the fixed-interval smoothers derived for Gauss-Markov processes.

The generalisation of current HMM-based TkBD methods to include acausal assumptions via HRPs is relatively straight forward for fixed-grid approximation techniques. Recently, the problem of optimal fixed-interval smoothing for finite-state, discrete-time HRPs was addressed via the Markov bridge approach in [19, 146, 147]. The resulting algorithm is similar to the Forward-Backward smoothers constructed for a HMM. In Chapter 3, we provide an extension to this work by utilising the Markov bridge approach to calculate the Maximum Likelihood Sequence Estimate (MLSE) for a HRP.

2.5 Track-Before-Detect

In general, TkBD methods can be divided into two approaches. The first approach considers TkBD in terms of tracking and estimation by treating the data as a measurement input into the system. The second approach considers TkBD from a detection perspective and is based on the fundamental idea that an improved SNR would result when an stationary signal is incoherently accumulated over a number of frames. In this section, an overview of TkBD techniques is presented with a focus from the tracking perspective. For a detailed review of TkBD techniques, the reader is referred to [39]. In subsequent chapters, we propose to modify and adapt some of these techniques to the active sonar problem.

Maybeck and Mercier [91] were the first to apply a non-linear filter to the TkBD problem by employing an EKF to the problem of detection and tracking using a forward looking infrared sensor. The EKF assumes that any non-linearity in the measurement image can be adequately described by a first order linearisation of the measurement process. In applications where the non-linearity is severe such as the TkBD problem, analytic approaches such as the EKF can perform poorly, except in the case of a high SNR target.

Techniques for TkBD then moved from the analytic to the sampling domain in which numerical fixed grid-based approaches were employed to approximate the state space. Barniv was the first to discretise the state space and describe the problem as a discrete HMM [9]. A HMM is a statistical model based on a Markov process in which the states of the process are 'hidden' and cannot be directly observed. He employed dynamic programming techniques in the form of

the Viterbi algorithm to search for the most likely target state sequence across the grid [7–9]. Tonissen furthered this work by analysing the detection statistics of the Viterbi algorithm [124]. An alternative HMM grid-based approach was implemented by Bruno via the Forward-Backward algorithm in which the posterior density was evaluated over the entire grid space and propagated with time [87].

Fixed grid approximation techniques are naturally multi-target algorithms, however the discretisation of the state space results in a higher computational complexity as a direct result of evaluating the state pdf over the entire grid. Much of this computation is wasted by evaluating the probability in the pdf tails where there is little to negligible probability. The desire for a more dynamic and adaptive grid led to the application of Particle Filter (PFs) to the TkBD problem in the radar context [17, 110, 111]. In contrast to fixed grid-based approaches that employ static samples, the PF utilises Monte Carlo random sampling.

In recent years, random finite sets have been also applied to the TkBD problem. The first application of RFS to tracking on image data was proposed in [130] based on a multi-Bernoulli filter. This was followed by applications to visual tracking at a cell by cell level [73] and to tracking ground moving targets with road constraints [151]. Extensions of the RFS to MIMO distributed networks [66] have also been considered. A comparison of RFS TkBD algorithms with other TkBD algorithms can be found in [152].

Another unique approach to TkBD is the Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT) [117, 120]. The H-PMHT algorithm is based on the application of Expectation-Maximisation (EM) data association to intensity map data and naturally returns an estimate of the target amplitude. This technique collates the measurement images as a histogram and assumes some underlying mixture density. Like the grid-based approaches, the H-PMHT is a natural multi-target algorithm and is efficient as it does not require the computation of likelihood ratios. The application of TkBD to extended object tracking has been proposed in [48, 150]. Research into the best theoretical mean square error bounds for TkBD can also be found in [92].

2.5.1 Forward-Backward Algorithm

The Forward-Backward algorithm is an optimal Bayesian smoother for discrete states that calculates the probability of a state at each time step given a sequence of observations using marginal posterior probabilities $p(\boldsymbol{x}_t|\mathsf{Z}_t)$ [105]. However this calculation does not produce the single best state sequence; instead the output is a sequence of states that are individually most likely, given all observations. This criterion maximises the number of correct individual states.

The Forward-Backward algorithm can be effectively regarded as a gradient descent algorithm searching for a minimum error.

The posterior marginal probabilities are calculated using two passes. The forward procedure advances in time and calculates the set of filtering probabilities that defines an initial estimate of the state. The backwards procedure computes a set of probabilities in reverse time and refines the estimates. The final state sequence is computed by combining the forwards and backwards variables to obtain the distribution of states at any specific point in time given the entire observation sequence. For a detailed review of the Forward-Backward algorithm, the reader is referred to Section 3.3.

The Forward-Backward algorithm accumulates probabilities over all paths to give an estimate of the state at each time. It is naturally suited to estimating multiple targets as it sums over all paths to any given state. At the final time step, the estimate will minimise all errors, however the path to the final state may violate the constraints imposed by the target dynamics model. In the next section, we review the Viterbi algorithm, which considers the joint distribution of the state sequence, rather than the marginals, and guarantees that the final solution will be a valid target path.

2.5.2 Viterbi Algorithm

Originally founded in speech recognition and communications, the Viterbi algorithm is a grid based approximation technique commonly used in tracking to compute Maximum Likelihood (ML) target trajectory estimates. It does so by making use of dynamic programming to derive efficient solution methods. In the context of tracking, the idea behind dynamic programming can be expressed as follows. If the optimal path between two points A and C is known to pass through point B, then the best path is the optimal path between A and B, followed by the optimal path from B to C. Based on this principle, a large problem can be effectively reduced to several smaller subproblems, and a more efficient implementation for an exhaustive search can be realised.

Unlike the Forward-Backward algorithm, which calculates a minimum variance estimator or a Maximum a Posteriori (MAP) estimator, the Viterbi algorithm produces a MAP probability sequence estimate. [105]. It does so calculating the best score over all paths that end in each state. By induction, the highest probability path to each state at the next time step can be calculated. To generate the final state sequence, a back-pointer array stores the state that maximised the cost function for every state each time step.

It is an example of a batch estimator as it recalculates the entire state sequence with every

new observation. One advantage of the Viterbi algorithm is that it guarantees that the final solution will be a valid target path as it considers the joint distribution of the state sequence. However, the Viterbi algorithm only returns the best path, hence all other paths that are close to optimal are disregarded.

The extension to multi-target tracking using dynamic programming can be challenging due to the additional computational complexity from the increased number of states. A multi-target dynamic programming for unknown number of targets is outlined in [74, 155].

Both the Viterbi and forward and backward algorithms are examples of optimal grid-based approximation techniques, which discretise the state space and calculate the Maximum Likelihood Estimator (MLE) of the parameters of a HMM, given a sequence of observations. The key difference between the two algorithm lies within their definition of an 'optimal' state sequence. The Forward-Backward algorithm is a Bayesian estimator that finds the sequence of states that are individually most likely using marginal posterior probabilities. On the other hand, the Viterbi algorithm is a dynamic programming method that finds the single most likely state sequence. For a detailed review of the Viterbi algorithm, the reader is referred to Section 3.4.1.

2.5.3 Likelihood Ratio Detection and Tracking

Another example of a Bayesian TkBD method based on a discretised grid is the Likelihood Ratio Detection and Tracking (LRDT) algorithm [116]. The LRDT algorithm proceeds by extending the state space to accommodate a null state to model the case when no target is present in the observation space. It forms a Bayesian recursion using likelihood ratios rather than target states, and cumulates measurement likelihood ratios over all possible target trajectories. The main benefit of this approach over other TkBD methods is that it allows information from multiple disparate sensors to be fused as it does not require explicit associations between sensors and tracks.

The LRDT assumes at most one target but it can be extended to the multi-target space by employing a bank of single independent trackers for scenarios with well separated targets. It has been successfully applied to at-sea sonar data, however its performance is dependent on accurate array localisation and estimates of background noise [116]. A variation of LRDT has been shown to give good performance in multi-sensor passive acoustic data [81].

2.5.4 Particle Filter

In this section, we introduce an alternate TkBD approach based on PFs. In the HMM approaches described in previous subsections, much of the computation is wasted by evaluating

the probability in the state pdf tails where there is little to negligible probability. The PF is similar to the Forward-Backward procedure as it also evaluates marginal posterior probabilities, however these calculations are performed in the filtering rather than smoothing context. Like the Forward-Backward algorithm, the PF is also a Bayesian estimator that finds the sequence of states that are individually most likely using marginal posterior probabilities.

PFs were first applied to the TkBD problem in [17, 111]. The TkBD PF algorithm proceeds by constructing two sets of particles; a set of birth particles and a set of continuing particles. The mixing proportions between the two sets are determined by the prior null state probability, probability of birth and probability of death. At the update stage, the null probability is calculated based on particle weights and the birth and continuing particles are combined to construct the pdf. Resampling is also carried out using uniform weights. A target is declared when the null probability falls below a predetermined threshold. Finally, state estimates are found by taking the conditional mean of state vectors over all particles.

A comparison of the performance of HMM methods with the TkBD PF for a single target scenario with varying speeds and SNR is presented in [39]. It was found that for most scenarios, all the algorithms gave similar performance, however the TkBD PF algorithm performed significantly faster as it uses substantially fewer sampling points than is required in a discrete grid.

2.5.5 Random Finite Sets

The extension of RFS theory to the TkBD problem is based on computing the posterior distribution of the RFS over the image observation. The result is an approximate solution to the Bayes multi-target filter [88, 89]. This approach is unique as it not only provides an estimate the current target states, but it also outputs an estimate of the number of targets. It does so by jointly estimating these variables from the measurement image.

The first application of RFS to tracking on image data was based on a MeMBer filter using a particle filter implementation [130]. The MeMBer filter is simply an approximation to the multi-target Bayes filter and can be modified to track on raw image data. It requires the assumption of a separable likelihood in which the distributions for the target and noise are assumed to be independent [157]. Unfortunately, this approach only outputs estimates of the state values at each time. An additional layer of track management is required to link the state values with time to give trajectory estimates. A solution to this issue for TkBD has been proposed through the idea of labelled RFS [100].

2.5.6 Histogram-Probabilistic Multi-Hypothesis Tracker

The Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT) was originally developed by Streit [117] in which he extended the Probabilistic Multi-Hypothesis Tracker (PMHT) algorithm to track on intensity modulated measurement data. The literature on the H-PMHT algorithm is limited and mostly restricted to linear Gaussian applications. However, in recent years the H-PMHT has been demonstrated with non-linear and non-Gaussian problems [35,36].

The H-PMHT is an attractive algorithm as it has been shown to give performance that is close to the optimal Bayesian filter at a fraction of computational cost [39]. The H-PMHT's unique definition of the measurement model provides data association weights that allows it to retain linear complexity with the number of targets.

The H-PMHT algorithm is based on the generation of a synthetic histogram by quantising the energy in the sensor data followed by the application of EM mixture modelling to describe the underlying data sources [86,140]. In the final step of the derivation, the limit of the quantisation is taken and the original sensor data is recovered. The quantisation step only appears in the derivation of the algorithm and not in its implementation. The interpretation of the quantized sensor image as a histogram and the subsequent use of PMHT data association led to the algorithm name. For a detailed review of the H-PMHT algorithm, the reader is referred to Chapter 4.

The H-PMHT can be applied to a wide range of problems as long as an appropriate state estimator exists to perform the maximisation step of the EM algorithm. The H-PMHT state estimation component of the EM procedure has been implemented using an EKF [35] and PF [36]. An implementation based on the Viterbi algorithm is derived for the H-PMHT in Chapter 4.

A problem with H-PMHT is that it fails to model fluctuating target amplitude. This limitation is addressed in Chapter 5, in which we consider an alternative measurement model based on a Poisson assumption on the quantised measurement counts. The new Poisson model is shown to be consistent with the original H-PMHT modelling assumptions but it now allows for an improved measure for track quality.

The process of taking the quantisation limit to zero also has other consequences when a Bayesian model is adopted: the infinite amount of synthetically generated data will overwhelm any prior. In Chapter 6, we propose a novel TkBD algorithm based on the application of EM data association to a new measurement model that directly describes continuous valued intensity maps and avoids using an intermediate quantisation stage like the H-PMHT.

Chapter 3

Maximum Likelihood Sequence Estimation for Hidden Reciprocal Processes

This chapter addresses the problem of maximum likelihood sequence estimation when a Hidden Reciprocal Process (HRP) is the underlying target model. Reciprocal Processes (RPs) are discrete-time stochastic processes that can be regarded as one-dimensional versions of a Markov random field (MRF), although they are not in general Markov processes. Unlike Markov processes that assume the target state at any given time is dependent only on its previous states, RPs are based on acausal processes. The key contribution of this chapter[1] is the first formal derivation of the maximum likelihood sequence estimator (MLSE) for a HRP in which the random variables are finite-state; this special case of a HRP is referred to as a Hidden Reciprocal Chain (HRC). The main contents of this chapter have also been published in journal article [149].

The chapter is arranged as follows: Section 3.1 discusses the background and motivation for an acausal target dynamics model in the context of the Track-Detect (TkBD) problem; Section 3.2 provides a general introduction to HRPs and discusses how they can be uniquely represented using Markov bridges; Section 3.3 reviews the procedure for computing an optimal smoother for a reciprocal target model based on the Markov bridge approach [147]; Section 3.4 proposes an extension of this method to evaluate the MLSE for a HRC; Section 3.5 verifies the state estimation performance of the new reciprocal-based MLSE through simulated scenarios featuring a

[1]Langford B. White, Professor of Telecommunications Engineering at the University of Adelaide is acknowledge here for his supervision of the work undertaken in this chapter.

reciprocal target trajectory. As the Hidden Markov Model (HMM) is an important model used in target tracking, the new estimator is compared to the equivalent MLSE for a HMM: the well known Viterbi Algorithm [7]. In addition, the performance of the new estimator is compared to the reciprocal-based optimal smoother. The key contributions of the chapter are summarised in Section 3.6.

3.1 Introduction

RPs can be regarded as one-dimensional versions of a Markov random field. Markov random fields are spatial processes and are defined as a set of random variables whose values only depend on the values of its nearest neighbours. RPs are also generalisation of Markov processes. Specifically, Markov models assume a causal process such that the target state at any given time is dependent only on its previous states. On the other hand, RPs are acausal processes where the target state at any given time is based on *both past and future* states.

Current techniques for target tracking predominantly assume a target dynamics model based on a Markov process. This approach is suitable for surveillance problems for which the

- observation period may be limited by the sensor's performance or by restrictions on the target resources,
- target intent and final destination are unknown.

However, in reality, some targets can travel over an extended period of time with a final destination that can potentially be clearly defined. In such a case, it is possible for the target's destination to be known prior to the final time point and the target state at any given time is dependent on both its past and future states. Thus it may be more appropriate to consider acausal target models such as RPs, which are capable of predicting termination points as well as tracking targets to their final destination.

An obvious example of an acausal target appears in the ship surveillance problem, in which information regarding a ship's movements is divided into two categories: sightings by other maritime platforms or satellites provide information about the ship's past movements, whereas information regarding refuelling stops provide information about the ship's future trajectory. Unlike Markov processes, which condition only on the past, RPs incorporate both categories of information by assuming a joint probability distribution on the initial and final states. The inclusion of target destination information in the target motion model by RPs allows *destination-aware tracking* to be performed [49, 50]. The ability to specify statistically dependent target

source and destination points can assist in mitigating common problems encountered in multi-target tracking. For example, RPs can potentially exploit destination information to distinguish targets of interest from benign targets for target classification purposes or resolve targets in a crossing target scenario.

The key contribution of this chapter is the *first formal derivation of the MLSE for a HRC*. The developments in this chapter show that, for certain scenarios, the replacement of the Markov model assumption with a target dynamics model such as the RP can result in improved state estimation performance.

A summary of the existing literature on RPs is provided in subsection 2.4. The generalisation of current HMM-based TkBD methods to include acausal assumptions via HRPs is relatively straight forward for fixed-grid approximation techniques. Recently, the problem of optimal fixed-interval smoothing for finite-state, discrete-time HRPs was addressed via the Markov bridge approach in [19, 146, 147]. A Markov bridge is defined as a Markov process conditioned on the initial and final states being known [55]. Note that the more widely known Brownian bridge is also a Markov process and can be considered the reference process for a Markov bridge [24, 109]. The authors combined Markovian target dynamics with a specified target source-destination distribution to yield a HRP model [77]. For the case when there are a finite number of states, denoted N_v, it was shown that a HRC can be represented by N_v Markov bridges, one corresponding to each of the N_v possible final states. This unique representation is used to derive the optimal fixed-interval smoother for a HRC and is similar to the Forward-Backward smoothers constructed for a HMM [147]. This chapter provides an extension to this work by utilising the Markov bridge approach to perform MLSE for a HRC.

3.2 Hidden Reciprocal Processes

Let $X_t, t = 0, \ldots, T$ for some integer $T > 2$ denote a sequence of random variables. Assume that each X_t takes values in the finite set $\mathcal{S} = \{1, \ldots, N_v\}$ for some finite integer $N_v \geq 2$. For a first order Markov process, the conditional probability distribution of X_t given all other values of X_s, where $s \neq t$, satisfies

$$p\left(X_t | X_s, \forall\ s \neq t\right) \quad = \quad p\left(X_t | X_{t-1}\right). \tag{3.1}$$

That is, the Markov process assumes that the target state at any given time is only dependent on its previous state. The Markov model is specified by the transition function (3.1) and an initial probability,

$$\Pi_i = p\left(X_0 = i\right). \tag{3.2}$$

The process X_t is said to be *reciprocal* if for each $t = 1, \ldots, T-1$ it satisfies

$$p\left(X_t | X_s, \forall\ s \neq t\right) \quad = \quad p\left(X_t | X_{t-1}, X_{t+1}\right), \tag{3.3}$$

i.e. RPs are one dimensional nearest neighbour processes where the conditional distribution of X_t depends only on its neighbours. In random field theory [65], this property is referred to as the "Markov" property. A RP can then be thought of as MRF with respect to the time parameter. In general, RPs are not necessarily Markov processes however, any Markov process is reciprocal [77]. The reciprocal model is specified by the transitions (3.3) and by the endpoints probability distribution,

$$\Pi_{i,k} = p\left(X_0 = i, X_T = k\right). \tag{3.4}$$

The term Reciprocal Chain (RC) will be used to specify that the random variables X_t are *finite-state*. The term 'hidden' will apply if the variables X_t cannot be observed directly. Instead a sequence of random observations Y_t, $t = 0, \ldots, T$ statistically dependent on X_t are available.

By fixing the endpoint of a RP, we generate what is commonly referred to as a Markov bridge [77]. Thus for the finite N_v state case, a RP can be thought to be equivalent to N_v Markov bridges, one corresponding to each of the possible final states taken by $X_T = k$, where $k = 1, \ldots, N_v$. For a given RC, let the three point transition function for $t = 1, \ldots, T-1$ be given by,

$$Q_{ijl}(t) = p(X_t = j | X_{t-1} = i, X_{t+1} = l). \tag{3.5}$$

In [147], the authors show how to construct the Markov bridge transitions corresponding to a specified three point transition function. First, define $B^k_{i,j}(t)$ to be the Markov bridge transitions that specify the probability of transiting from state i at time t to state j at time $t+1$, given a final destination of state k. A backward recursion [77] can be derived that fully specifies the set of N_v Markov bridge transitions for $T-2, \ldots, 0$ from the RC transition function Q_{ijl} such that,

$$\begin{aligned} B^k_{i,j}(t) \quad &= \quad p\left(X_{t+1} = j | X_t = i, X_T = k\right) \\ &= \quad \frac{Q_{ijl}(t+1)}{B^k_{j,l}(t+1)} \left(\sum_{m=1}^{N_v} \frac{Q_{iml}(t)}{B^k_{m,l}(t+1)} \right)^{-1}. \end{aligned} \tag{3.6}$$

Initialisation is with $B^k_{i,j}(T-1) = 1$ for $j = k$ and zero otherwise. This ensures that for the Markov bridge that terminates in state k at time T, the target at the penultimate time step is also in state k. Let π^k_i denote the initial probability distribution for the Markov bridge that

ends in state k. It is given by the conditional distribution,

$$\begin{aligned} \pi_i^k &= p\left(X_0 = i | X_T = k\right) \\ &= \frac{\Pi_{i,k}}{\sum_{j=1}^{N_v} \Pi_{j,k}}. \end{aligned} \tag{3.7}$$

It can be seen that any RC can be uniquely specified by a finite set of Markov bridges with transitions (3.6) and initial distributions (3.7).

3.3 Optimal Smoothing

This section discusses the procedure for computing an optimal smoother for a HRC using the Markov bridge approach [147], where the underlying RP is derived from a stationary Markov process. We wish to emphasise that the procedure for deriving the smoother is applicable to any RP without loss of generalisation. The more general case, where the underlying dynamics of the RP is not specified directly by a Markov transition is addressed in [147].

Before the optimal smoother for a HRC is presented, we first review the optimal smoother for a Hidden Markov Chain (HMC), that is, the well-known Forward-Backward algorithm [105].

3.3.1 Optimal smoothing for Hidden Markov Models: Forward-Backward Algorithm

The Forward-Backward algorithm is a discrete state technique and forms the optimal Bayesian smoother by calculating the probability of a state at time t given a sequence of observations using marginal posterior probabilities.

For a discrete state tracking problem, it is assumed that the target state space is broken up into $N_v \geq 2$ distinct cells where each cell corresponds to a state in a HMC. Let X_t be a deterministic mapping from index space into physical space. At any given time, our target of interest can occupy any one of the distinct states.

At regular discrete time points t, the system undergoes change in which the target can transition with some probability to another state or remain in its previous state. In the general case, the probabilistic motion of the system is described in terms of the current state at time t as well as all of its predecessor states. However for a first order discrete Markov process, it is sufficient to describe the system in terms of the current and previous state. That is, the probabilistic model is assumed to be independent of time with state transition probabilities,

$$A_{ij}(t) = P(X_t = j | X_{t-1} = i). \tag{3.8}$$

The Forward-Backward algorithm calculates the marginal probability of a state at time t given a sequence of observations $p(X_t = i|Y_0, \ldots, Y_t)$, it does not produce the single best state sequence. Instead the output is a sequence of states that are individually most likely, given all observations. The posterior marginal probabilities are calculated using two passes. The forward procedure advances in time and calculates the set of filtering probabilities that defines an initial estimate of the state. The backwards procedure computes a set of probabilities in reverse time and refines the estimates. First define $\alpha_t(X_t = j)$ as the forward variable, which is the probability of seeing observation sequence $Y_0, \ldots, Y_t$ and being in state i at time t. For $t = 0, \ldots, T$, the forward probabilities can be calculated recursively as follows,

$$\begin{aligned}
\alpha_t(X_t = i) &= p(X_t = i, Y_0, \ldots, Y_t) \\
&= \sum_{j=1}^{N_v} p(X_t = i, X_{t-1} = j, Y_0, \ldots, Y_t) \\
&= \sum_{j=1}^{N_v} p(X_t = i|X_{t-1} = j) P(X_{t-1} = j, Y_0, \ldots, Y_{t-1}) p(Y_t|X_t = i) \\
&= \sum_{j=1}^{N_v} A_{ji}\, \alpha_{t-1}(X_{t-1} = j) p(Y_t|X_t = i), \qquad (3.9)
\end{aligned}$$

where $p(Y_t|X_t = i)$ is the likelihood of observing measurement Y_t conditioned on being in state i at time t. Also, define the backward variable $\beta_t(X_t = i)$ as the probability of seeing observation sequence $Y_{t+1}, \ldots, Y_T$ conditioned on being in state i at time t. For $t = T - 1, \ldots, 0$, it can be calculated recursively as follows,

$$\begin{aligned}
\beta_t(X_t = i) &= P(Y_{t+1}, \ldots, Y_T|X_t = i) \\
&= \sum_{j=1}^{N_v} p(Y_{t+1}, \ldots, Y_T, X_{t+1} = j|X_t = i) \\
&= \sum_{j=1}^{N_v} p(X_{t+1} = j|X_t = i) p(Y_{t+2}, \ldots, Y_T|X_{t+1} = j) p(Y_{t+1}|X_{t+1} = j) \\
&= \sum_{j=1}^{N_v} A_{ij} \beta_{t+1}(X_{t+1} = j) p(Y_{t+1}|X_{t+1} = j), \qquad (3.10)
\end{aligned}$$

where $p(Y_{t+1}|X_{t+1} = j)$ is the likelihood of observing measurement Y_{t+1} conditioned on being in state j at time $t + 1$. The final state sequence is computed by combining the forward and backward variables to obtain the posterior probability of X_t given the entire observation sequence. After the normalisation, the posterior probabilities are given by,

$$p(X_t = i|, Y_0, \ldots, Y_T) = \frac{\alpha_t(X_t = i)\beta_t(X_t = i)}{\sum_{j=1}^{N_v} \alpha_t(X_t = j)\beta_t(X_t = j)}. \qquad (3.11)$$

To find the best state for $t = 1, \ldots, T$, the Maximum a Posteriori (MAP) estimator (i.e. finds the mode of the posterior distribution) can be calculated by choosing the state with the highest probability mass,

$$\hat{X}_t = \underset{i}{\operatorname{argmax}}\, p(X_t = i | Y_0, \ldots, Y_T). \tag{3.12}$$

Alternatively, the best state can also be calculated using the conditional mean estimate as follows,

$$\begin{aligned} \hat{X}_t &= E[X_t | Y_0, \ldots, Y_T] \\ &= \sum_{i=1}^{N_v} i\; p(X_t = i | Y_0, \ldots, Y_T). \end{aligned} \tag{3.13}$$

As we are approximating the continuous position and velocity state space by a discretised grid, the conditional mean is an appropriate state estimator. The forward procedure can be initialised using Π_i via (3.2) and the likelihood that the state is in i at the first time point. For the backward procedure, the probabilities are arbitrarily assigned to unity for all states, i.e. a uniform prior. A summary of the Forward-Backward algorithm is provided in Algorithm 1 on page 40.

3.3.2 Optimal Smoothing for Hidden Reciprocal Chains

In [147], White presents a generalisation of finite state optimal smoothing to HRCs, which is based on a modification of the Forward-Backward algorithm to allow for a reciprocal dynamics model. Consider the class of HRCs generated by a specified set of Markov transition matrices and a specified joint distribution on the endpoints. We now describe how we can obtain a Markov bridge from a general Markov chain $\{X_t\}$. Suppose we have the Markov transition probability matrix $A(t), t = 0, \ldots, T-1$, and an initial probability distribution Π_i (3.2). To generate a Markov bridge, we must ensure that the probability of the endpoint $X_T = k$ is unity for some state k. The resulting process is referred to as the (i, j, k) bridge derived from X_t. Using Bayes' rule, the Markov bridge transitions $B^k_{i,j}(t)$ can be expressed in terms of the Markov transitions $A_{i,j}(t)$,

$$\begin{aligned} B^k_{i,j}(t) &= p\left(X_{t+1} = j | X_t = i, X_T = k\right) \\ &= \frac{p\left(X_{t+1} = j, X_T = k | X_t = i\right)}{p\left(X_T = k | X_t = i\right)} \\ &= \frac{p\left(X_{t+1} = j | X_t = i\right)\; p\left(X_T = k | X_{t+1} = j\right)}{p\left(X_T = k | X_t = i\right)} \\ &= \frac{A_{i,j}(t)\; \Phi_{j,k}(t+1, T)}{\Phi_{i,k}(t, T)}\;, \end{aligned} \tag{3.14}$$

Algorithm 1 Forward-Backward Algorithm

1. Initialise forward and backward probabilities for all state X grid points $i = 1, \ldots, N_v$,

$$\begin{aligned} \alpha_0(X_0 = i) &= \Pi_i p(Y_0|X_0 = i) \\ \beta_T(X_0 = i) &= 1, \end{aligned}$$

   where Y_0 denotes the initial measurement.

2. Recurse forward in time for $t = 1, \ldots, T$,

   - For each state $i = 1, \ldots, N_v$, calculate forward probabilities,

$$\alpha_t(X_t = i) = \sum_{j=1}^{N_v} A_{ji}\, \alpha_{t-1}(X_{t-1} = j) p(Y_t|X_t = i),$$

     where Y_t denotes the measurement received at time t.

3. Recurse backward in time for $T - 1, \ldots, 0$,

   - For each state $i = 1, \ldots, N_v$, calculate backward probabilities,

$$\beta_t(X_t = i) = \sum_{j=1}^{N_v} A_{ij} \beta_{t+1}(X_{t+1} = j) p(Y_{t+1}|X_{t+1} = i)$$

4. Recurse forward in time for $t = 1, \ldots, T$, compute the conditional mean state estimate given the observations $Y_0, \ldots, Y_T$,

$$\hat{X}_t = \sum_{i=1}^{N_v} i\; p(X_t = i|Y_0, \ldots, Y_T),$$

   where

$$p(X_t = i|Y_0, \ldots, Y_T) = \frac{\alpha_t(X_t = i)\beta_t(X_t = i)}{\sum_{j=1}^{N_v} \alpha_t(X_t = j)\beta_t(X_t = j)}$$

where $\Phi(s,t)$ is defined for $s \leq t$ by,

$$\Phi(s,t) \;=\; \begin{cases} I & s = t \\ A(t-1)\,A(t-2)\,\cdots\,A(s) & s < t\ . \end{cases} \tag{3.15}$$

In order to construct a RC X_t, the endpoints $(X_0 = i, X_T = k)$ can be drawn from the endpoint distribution (3.4), and a Markov bridge starting at $X_0 = i$ can be constructed using the transition probabilities defined in (3.14). The equivalent three-point transition function for this process is specified by the Markov transitions $A(t)$ as

$$\begin{aligned} Q_{jik}(t) &= p\left(X_t = i | X_{t-1} = j, X_{t+1} = k\right) \\ &= \frac{A_{j,i}(t-1)\ A_{i,k}(t)}{\Phi_{j,k}(t-1, t+1)}\ . \end{aligned} \tag{3.16}$$

Jamison [77] proves that all RPs are attributed a three-point transition function of this form.

Consider a HRC that consists of the RC X_t and a set of observation variables $Y_t,\ t = 0, \ldots, T$, which are related through the following conditional independence property,

$$p\left(Y_0, \ldots, Y_T | X_0, \ldots, X_T\right) \;=\; \prod_{t=0}^{T} p\left(Y_t | X_t\right)\ . \tag{3.17}$$

The same assumption is made in a HMC. Assume that the conditional likelihoods $p\left(Y_t | X_t = i\right)$ are well-defined and can be evaluated.

To compute the optimal smoother for a HRC, define the forward procedure $\alpha_i^k(t)$ to be the joint probability of observing the sequence $Y_0, \ldots, Y_t$ and being in state i at time t. However under the reciprocal model, we fix the endpoint so that the forward probabilities are also conditioned on being in state k at final time T. For the k^{th} Markov bridge, it can be calculated recursively as follows,

$$\begin{aligned} \alpha_i^k(t) &= p(X_t = i, Y_0, \ldots, Y_t | X_T = k) \\ &= \sum_{j=1}^{N_v} p(X_t = i, X_{t-1} = j, Y_0, \ldots, Y_t | X_T = k) \\ &= \sum_{j=1}^{N_v} p(X_t = i | X_{t-1} = j, X_T = k)\ p(X_{t-1} = j, Y_0, \ldots, Y_{t-1} | X_T = k)\ p(Y_t | X_t = i) \\ &= \sum_{j=1}^{N_v} B_{ji}^k(t-1)\ \alpha_j^k(t-1)\ p(Y_t | X_t = i), \end{aligned} \tag{3.18}$$

for $t = 1, \ldots, T-1$. Above, we have applied the property that a RC terminating at the endpoint $X_T = k$ is equivalent to a Markov bridge with three point transition probabilities B_{ji}^k and initial distribution π_i^k given by (3.7).

For the HRC backward procedure, define $\beta_i^k(t)$ to be the joint probability of observing the sequence $Y_{t+1}, \ldots, Y_T$, conditioned on being in state i at time t *and* having a final destination $X_T = k$. For the k^{th} Markov bridge, the backward procedure can be calculated recursively as follows,

$$\begin{aligned}
\beta_i^k(t) &= p(Y_{t+1}, \ldots, Y_T | X_t = i, X_T = k) \\
&= \sum_{j=1}^{N_v} p(Y_{t+1}, \ldots, Y_T, X_{t+1} = j | X_t = i, X_T = k) \\
&= \sum_{j=1}^{N_v} p(X_{t+1} = j | X_t = i, X_T = k)\ p(Y_{t+2}, \ldots, Y_T | X_{t+1} = j, X_T = k)\ p(Y_{t+1} | X_{t+1} = j) \\
&= \sum_{j=1}^{N_v} B_{ij}^k(t)\ \beta_j^k(t+1)\ p(Y_{t+1} | X_{t+1} = j), \qquad (3.19)
\end{aligned}$$

for $t = T-2, \ldots, 0$. As per the Forward-Backward algorithm, the final state sequence is computed by combining the forwards and backwards variables to compute the posterior probabilities for the Markov bridge ending in $X_T = k$,

$$\begin{aligned}
p(X_t = i | Y_0, \ldots, Y_T, X_T = k) &\propto \alpha_i^k(t) \beta_i^k(t) \\
&= \delta_i^k(t). \qquad (3.20)
\end{aligned}$$

It can be seen that the optimal smoother for a HRC is equivalent to forming N_v Markov bridge smoothers, each corresponding to termination point in state k. Bayes' rules allows us to combine the N_v bridges by marginalising over the final state distribution $P(X_T = k)$ and thus the posterior probabilities are given by,

$$\begin{aligned}
p(X_t = i | Y_0, \ldots, Y_T) &\propto \sum_{k=1}^{N_v} \alpha_i^k(t) \beta_i^k(t)\ p(X_T = k) \\
&= \delta_i(t), \qquad (3.21)
\end{aligned}$$

where $p(X_T = k)$ is the marginal obtained from the joint endpoints distribution given in (3.4). A conditional mean of the state estimates can be obtained in the standard way (3.13) by normalising the probabilities such that,

$$p(X_t = i | Y_0, \ldots, Y_T) = \frac{\delta_i(t)}{\sum_{j=1}^{N_v} \delta_j(t)}. \qquad (3.22)$$

The forward and backward probabilities can be initialised as following,

$$\begin{aligned} \alpha_i^k(0) &= p(X_0 = i | X_T = k) \\ &= \pi_i^k P(Y_0 | X_0 = i), \end{aligned} \tag{3.23}$$

$$\begin{aligned} \beta_i^k(T-1) &= p(Y_T | X_{T-1} = i, X_T = k) \\ &= p(Y_T | X_T = k), \end{aligned} \tag{3.24}$$

where π_i^k is the Markov bridge endpoints distribution defined in (3.7) and let $\beta_i^k(T) = 1$ for states i and Markov bridges k. Note that $\beta_i^k(T-1)$ is identical for $i = 1, \ldots, N_v$. As the HRC optimal smoother iterates over all Markov bridges terminating in state k, computational complexity is $O(N_v^3 T)$ compared with $O(N_v^2 T)$ for the HMC optimal smoother. The procedure for computing the optimal smoother for a HRC is summarised in Algorithm 2 on page 44.

3.4 Maximum Likelihood Sequence Estimation

In this section, the procedure for computing the MLSE for a HRC target model is outlined assuming that the underlying RC is again derived from a stationary Markov process. The method for computing the MLSE for a HRC target model is the key contribution for this chapter. Before we outline the procedure, we first review the MLSE for a HMC, namely the well known Viterbi algorithm.

3.4.1 MLSE for Hidden Markov Models: Viterbi Algorithm

Like the Forward-Backward algorithm, the Viterbi algorithm is also a discrete state technique and is commonly used in tracking to compute maximum likelihood (ML) target trajectory estimates. It does so by making use of dynamic programming to derive efficient solution methods. In the context of tracking, the idea behind dynamic programming can be expressed as follows. If the optimal path between two points A and C is known to pass through point B, then the best path is the optimal path between A and B, followed by the optimal path from B to C. Based on this principle, a large problem can be effectively reduced to several smaller subproblems, and a more efficient implementation for an exhaustive search can be realised.

Unlike the Forward-Backward algorithm, which calculates a minimum variance estimator or a MAP estimator, the Viterbi algorithm produces a MAP probability sequence estimate. It does so by finding the single best state sequence $X_0, \ldots X_T$, given a set of noisy observations $Y_0, \ldots, Y_T$. The other key difference between the Viterbi algorithm and other filtering methods such as the

Algorithm 2 Optimal Smoother for HRC

1. Initialise forward and backward probabilities for all state X grid points $i = 1, \ldots, N_v$,

$$\begin{aligned} \alpha_i^k(0) &= \pi_i^k p(Y_0|X_0 = i) \\ \beta_i^k(T) &= 1 \\ \beta_i^k(T-1) &= p(Y_T|X_T = k), \end{aligned}$$

   where Y_0 and Y_T denote the initial and final measurement, respectively.

2. Recurse forward in time for $t = 1, \ldots, T-1$,

   - For each state $j = 1, \ldots, N_v$ and Markov bridge ending in state $X_T = k$, calculate forward probabilities,

$$\alpha_i^k(t) = \sum_{j=1}^{N_v} B_{ji}^k(t-1)\ \alpha_j^k(t-1)\ p(Y_t|X_t = i),$$

     where Y_t denotes the measurement received at time t.

3. Recurse backward in time for $T-2, \ldots, 0$,

   - For each state $j = 1, \ldots, N_v$ and Markov bridge ending in state $X_T = k$, calculate backward probabilities,

$$\beta_i^k(t) = \sum_{j=1}^{N_v} B_{ij}^k(t)\ \beta_j^k(t+1)\ p(Y_{t+1}|X_{t+1} = j)$$

4. Recurse forward in time for $t = 1, \ldots, T$, compute the conditional mean state estimate given the observations $Y_0, \ldots, Y_T$,

$$\hat{X}_t = \sum_{i=1}^{N_v} i\ p(X_t = i|Y_0, \ldots, Y_T),$$

   where

$$p(X_t = i|Y_0, \ldots, Y_T) = \frac{\delta_i(t)}{\sum_{j=1}^{N_v} \delta_j(t)}$$

$$\delta_i(t) = \sum_{k=1}^{N_v} \alpha_i^k(t)\beta_i^k(t)p(X_T = k)$$

Forward-Backward algorithm lies within their definition of an 'optimal' state sequence. Filters are causal estimators that find the states that are individually most likely using marginal posterior probabilities. On the other hand, the Viterbi algorithm is a dynamic programming method that finds the single most likely state sequence.

In order to find the most likely state sequence, first define the quantity $C_t(X_t)$ to be the best score over all paths that end in state i from time $0, \ldots, t$ as

$$C_t(X_t = i) = \max_{X_0,\ldots,X_{t-1}} P(X_0, \ldots, X_t = i, Y_0, \ldots, Y_t). \tag{3.25}$$

By induction, the highest probability path can be calculated at the next time step for state $X_{t+1} = j$ as follows,

$$C_{t+1}(X_{t+1} = j) = \max_i \left\{ C_t(X_t = i)\ A_{ij}(t) \right\} P(Y_{t+1}|X_{t+1} = j). \tag{3.26}$$

To generate the final state sequence, a back-pointer, denoted by $\theta_t(X_t)$, is required that stores the state i that maximised equation (3.26) for each time step. The cost functions $C_0(X_t = i)$ can be initialised using the initial distributions (3.2) and the likelihood at the first time point $p(Y_0|X_0 = i)$ for states $i = 1, \ldots, N_v$.

The Viterbi algorithm calculates the probability of a partial state sequence at any given time, while keeping a back-pointer θ_t indicating how the current state could be reached. It is similar in implementation to the forward procedure of the Forward-Backward algorithm except that it replaces the summation over previous states with a maximisation. The Viterbi algorithm also has the additional backtracking step to find the overall best sequence of states. One advantage of the Viterbi algorithm is that it guarantees that the final solution will be a valid target path as it considers the joint distribution of the state sequence. However, the Viterbi algorithm only returns the best path, hence all other paths that are close to optimal are disregarded. Thus the Viterbi algorithm is naturally a single target tracker. For multi-target tracking, the Viterbi algorithm can be modified to return either the M best paths or all paths that fall above some threshold set by the user. However, note that these M best paths may not necessarily correspond to good tracks for the M targets; several paths may differ by only one or two points and may be redundant estimates for a single target. A summary of the Viterbi algorithm for single target tracking is provided in Algorithm 3 on page 46.

Unfortunately, the probability of observing any particular state sequence can be very small due to the size of the discretised Viterbi grid. As a result, the recursive computation of these very small conditional probabilities can potentially lead to numerical stability problems. To avoid these issues, an implementation of the Viterbi algorithm using logarithms of the conditional probabilities is used here [105]. In the next section, we describe how to calculate the MLSE for

a HRC. This procedure is based on a modification of the Viterbi algorithm to accommodate a reciprocal target model.

Algorithm 3 Viterbi Algorithm

1. Initialise the scoring quantities and the back-pointer array for all state X grid points $i = 1, \ldots, N_v$,

$$\begin{aligned} C_0(X_0 = i) &= \Pi_i p(Y_0|X_0 = i) \\ \theta_0(X_0 = i) &= 0, \end{aligned}$$

   where Y_0 denotes the initial measurement.

2. Recurse forward for time points $t = 1, \ldots, T$,

   - For each state j, calculate $C_t(X_t = j)$, the cost to transition to state from state i at time $t-1$ to state j at time t. Store the state i that maximised the cost function in back-point array $\theta_t(X_t) = j$,

$$\begin{aligned} C_t(X_t = j) &= \max_i \left\{ C_{t-1}(X_{t-1} = i)\ A_{ij}(t) \right\} P(Y_t|X_t = j) \\ \theta_t(X_t = j) &= \operatorname*{argmax}_i \left\{ C_{t-1}(X_{t-1} = i)\ A_{ij}(t) \right\}, \end{aligned}$$

     where Y_t denotes the measurement received at time t.

3. On completion, the state estimate at time T is found by maximising C_T, and the path to the final state is traced back to the initial time via back-pointers stored in θ_t.

$$\begin{aligned} \hat{X}_T &= \operatorname*{argmax}_i\ C_T(X_T = i) \\ \hat{X}_t &= \theta_{t+1}(\hat{X}_{t+1}) \quad \text{for } t = T-1, \ldots, 1. \end{aligned}$$

3.4.2 MLSE for Hidden Reciprocal Chains

In Section 3.3.2, optimal smoothers were utilised to estimate the hidden states of a RC using the Markov bridge approach [147], where the underlying RC was derived from a stationary Markov process. In this section, we show that it is relatively straight forward to extend this approach to calculate the MLSE for a HRC, which is the MAP estimate derived from the joint probability mass function of the entire sequence. That is,

$$\max_{X_0,\ldots,X_{T-1}} p\left(Y_0,\ldots,Y_T,X_0,\ldots,X_T=k\right), \tag{3.27}$$

which is exactly the expression evaluated by the Viterbi algorithm, only pinned at the final time point.

Again, let us assume that the hidden states of the RC are derived from a stationary Markov process with transition matrix $A(t)$. We also again make use of the knowledge that a HRC can be uniquely represented by N_v Markov bridges coupled with an initial joint probability on the endpoints.

Define the quantity $\delta_i^k(t)$ to be the best score over all paths that end in state i at time t, given that our final destination at time T is state k,

$$\delta_i^k(t) \quad = \quad \max_{X_0,\ldots,X_{t-1}} p\left(Y_0,\ldots Y_t,X_0,\ldots,X_{t-1},X_t=i|X_T=k\right). \tag{3.28}$$

The highest probability path at the next time step in the HRC chain $X_{t+1}=j$ can be calculated by utilising the Markov bridge property and the conditional likelihoods, as in HMC dynamic programming,

$$\delta_j^k(t+1) \quad = \quad \max_i \left\{B_{i,j}^k(t)\ \delta_i^k(t)\right\} p\left(Y_{t+1}|X_{t+1}=j\right), \tag{3.29}$$

where $B_{i,j}^k(t)$ is the Markov bridge transitions (3.14) that have been derived from a Markov process with transition matrix $A(t)$. Define a back-pointer $\psi_j^k(t+1)$, which stores the maximising previous state X_t, for each time step,

$$\psi_j^k(t+1) \quad = \quad \arg\max_i \left\{B_{i,j}^k(t)\ \delta_i^k(t)\right\}. \tag{3.30}$$

It can be seen that the MLSE calculates the probability of a partial state sequence at any given time, while keeping a back-pointer indicating how the current state could be reached.

Initialisation of the algorithm is performed at time $t=0$ using the conditional likelihood and endpoint distribution (3.4),

$$\delta_i^k(0) \quad = \quad p\left(X_0=i|X_T=k\right)p\left(Y_0|X_0=i\right). \tag{3.31}$$

The MLSE will then implicitly search through all potential state sequences before providing the best final sequence estimate. At final time T, we have the following quantity,

$$\delta_i^k(T) \quad = \quad \max_{X_0,\ldots,X_{T-1}} p\left(Y_0,\ldots,Y_T,X_0,\ldots,X_{T-1},X_T=i|X_T=k\right). \tag{3.32}$$

Note that the above quantity is zero except when $i=k$. When this occurs, we have

$$\delta_k^k(T) \quad = \quad \max_{X_0,\ldots,X_{T-1}} p\left(Y_0,\ldots,Y_T,X_0,\ldots,X_{T-1}|X_T=k\right). \tag{3.33}$$

Therefore the MAP estimates can be calculated using

$$\max_{X_0,\ldots,X_T} p\left(Y_0,\ldots,Y_T,X_0,\ldots,X_{T-1},X_T=k\right)=\delta_k^k(T)\;p\left(X_T=k\right), \tag{3.34}$$

where $p\left(X_T=k\right)$ is determined from the marginal of the joint endpoints distribution (3.4). Thus, the most likely value for X_T can be found by first maximising across all N_v Markov bridges,

$$\hat{X}_T=\arg\max_k\;\left\{\delta_k^k(T)\;p\left(X_T=k\right)\right\}. \tag{3.35}$$

Finally, the estimates for the entire track for the HRC MLSE are generated in the normal dynamic programming way. That is, the path to the final state is traced back to the initial time via back-pointers stored in $\psi_i^k(t+1)$,

$$\hat{X}_t=\psi_{\hat{X}_{t+1}}^{\hat{X}_T}(t+1). \tag{3.36}$$

Again, the Viterbi algorithm can be modified to perform multi-target tracking by returning either the M best paths or all paths that fall above some existence threshold set by the user. Like the HRC optimal smoother, computational complexity is also $O(N_v^3T)$, compared with $O(N_v^2T)$ for the Viterbi algorithm. The procedure for calculating the MLSE for a HRC is summarised in Algorithm 4.

3.5 Simulations

This section verifies the performance of the proposed HRC MLSE through two sets of simulations. We show that the HRC MLSE gives improved state estimation performance over the Viterbi algorithm for scenarios featuring a reciprocal target. This constitutes one of the key contributions of this chapter. In the first set of simulations, the HRC MLSE is used to track a reciprocal target for which the underlying RC is derived from a stationary Markov process. Two scenarios are considered in which the distribution on the reciprocal target endpoints is varied. We perform Monte Carlo simulations and calculate MLSE state estimation errors under both HMC and HRC estimators. In the second set of simulations, three scenarios are considered in which the target trajectories are progressively made to behave in a more reciprocal manner. The relative gains in performance between the HMC and HRC MLSEs are calculated for each of the three scenarios.

The performance of the Markov and reciprocal MLSE estimators is also compared to their optimal smoother counterparts. In the case of the HMC, the optimal smoother is the Forward-Backward algorithm. Hence in each set of simulations, the following four estimators were implemented:

Algorithm 4 MLSE for HRC Algorithm

1. Initialise the scoring quantities and back-pointer array for all state grid points $i = 1, \ldots, N_v$, and for all endpoint destinations $k = 1, \ldots, N_v$,

$$\begin{aligned} \delta_i^k(0) &= p\left(X_0 = i | X_T = k\right) p\left(Y_0 | X_0 = i\right), \\ \psi_i^k(0) &= 0, \end{aligned}$$

   where Y_0 denotes the initial measurement.

2. Recurse forward for time points $t = 1, \ldots, T$,

   - For each Markov bridge $k = 1, \ldots, N_v$,

     – For each state j, calculate $\delta_j^k(t)$, the cost to transition to state from state i at time $t-1$ to state j at time t, given that the final destination is state k. Store the state i that maximised the cost function in back-point array $\psi_j^k(t)$,

$$\begin{aligned} \delta_j^k(t) &= \max_i \left\{ B_{i,j}^k(t-1)\ \delta_i^k(t-1) \right\} p(Y_t | X_t = j), \\ \psi_j^k(t) &= \arg\max_i \left\{ B_{i,j}^k(t-1)\ \delta_i^k(t-1) \right\}, \end{aligned}$$

       where Y_t denotes the measurement received at time t.

3. The state estimate at time T is found by maximising across all N_v Markov bridge to find the best Markov bridge k. The path to the final state is traced back to the initial time via back-pointers stored in $\psi_{\hat{X}_t}^{\hat{X}_T}(t)$ for Markov bridge ending in state k,

$$\begin{aligned} \hat{X}_T &= = \arg\max_k \left\{ \delta_k^k(T)\ p\left(X_T = k\right) \right\}, \\ \hat{X}_t &= \psi_{\hat{X}_t}^{\hat{X}_T}(t) \quad \text{for } t = T-1, \ldots, 1. \end{aligned}$$

- **HMC MLSE**: Maximum likelihood sequence estimate assuming a Markov target model (Viterbi Algorithm).
- **HMC Optimal**: Optimal smoother assuming a Markov target model Forward-Backward Algorithm).
- **HRC MLSE**: Maximum likelihood sequence estimate assuming a reciprocal target model (new method derived in this chapter).
- **HRC Optimal**: Optimal smoother assuming a reciprocal target model (see [147] for details).

In the first set of simulations, the problem of detecting a reciprocal target moving in a one-dimensional space for which the underlying RC is derived from a stationary Markov process is considered. Two scenarios featuring different assumptions on the reciprocal endpoint distributions are considered. We reiterate that the procedure for deriving the reciprocal estimators via Markov Bridges is applicable for any RP without loss of generalisation.

The reciprocal target chain was derived from the (stationary) Markov transition matrix:

$$A_{i,j} = \begin{cases} \frac{1}{3}, & \text{for } j = i, i-1, \text{and } i+1, \\ 0, & \text{otherwise,} \end{cases}$$

that is, a target is only allowed to transition to adjacent states or remain in its current state with equal probability. In the case when the target resides in the endpoints i.e. $i = 1$ or N_v, the target will remain in its current state or transition to its single neighbouring state, both with probability $\frac{1}{2}$. An example of the transition probabilities from state 5 (blue line) and from the endpoint state 1 (red line) for all time scans are shown in Figure 3.1. The observations $Y_t \sim \mathcal{N}(X_t, \sigma^2)$ were chosen to be independent Gaussian random variables with mean given by the state value (i.e., $1, 2, \ldots, N_v$) and constant noise variance σ^2. All simulations results were averaged over 10,000 Monte Carlo runs and time scans T, for a total number of states $N_v = 10$.

Recall that the Markov bridge transition matrix formulation requires that if a target terminates in state k, the target state at the penultimate time step has to also be in state k. At the first time scan, an initial target state i with uniform probability is selected. As the target is only able to transition at most to adjacent states at each time step, this implies that the target needs a minimum number of time steps to ensure that all combinations of the initial and termination points (i, k) are possible under the simulation. That is, a target that starts in state $i = 1$ at $t = 1$ requires $T = N_v + 1$ time steps to be able to terminate in state $k = N_v$, in order to satisfy the condition that the target at time $T - 1$ must also be in state k. As we have set $N_v = 10$ for the first set of simulations, the sequence length will be $T = 11$.

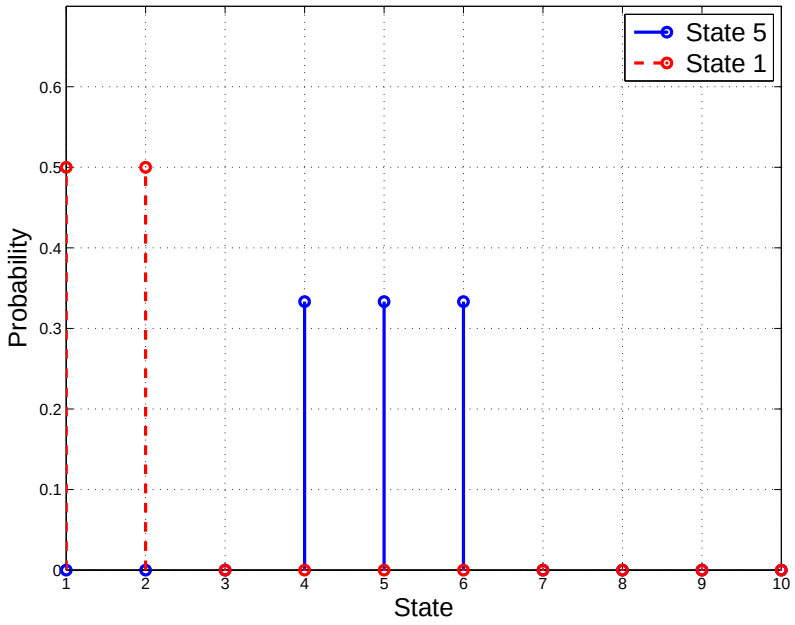
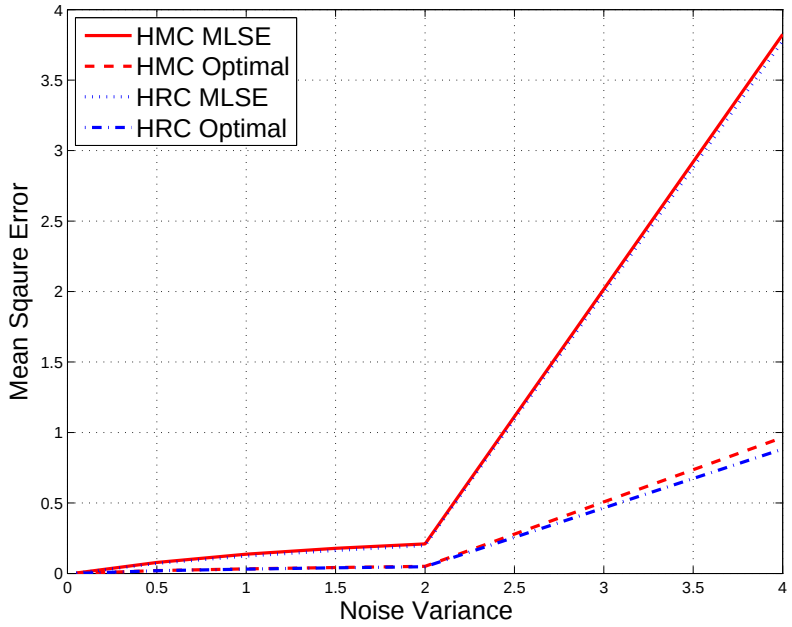


Figure 3.1: Example of Markov transition probabilities from state 5 (blue) and from state 1. Assumes number of states $N_v = 10$.

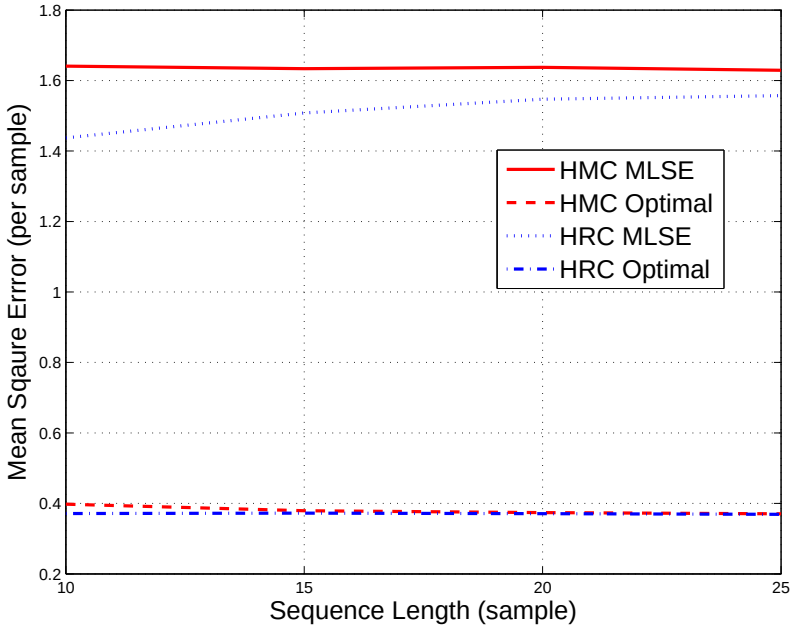
In the first scenario, a reciprocal target with an unconstrained source and destination was generated by assuming a uniform joint distribution on the endpoints. Hence for any given start point, the target destination point can end up in any state in the discrete grid.

Figure 3.2 shows the calculated Mean Square Estimation Error (MMSE) averaged over time for the uniform endpoints scenario plotted against (a) measurement noise variance σ^2 and (b) sequence length T. Observe that as the minimum variance estimators, the optimal smoothers give superior performance over their corresponding MLSEs. As expected, the error for all four estimators increase with measurement noise variance, however the MLSE error seems to be more sensitive to the measurement noise variance than the optimal smoother, as can be observed particularly for noise variance $\sigma^2 = 4$. In Figure 3.2.(b), observe that the performance of the estimators remains relatively constant with sample length T, particularly for sequence lengths longer than $T = 15$. This implies that as T increases, the error performance becomes independent of the batch length T.

It is expected that as the target truth follows a reciprocal dynamics model, the reciprocal estimators should outperform the traditional Markov estimators. However, it is evident that there is minimal improvement in performance by the reciprocal estimators. This may be due to the assumption of a uniform distribution on the endpoints, which could result in a large proportion of reciprocal realisations that appear very similar to a Markov chain. Hence in this



(a) Mean square estimation error vs. measurement noise variance σ^2 for sequence length $T = 12$.



(b) Mean square estimation error vs. sequence length T for measurement noise variance $\sigma^2 = 1$.

Figure 3.2: Comparison of mean square state estimation error for HRC and HMC MLSEs and optimal smoothers for the uniform endpoints scenario. Assumes number of states $N_v = 10$.

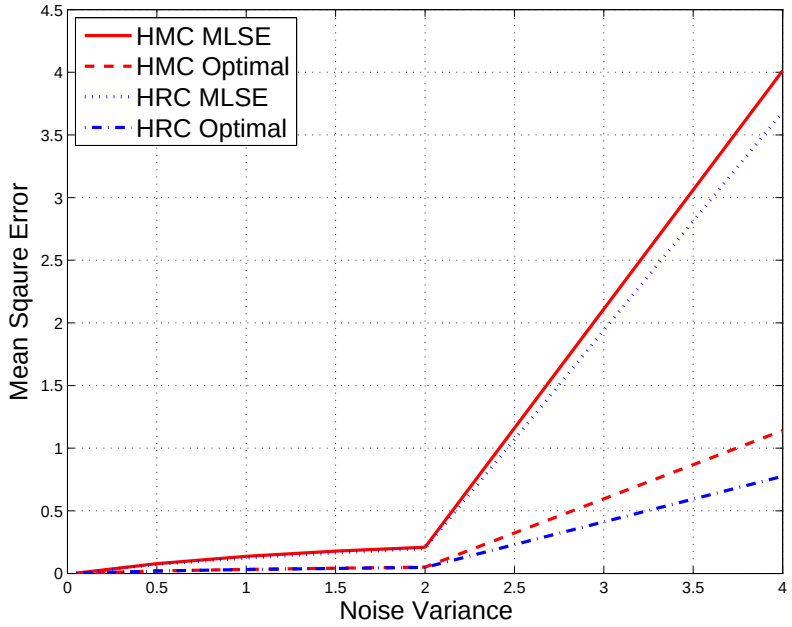
case, state estimation performance via a reciprocal estimator would provide limited benefit.

In order to verify this, a second scenario was considered in which the RC boundary distribution was specified to emphasise the source-destination pairs $(i, N_v - i + 1)$ for $i = 1, \ldots, N_v$. The initial distribution π_0 was chosen to be uniform and the conditional distribution $p\left(X_T = i | X_0 = j\right)$ was set to zero unless $j = N_v - i + 1$, in which case it was unity. The boundary distribution specified by Bayes' rule has been chosen to emphasise realisations of the RC in which the set of possible endpoints are markedly different from the endpoints that could possibly be generated by a Markov transition. This deliberate design choice was done in order to maximise the difference between the reciprocal and Markov estimators. Effectively, we can think of this second target model as being more "informative" than in the first scenario, as the reciprocal estimator has been provided with additional information regarding target destination.

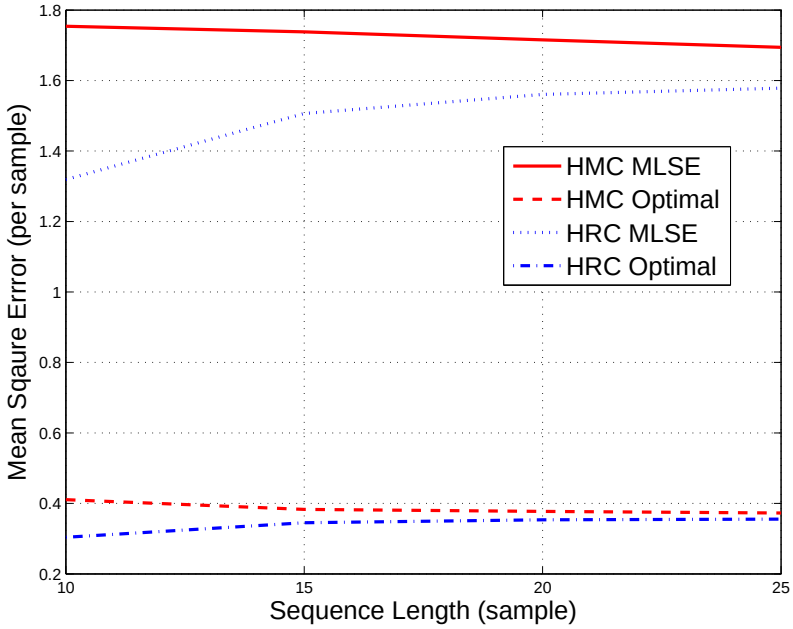
Figure 3.3 shows a comparison of state estimation performance for different values of measurement noise variance σ^2 and sequence length T for the informative endpoints scenario. Again, the optimal estimators give superior performance over the MLSEs and the slope of the MLSE error with noise variance is much greater than the slope for the optimal smoother. However, observe that the reciprocal estimators now outperform their Markov counterparts for all noise variances and sequence lengths. As per the first scenario, the largest gains in performance are seen at shorter sequence lengths. Observe that in Figure 3.3 (b), as the sequence length T increases, the performance of the Markov optimal estimator begins to converge to the reciprocal optimal estimator. This convergence behaviour is not as obvious in the reciprocal MLSE. Figure 3.4 shows the results of extending the sequence length in Figure 3.3 (b) from 25 scans to a total of 100 scans. It is clear that the performance of the reciprocal MLSE converges to the Markov MLSE if the batch length is sufficiently long.

The correlation between state estimation performance and sequence length can be attributed to the fact that the underlying reciprocal target dynamics in the simulations was derived from a stationary Markov process. It is reasonable to expect that as both processes near their destination points, the RC will become more distinct from its underlying Markov process. That is, the RP will look very similar to its derived Markov process except as it nears the final time T due to the additional information contained in the joint probability endpoints distribution. The proportion of time that the two targets exhibit similar behavior increases as T becomes larger, thus the reciprocal estimators will always converge to its Markov counterpart given the batch length is sufficiently long. On the other hand, when the batch length is kept relatively short (i.e. $T < 20$), the reciprocal estimators will consistently outperform their Markov counterparts, as the RP behaves less like a Markov process at short sequence lengths.

The results from these simulations seem to suggest that for scenarios featuring a more infor-



(a) Mean square estimation error vs. measurement noise variance σ^2 for sequence length $T = 12$.



(b) Mean square estimation error vs. sequence length T for measurement noise variance $\sigma^2 = 1$.

Figure 3.3: Comparison of mean square state estimation error for HRC and HMC MLSEs and optimal smoothers for the informative endpoints scenario. Assumes number of states $N_v = 10$.

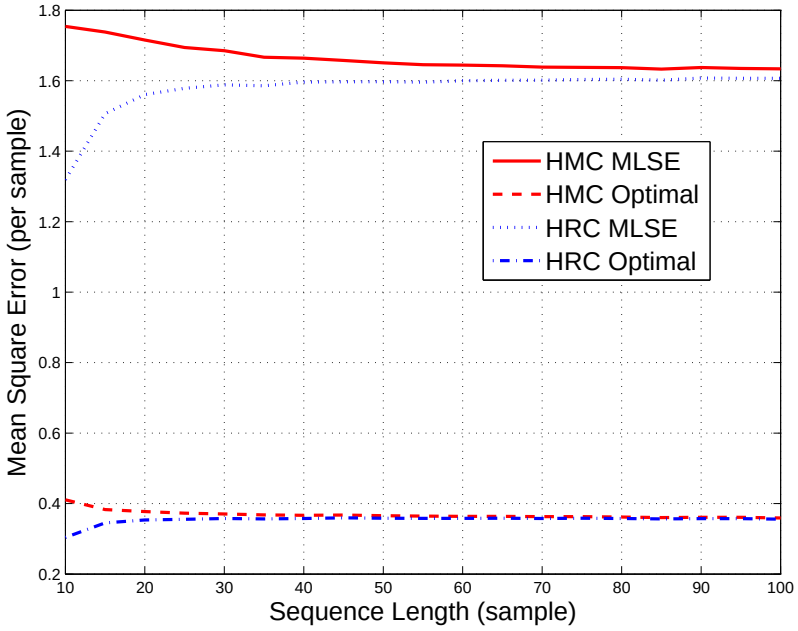


Figure 3.4: Mean square state estimation error vs. sequence length extended to $T = 100$ for the informative endpoints scenario. Assumes number of states $N_v = 10$, measurement noise variance $\sigma^2 = 1$.

mative target model (as seen in the second scenario), gains in estimation performance can be achieved by assuming a reciprocal target model rather than the traditional Markov dynamics. It is also evident that the MLSE does not perform as well as the optimal smoothers and is more sensitive to changes in noise variance in both the reciprocal and Markov cases. Under high measurement noise, the MLSE and optimal smoother have the potential to be seduced away from truth. As the MLSE is required to return a valid sequence under the target dynamics model, it will struggle to recover from large deviations from the truth as the target model only allows at most, a transition to adjacent states at each time step. On the other hand, the optimal smoother has no requirement for a valid path and has superior performance when noise variance is increased.

Also, by definition, the optimal smoother provides the minimum variance estimate whereas MLSE generates a MAP estimate, thus limiting its performance to the resolution of its state grid. For these simulations, the truth is discrete, therefore errors of magnitude less than the grid resolution will have minimal impact on the MLSE performance. However as the noise variance increases, the MLSE error can become larger than the grid resolution and can be magnified due to rounding. The magnitude of these quantisation errors will be highly dependent on the resolution of the state space. Despite this, the MLSE remains an attractive algorithm as the optimal smoother cannot be used for some applications, as it returns target estimates that

can be invalid under the discrete state space or state sequences that violate the target motion model.

Although the second scenario used trajectories where the reciprocal model was more informative, the error statistics were averaged over different trajectories where the most informative cases were mixed with the least. Three different cases are now considered to highlight when the use of a reciprocal model provides a low, medium and high amount of information gain. For the three cases, the target trajectories are selected to progressively appear less like a Markov process. For these simulations, the total number of finite states is extended to $N_v = 20$, to remove edging effects for target transitions near the end points. The exact paths for the three cases are shown in Figure 3.5. The target trajectory in each case was selected to behave according to the following dynamics models:

- **Markov**: In the first case, consider a target that follows a stationary Markov dynamics model (pink line) such that the trajectory starts and terminates in state 5.
- **Intermediate**: The second trajectory (cyan line) was chosen to mimic a target behaving in a more reciprocal manner, thus the target starts in state 5 and terminates in state 10.
- **Reciprocal**: In the final case, consider a target that follows a reciprocal dynamics model with a trajectory that again starts in state 5 but ends in state 15 (black line).

To correlate the 'reciprocalness' of each trajectory with state estimation performance, simulations using 10,000 Monte Carlo runs with randomised measurements were considered for each case. State estimation is performed using MLSE and optimal smoothing under both Markov and reciprocal target models where the Markov transitions are given by (3.37) and the RC transitions are again assumed to be derived from the Markov model. Define the relative goodness of fit under each model by the following ratio

$$F = \frac{MSE_{_M} - MSE_{_R}}{MSE_{_M}} \times 100, \tag{3.37}$$

where $MSE_{_M}$ and $MSE_{_R}$ are the mean square estimation errors for the HMC and HRC estimators respectively. This ratio describes the relative improvement in the HRP mean square error, where the larger the ratio, the better the fit under the HRP target model.

Figures 3.6 shows the goodness of fit when a HRC model rather than a HMC is assumed for each of the three scenarios using MLSE. Figure 3.7 shows the same relationship using optimal smoothers. Both figures show the results for varying measurement noise variance $\sigma^2 = 0.05, 0.5, 1, 1.5$ and 2. For low measurement noise, observe that the goodness of fit under the HRC model improves significantly for both the MLSE and optimal smoothers, with a gain of

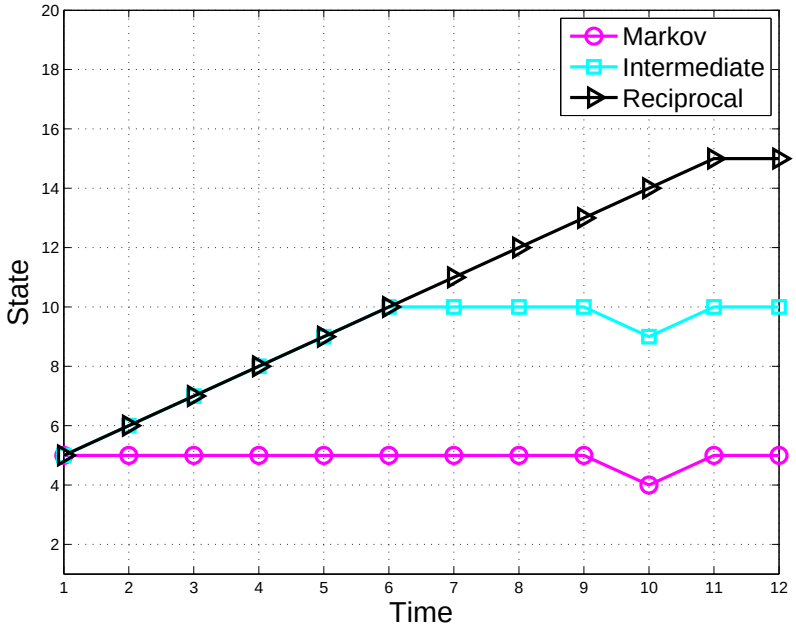


Figure 3.5: Target trajectories representative of Markovian and reciprocal behaviour to varying degrees.

approximately 50% as we transition from the Markov trajectory to the reciprocal trajectory. However, the improvement in the MLSE gradually reduces as the measurement noise variance increases. On the other hand, the improvement in the goodness of fit for the optimal smoother from the Markov to the reciprocal scenario seems to be less sensitive to changes in the noise variance. This supports the results from the first set of simulations where it was observed that the correlation between MLSE error and noise variance was stronger than for the optimal smoother.

Figures 3.8 and 3.9 also show the correlation between each scenario trajectory and F but for varying sequence length $T = 12, 15, 20$ and 25 for the MLSE and optimal smoother, respectively. It can be seen that as a target behaves in a more reciprocal manner, the goodness of fit under the HRP model also increases under both estimators for all sequence lengths. Again, the optimal smoothers perform better than their MLSE counterparts with approximately double the amount of improvement in the goodness of fit across all scenarios for sequence lengths $T \geq 15$. As mentioned before, this can be attributed to the MLSE requirement for a valid target path and due to rounding errors as the MLSE calculates a MAP estimate rather than a minimum variance estimator. Observe also that the correlation between the goodness of fit under the HRP model and reciprocalness of a target is stronger for shorter sequences. This agrees with our observations earlier in which the RC appeared to become more distinct from its underlying

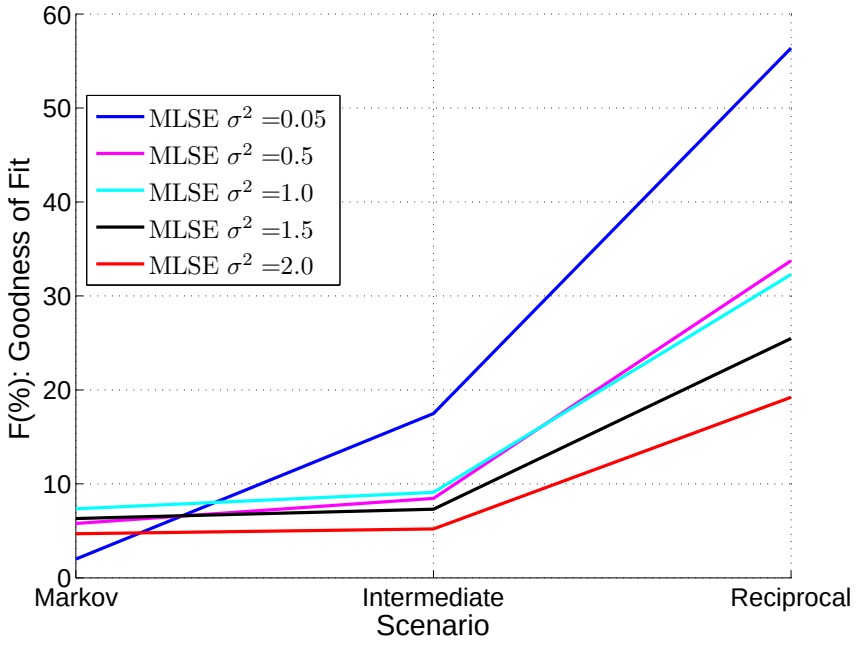


Figure 3.6: Goodness of fit F for each scenario using MLSE for varying noise variance σ^2. Assumes number of states $N_v = 20$ and sequence length $T = 12$.

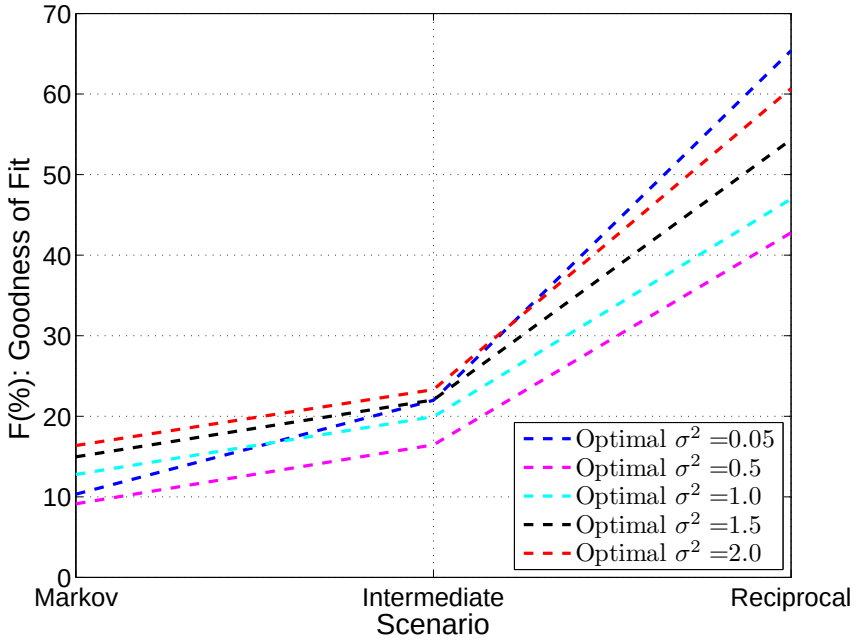


Figure 3.7: Goodness of fit F vs. "reciprocalness" of a target using optimal smoothing for varying noise variance σ^2. Assumes number of states $N = 20$ and sequence length $T = 12$.

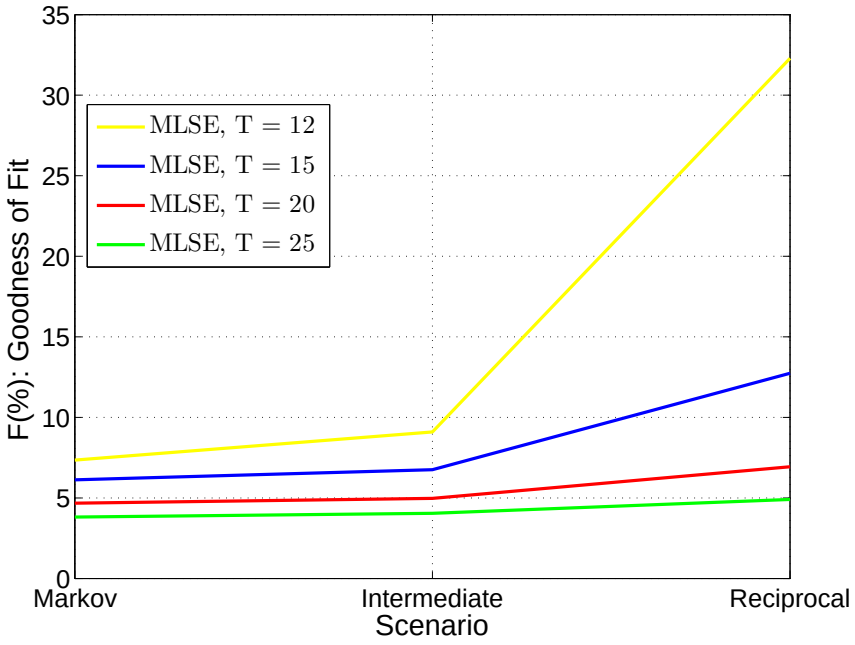


Figure 3.8: Goodness of fit F for each scenario using MLSE for varying sequence length T. Assumes number of states $N_v = 20$ and measurement noise variance $\sigma^2 = 1$

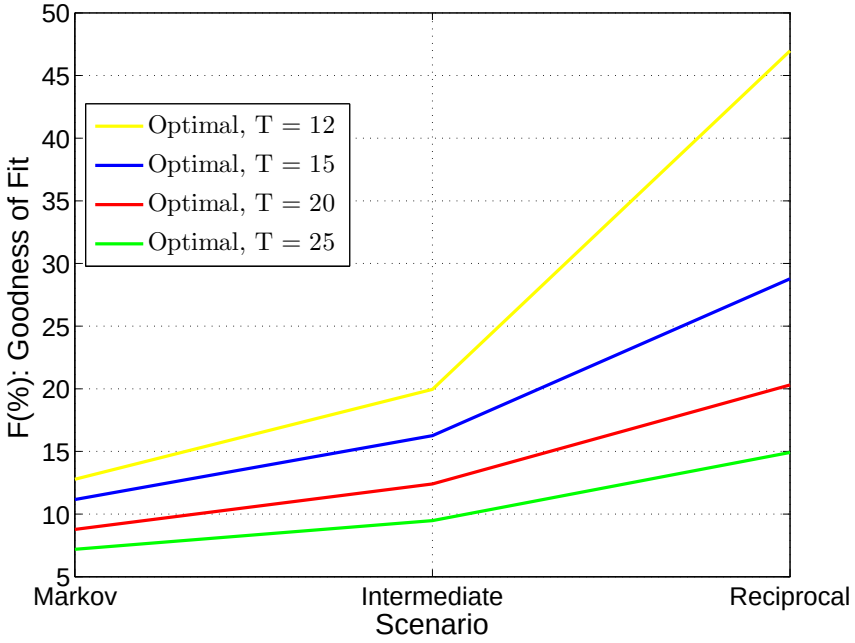


Figure 3.9: Goodness of fit F for each scenario using optimal smoothing for varying sequence length T. Assumes number of states $N_v = 20$ and measurement noise variance $\sigma^2 = 1$.

Markov process as it reached its termination point, resulting in the RC behaving overall less like a Markov process at short sequence lengths.

In general, the benefits of having target destination information is highly dependent on the batch length T and size of the state space N_v. For example, if the batch length is small in comparison to the size of the state space, then possessing target destination information will be very useful as it will narrow down the valid target trajectories to those paths that allow the target endpoint to be reached in the given time length. Conversely, if the target destination is known and the batch length is large in comparison to the size of the state space, there would exist an infinite number of paths to reach the target endpoint. In such a case, having target destination information would provide little advantage.

3.6 Summary

In this chapter, we reviewed the well known Viterbi algorithm, which calculates the MLSE for a HMC. The extension of the Viterbi algorithm to accommodate reciprocal processes is relatively simple through the Markov bridge property; the probability transition models are specified in a similar way to the Markov models and the final destination points are modelled as prior knowledge. As the key contribution for this chapter, we presented the first derivation of the MLSE for a HRC, as first reported in [149]. Simulations of the devised estimator were performed for scenarios featuring a uniform assumption on the endpoints distribution, as well as a scenario featuring a more informative assumption on the end points. It was observed that for a reciprocal target whose endpoint realisations are markedly different from the predicted endpoints generated from a Markov transition, the HRC estimators showed improved state estimation performance relative to the traditional Markov estimators. It was also shown that as a target progressively behaves in a more reciprocal manner, the goodness of fit under a HRC model also improves. The performance of the reciprocal MLSE was also compared to that of a reciprocal-based optimal smoother. Simulation results showed that the reciprocal MLSE was not as effective in tracking the target as the reciprocal optimal smoother and was more sensitive to changes in noise variance. However this result could be attributed to quantisation errors in the discretisation of the state space and the constraint on the MLSE to generate a valid state sequence under the target dynamics model, which can result in poor performance in the case of high measurement noise.

Our simulations were performed with a relatively small number of state points, however the computational complexity can become prohibitive if a large number of discrete states was required to adequately model the target and observation processes.

Chapter 4

Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT)

Under conventional target tracking, the data received from a sensor is generally reduced from an intensity map representation to a point measurement form via a detection thresholding process. The role of the tracker is to associate point measurements from a common target across time and return estimates of the target's trajectory. Track-Before-Detect (TkBD) is an alternative tracking technique which supplies intensity map data directly to the tracker, rather than point measurements from thresholded detections. As a result, all information relating to the target and environment are preserved in the measurement. TkBD can be thought of as a concurrent detection and tracking scheme and has been shown to provide significant gains against targets with a low Signal-to-Noise-Ratio (SNR) [39]. A summary of the existing TkBD approaches in the literature and related references are provided in Chapter 2.

In this chapter we focus on a particular TkBD method called the Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT) [85, 86, 117]. The H-PMHT algorithm can be applied to a wide range of problems as long as an appropriate state estimator exists to perform the maximisation step of the Expectation-Maximisation (EM) algorithm. The key contribution of this chapter is the first implementation of the H-PMHT using a dynamic programming fixed-grid approximation through application of the Viterbi algorithm. The details of this contribution have been published in conference article [133] and journal article [42].

The chapter is arranged as follows: Section 4.1 provides some background about the H-PMHT algorithm; Section 4.2 reviews the derivation of the original H-PMHT, which assumes a multi-

nomial distribution on the quantised measurements; Section 4.3 reviews the Kalman Filter (KF) and Particle Filter (PF) implementations of the H-PMHT, and proposes an alternative implementation via the Viterbi algorithm. This Viterbi-based implementation is compared with the KF and PF implementations using simulated single target scenarios in Section 4.4. Section 4.5 discusses some of the limitations of the H-PMHT algorithm, and Section 2.6 summarises and concludes the chapter.

4.1 Introduction

The H-PMHT algorithm is an efficient multi-target approach to the TkBD problem. Originally introduced in [85, 86, 117], the H-PMHT algorithm is an extension of the Probabilistic Multi-Hypothesis Tracker (PMHT) [121] to track on intensity modulated data. Unlike other TkBD approaches, the H-PMHT algorithm employs a parametric fitting approach and has been shown to give performance that is close to the optimal Bayesian filter at a fraction of the computational cost [39]. The H-PMHT inherently assumes a multi-target scenario but retains linear complexity with the number of targets.

The H-PMHT algorithm is based on the generation of a synthetic histogram by quantising the energy in the sensor data followed by the application of EM [43] mixture modelling to describe the underlying data sources. The quantisation step converts the continuous-valued measurement data in each pixel into an integer-value, which is interpreted as a *count of the number of shots* that fell in each pixel. The counts are assumed to be a realisation of a point process and can be modelled using a multinomial distribution. The H-PMHT compiles the counts to create a synthetic histogram which can be interpreted as the received power across the measurement image. It will be shown later in Chapter 5 that the multinomial assumption is consistent with a Poisson Point Process [118] and an alternative derivation of the H-PMHT is possible under this new measurement model.

The H-PMHT can be applied to a wide range of problems as long as an appropriate state estimator exists to perform the maximisation step of the algorithm. For the case when both the target dynamics and measurement model are assumed to be linear with Gaussian noise, a KF (or smoother in the case of batch processing) can be incorporated into the H-PMHT to perform the state estimation component of the algorithm [119]. This result allows for simple implementation and has thus led to the Kalman-based H-PMHT in the literature [39, 98]. More recently, H-PMHT has been applied to non-linear non-Gaussian problems using non-linear state estimation techniques [35, 36].

In this chapter, we outline the *procedure for computing the first Viterbi-based H-PMHT imple-*

mentation. The Viterbi algorithm is a MAP sequence estimation method for a discrete-state hidden Markov model. In the tracking context, it has been used to solve the non-linear TkBD problem by discretising the state-space. Here it is used to optimise the EM auxiliary function on a per-target basis. For multi-target tracking, fixed-grid approximation techniques can grow exponentially in computational complexity due to the large number of discrete states required. However, when combined with the H-PMHT, the complexity is linear in the number of targets as the H-PMHT provides data association weights that can be used in a bank of independent single-target Viterbi estimators. In the next section, we present a detailed review of the derivation of the H-PMHT algorithm.

4.2 Derivation

This section describes the derivation of the H-PMHT for batch form processing, however the algorithm can be easily modified to perform time-recursive filtering. In subsequent sections, we describe how the H-PMHT algorithm can be implemented for both the smoothing and filtering case.

The derivation of the H-PMHT uses quantisation of the energy in the sensor image. Quantisation is only a stage in the derivation of the algorithm and is not required for implementation. The interpretation of the quantised sensor image as a histogram and the subsequent use of PMHT data association [121] led to the algorithm name.

Assume a scenario in which a sensor observing M targets collects images

$$\mathbf{Z}_t = \{\boldsymbol{z}_t^1, \ldots, \boldsymbol{z}_t^I\}, \tag{4.1}$$

at discrete times $t = 1 \ldots T$ where $\boldsymbol{z}_t^i$ denotes the energy in the ith pixel of the sensor image at time t and I denotes the total number of *observed* pixels. Note that the single index does not constrain the dimensionality of the sensor but rather simplifies notation. For the special case of two-dimensional images, an efficient matrix formulation is possible [35].

The H-PMHT algorithm is also able to accommodate the effect of missing data and assumes that there are other sensor pixels $\mathbf{Z}_t^c$ for which no data was collected. Define

$$\mathbf{Z}_t^c = \{\boldsymbol{z}_t^{I+1}, \ldots, \boldsymbol{z}_t^S\}, \tag{4.2}$$

where S denotes the total number of all possible pixels such that $1 \leq I \leq S$. The remaining measurements in pixels $i = I+1, \ldots, S$ are assumed to be unobservable by the sensor. One use for this concept is in tracking targets near the edge of the sensor frame [120]. The set $\mathbf{Z}_t^c$ can be empty, so this feature can be easily removed if not required.

Let $\boldsymbol{x}_t^m$ denote the state of component m at time t for $m = 0 \dots M$. A component can be attributed to either a clutter or target object, therefore define component $m = 0$ to be the clutter component, which is assumed to be an empty set for all time scans, i.e. $\boldsymbol{x}_t^0 = \emptyset$. Effectively, the clutter distribution is assumed to be known. Assume that the remaining components $m = 1, \dots, M$ are target objects that evolve according to a known process which may be non-linear and stochastic, and let $\mathbf{X} = \boldsymbol{x}_{1:T}^{0:M}$ denote the collection of all component states at all time scans.

For the observed pixels, define the total energy received from the image to be

$$\|\mathbf{Z}_t\| = \sum_{i=1}^{I} \boldsymbol{z}_t^i, \tag{4.3}$$

namely the $L1$-norm of the sensor image. The energy $\boldsymbol{z}_t^i$ in each of the measurement pixels is assumed to be continuous-valued and thus the evaluation of the measurement likelihood is non-trivial. The H-PMHT algorithm provides a convenient way to calculate the likelihood of observing the current measurement $\mathbf{Z}_t$ by employing a quantisation over the sensor data. In Chapter 7, we propose an alternative derivation which modifies the H-PMHT to deal directly with the continuous-valued measurement image via an interpolated Poisson measurement model.

In the first step of the H-PMHT algorithm, the energy in each measurement pixel i is quantised. Let $c^2 > 0$ denote some arbitrary quantisation level and define the quantised vector corresponding to $\mathbf{Z}_t$ as follows,

$$\mathbf{N}_t = \{\mathbf{n}_t^1, \dots, \mathbf{n}_t^I\}, \tag{4.4}$$

where $\mathbf{n}_t^i$ is the quantised energy in the observed pixel i, defined by

$$\mathbf{n}_t^i = \left\lfloor \frac{\boldsymbol{z}_t^i}{c^2} \right\rfloor. \tag{4.5}$$

Note that $\lfloor x \rfloor$ is the floor function which specifies the greatest integer less than or equal to x. In a similar way to (4.3), we can define the total quantised energy in the image,

$$||\mathbf{N}_t|| = \sum_{i=1}^{I} \mathbf{n}_t^i, \tag{4.6}$$

which can also be interpreted as the sample size of the histogram. We emphasise again that the use of the quantised measurement $\mathbf{N}_t$ rather than the original measurement $\mathbf{Z}_t$ is only an intermediate step in the derivation; in the final step, the original sensor data $\mathbf{Z}_t$ is recovered by taking the limit of the quantisation as $c^2 \to 0$.

The H-PMHT measurement model assumes that the integer-valued image $\mathbf{N}_t$ generated after quantisation is a realisation of some mixture process, where the clutter and targets are considered as individual components of the mixture model [118]. That is, the quantised measurement

$\mathbf{n}_t^i$ provides a count of the number of shots in each pixel, where each shot is assumed to be an independent identically distributed (iid) random variable following a distribution defined by the density function, $f_t(\tau|\boldsymbol{x}_t^{1:M};\pi_t^{0:M})$ over continuous space. The parameters $\pi_t^{0:M}$ denote the mixing proportions and will be discussed in more detail shortly. The underlying density can be written as the superposition of target components and background clutter as follows,

$$f_t(\tau|\boldsymbol{x}_t^{1:M};\pi_t^{0:M}) = \pi_t^0 G_0(\tau) + \sum_{m=1}^{M} \pi_t^m h\left(\tau|\boldsymbol{x}_t^m\right), \tag{4.7}$$

where $G_0(\tau)$ is the clutter contribution and $h\left(\tau|\boldsymbol{x}_t^m\right)$ is the point spread function (psf) that describes the influence of target m on the measurement image such that $\int h\left(\tau|\boldsymbol{x}_t^m\right) d\tau = 1$. The mixing proportions π_t^0 and π_t^m form a probability vector, i.e. $\pi_t^m \geq 0$ and $\sum_{m=0}^{M} \pi_t^m = 1$, where $m = 0$ denotes the clutter component. The mixing proportion π_t^m can be interpreted as the relative power of component m at time t.

The counts $\mathbf{N}_t$ are assumed to be multinomially distributed and are generated by making $||\mathbf{N}_t||$ independent draws (with replacement) from the measurement image. The result of each draw leads to a "success" in one of I categories, or equivalently a shot falling into one of I pixels with probability,

$$f_t^i\left(\boldsymbol{x}_t^{1:M};\pi_t^{0:M}\right) = \pi_t^0 h^i(\emptyset) + \sum_{m=1}^{M} \pi_t^m h^i(\boldsymbol{x}_t^m), \tag{4.8}$$

where $f_t^i(\cdot)$ is the probability that τ given in (4.7) is inside pixel i, $h^i(\boldsymbol{x}_t^m)$ denotes the probability that a shot (quantised measurement) due to target m falls in pixel i and is defined as the integral of $h\left(\tau|\boldsymbol{x}_t^m\right)$ over B_i, the spatial extent of pixel i,

$$h^i\left(\boldsymbol{x}_t^m\right) = \int_{B_i} h\left(\tau|\boldsymbol{x}_t^m\right) d\tau. \tag{4.9}$$

Similarly, $h^i(\emptyset)$ denotes the probability of a clutter shot falling in pixel i: the integral of $G_0(\tau)$ over pixel i,

$$h^i\left(\emptyset\right) = \int_{B_i} G_0(\tau) d\tau. \tag{4.10}$$

In the absence of other information, a uniform model for the background clutter can be assumed, i.e. $G_0(\tau)$ is a constant. In many applications, the measurement function is a feature of the sensor, not the target and thus it is common to assume that the psf $h(\cdot)$ is known.

The sensor image does not identify which component of the mixture gave rise to each shot or the precise location of the shot within the pixel. Both of these are treated as missing data and EM is used to marginalise them out of the problem and optimise the target state parameters (i.e. the density map). As mentioned earlier, the H-PMHT algorithm also allows for unobserved pixels $\mathbf{Z}_t^c$. Thus also define $\mathbf{N}_t^c$ as the quantised missing data corresponding to $\mathbf{Z}_t^c$ such that

$$\mathbf{N}_t^c = \{\mathbf{n}_t^{I+1}, \ldots, \mathbf{n}_t^S\}, \tag{4.11}$$

where $\mathbf{n}_t^i$ denotes the missing counts in unobserved pixels $i = I+1, \ldots, S$. The data from these unobserved pixels is also treated as missing data.

If a Bayesian method is adopted to model the mixture parameters of the target states, an expression for the prior density for the target states is required. In the next section, we will discuss the requirements for the prior density under a Bayesian model.

4.2.1 Prior Density

Recall that the output of the quantisation is an integer-valued image where the vector $\mathbf{N}_t$ is assumed to follow a multinomial distribution. This is equivalent to saying that the counts $\mathbf{N}_t$ are generated by making $\|\mathbf{N}_t\|$ independent draws (with replacement) on I categories with probabilities $f_t^i\left(\boldsymbol{x}_t^{1:M}; \pi_t^{0:M}\right)$ for $i = 1 \ldots I$. The assumption of independence between the shot measurements is questionable as the counts are derived from quantised intensities. The quantisation level c^2 influences the counts, and under the independence assumption, dictates the amount of information created in the synthetic measurement data generation, yet the true information content is not related to c^2 at all. Ultimately, when the limit of the quantisation is taken to zero, the synthetically generated data counts in the histogram become infinite. The histogram model assumes there is more data available than the measured $\mathbf{Z}_t$ can in reality possess. Under a Bayesian model, this overabundance of data can overwhelm any prior information and the prior will have no influence on the state estimates.

This problem can be resolved by choosing a prior that is sufficiently non-diffuse to compensate for the overabundance of information in the likelihood function, for example a Markov Model. Assuming independence with time and targets, a standard Bayesian prior for the target components would be a first-order Markov Model such that,

$$\begin{aligned} p(\mathbf{X}) &= \prod_{m=1}^{M} p(\boldsymbol{x}_{0:T}^m) \\ &= \prod_{m=1}^{M} \left[p(\boldsymbol{x}_0^m) \prod_{t=1}^{T} p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m) \right]. \end{aligned} \tag{4.12}$$

In his derivation, Streit [117] proposes an alternative Bayesian prior (based on the above Markov prior),

$$p(\mathbf{X}) = \prod_{m=1}^{M} \left[p(\boldsymbol{x}_0^m) \prod_{t=1}^{T} \left\{ p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m) \right\}^{||\mathbf{N}_t^{total}||} \right], \tag{4.13}$$

where $||\mathbf{N}_t^{total}|| = ||\mathbf{N}_t^c|| + ||\mathbf{N}_t||$ such that $||\mathbf{N}_t^c||$ is the sum of shots over all unobserved pixels. The addition of the $||\mathbf{N}_t^{total}||$ term allows an instance of the prior density to be applied to each individual shot at every time scan. This ensures that the prior is not overwhelmed by the abundance of data as a result of the quantisation step.

4.2.2 Expectation-Maximisation

Before we outline the EM steps for the derivation of the H-PMHT algorithm, first define K_t^{ir} to be the component index associated with r^{th} shot in pixel i at time t. Also, define the vector $\mathbf{L}$ to be the precise location of each shot inside the pixel. Recall that the H-PMHT derivation makes use of quantised measurements, therefore let $\mathbf{N} = \mathbf{N}_{1:T}$ and $\mathbf{N}^c = \mathbf{N}_{1:T}^c$ to be the quantised observed and missing data, respectively, over all time scans.

Define the observer $\mathbf{O}$ consisting of component states $\mathbf{X} = \boldsymbol{x}_{1:T}^{0:M}$ and set of assignments $\mathbf{K} = \{K_t^{ir}\}$ of components to measurement shots $\mathbf{N}$, for $i = 1, \ldots, I, r = 1, \ldots, \mathbf{n}_t^i$ and $t = 1, \ldots, T$. Note that $\mathbf{O}$ also depends on Π, the prior distribution on the assignments $\mathbf{K}$, where $\Pi = \pi_{1:T}^{0:M}$ is defined to be the collection of all $M+1$ component mixing proportions across all time scans.

The observer $\mathbf{O} : \{\mathbf{X}, \mathbf{K}, \Pi, \mathbf{N}\}$ is unknown and the calculation of the Maximum Likelihood Estimate (MLE) for the states $\mathbf{X}$ and prior Π is infeasible due to the exponential complexity of enumerating over all data associations $\mathbf{K}$ of shots to targets.

The solution is to employ the EM method to calculate the maximum likelihood estimate for $\mathbf{X}$ and Π. The EM method provides a general iterative procedure for calculating the MLE given missing data. In the H-PMHT case, the missing data consists of assignments $\mathbf{K}$, the unobserved measurements $\mathbf{N}^c$, and the precise location $\mathbf{L}$ of each shot in the pixels.

At a given time step t, assuming there is an existing estimate of the component states and mixing proportions, the H-PMHT algorithm employs an iterative procedure to determine the probability of the missing data (E-step) and then refines the component and mixing proportions estimates (M-step).

4.2.3 E-Step

At every iteration of the EM algorithm, the E-step of the algorithm evaluates the conditional expectation of the logarithm of the complete data likelihood. The expectation is taken with respect to the missing data $\{\mathbf{N}^c, \mathbf{L}, \mathbf{K}\}$ and conditioned on the observed measurements $\mathbf{N}$ and the previous EM estimates of $\mathbf{X}$ and Π, denoted by $\mathbf{X}'$ and Π'. This is given by the auxiliary

function $Q^{(H)}$ at the current iteration:

$$\begin{aligned} & Q^{(H)}(\mathbf{X}, \Pi | \mathbf{X}', \Pi') \\ =& E_{\mathbf{N}^c \mathbf{L} \mathbf{K}} \Big[\log \Big\{ p(\mathbf{X}, \mathbf{N}, \mathbf{N}^c, \mathbf{L}, \mathbf{K}) \Big\} \Big| \mathbf{X}', \Pi', \mathbf{N} \Big] \\ =& \sum_{\mathbf{N}^C, \mathbf{K}} \int_{\mathbf{L}} \log \Big\{ p(\mathbf{X}, \mathbf{N}, \mathbf{N}^c, \mathbf{L}, \mathbf{K}) \Big\} p(\mathbf{N}^c, \mathbf{L}, \mathbf{K} | \mathbf{X}', \mathbf{N}) d\mathbf{L}, \end{aligned} \tag{4.14}$$

where $E_{\mathbf{N}^c, \mathbf{L}, \mathbf{K}}$ denotes the expectation with respect to the missing data. In the evaluation of the conditional expectation, we have made use of the property $E_B[p(ABC)|C] = \sum_B p(ABC)p(B|C)$ for discrete variables. Note that for the conditional expectation with respect to the sample locations $\mathbf{L}$, the expectation is evaluated using an integral as the exact location in which a shot falls inside a pixel is a continuous variable. Also because Π is a parameter for the prior distribution $\mathbf{K}$, it has been dropped from the last line in (4.14): the dependency on Π' is implied through the inclusion of $\mathbf{K}$.

The first term in (4.14) is exactly the complete data log likelihood. The second term in (4.14) is the conditional density of the missing data and can be expressed as

$$p(\mathbf{N}^c, \mathbf{L}, \mathbf{K} | \mathbf{X}', \mathbf{N}) = \frac{p(\mathbf{X}', \mathbf{N}, \mathbf{N}^c, \mathbf{L}, \mathbf{K})}{p(\mathbf{X}', \mathbf{N})}. \tag{4.15}$$

The numerator and the denominator in (4.15) are exactly the complete and incomplete data likelihoods, respectively. We now derive expressions for both likelihoods.

4.2.3.1 Incomplete Data Likelihood

We now derive an expression for the incomplete data likelihood, which can be interpreted as the joint probability of all component states $\mathbf{X}$ and the quantised counts $\mathbf{N}$. First consider evaluating $p(\mathbf{N}|\mathbf{X})$, the probability of drawing $\mathbf{n}_t^i$ shots from each pixel i given that the source for each shot is unknown. Recall that the counts $\mathbf{n}_t^i$ have assumed to be multinomially distributed. Assuming independence between pixels, the probability is then given by

$$p(\mathbf{N}_t | \mathbf{X}_t') = \gamma_t \prod_{i=1}^{I} \left[\frac{f_t^{i'}}{F_t'} \right]^{\mathbf{n}_t^i}, \tag{4.16}$$

where $f_t^{i'} = f_t^i \left(\left[\boldsymbol{x}_t^{1:M} \right]'; \left[\pi_t^{0:M} \right]' \right)$ and $F_t' = \sum_{j=1}^{S} f_t^{j'}$ are calculated using the estimates of the state and mixing proportions from the previous EM iteration. In (4.16), we have assumed that the probability of an individual shot falling into a pixel is equal to $f_t^{i'}/F_t'$, and is identical for all $\mathbf{n}_t^i$ shots in pixel i. The parameter γ_t denotes the number of ways that $||N_t||$ shots can be

arranged to achieve $\mathbf{N}_t$ across the pixels,

$$\gamma_t = \frac{||N_t||\,!}{\prod_{i=1}^{I} \mathbf{n}_t^i!}. \tag{4.17}$$

Assuming that the temporal elements of $\mathbf{N}$ are independent, the incomplete data likelihood can be expressed as,

$$\begin{aligned} p(\mathbf{X}', \mathbf{N}) &= p(\mathbf{N}|\mathbf{X}')p(\mathbf{X}') \\ &= p(\mathbf{X}') \prod_{t=1}^{T} p(\mathbf{N}_t|\mathbf{X}'_t) \\ &\propto p(\mathbf{X}') \prod_{t=1}^{T} \prod_{i=1}^{I} \left[\frac{f_t^{i'}}{F_t'} \right]^{\mathbf{n}_t^i}. \end{aligned} \tag{4.18}$$

The last simplification in (4.18) is possible because the γ_t term is not a function of $\mathbf{X}$ or Π, so in the context of the optimisation problem in (4.14), it is a scaling constant that can be ignored.

Observe that the $F_t^{'}$ terms consists of a summation of the per-pixel probability over both the observed and unobserved measurement space. If the unobserved measurement space is used to model edging effects in the sensor, then $F_t^{'}$ is composed of a summation over a finite number of observed pixels and conceivably an infinite number of unobserved pixels, that is,

$$F_t^{'} = \sum_{j=1}^{I} f_t^{j'} + \sum_{j=I+1}^{\infty} f_t^{j'}. \tag{4.19}$$

However, as $F_t^{'}$ provides an approximation to the integral of the mixture density defined in (4.7), we can make the following approximation,

$$F_t^{'} = 1 - \epsilon_{\bar{o}}, \tag{4.20}$$

where $\epsilon_{\bar{o}}$ denotes the contribution from the unobserved pixels. Clearly, if the set of unobserved pixels is empty, then $F_t^{'}$ evaluates to unity. However, if the psf $h(\cdot)$ is characterised by a pdf with infinite support, with intensity concentrated towards a region of interest and decaying tails, e.g. a Gaussian, then it is sufficient to assume that the contribution of $\epsilon_{\bar{o}}$ to $F_t^{'}$ will be small, and $F_t^{'}$ can be assumed to be approximately unity. This is important and will be used later to simplify the auxiliary function (4.14).

4.2.3.2 Complete Data Likelihood

Next we derive an expression for the complete data likelihood by considering the joint probability of all the variables in the model. Expand the likelihood such that

$$p(\mathbf{X}, \mathbf{N}, \mathbf{L}, \mathbf{K}) = p(\mathbf{L}|\mathbf{X}, \mathbf{N}, \mathbf{K})p(\mathbf{K}, \mathbf{N}|\mathbf{X})p(\mathbf{X}). \tag{4.21}$$

Note that the unobserved data $\mathbf{N}^c$ has been dropped for notational simplicity as it does not contribute to the conditional probability of $\{\mathbf{L}, \mathbf{K}\}$. Also, observe that the last term on the right hand side (rhs) of equation (4.21) is exactly the prior density discussed earlier in (5.10). We need only consider expressions for the remaining terms on the rhs of (4.21).

The first term $p(\mathbf{L}|\mathbf{X}, \mathbf{N}, \mathbf{K})$ is the probability of drawing the locations $\mathbf{L}$ given the position of all component states $\mathbf{X}$, the quantised counts $\mathbf{N}$, and the assignment of components to shots $\mathbf{K}$ is known. This probability is implicitly conditioned on the pixel location being known. The probability of drawing the r^{th} shot when we know which component m the shot originated from is given by,

$$\begin{aligned} p(\mathbf{L}_t^{ir}|\mathbf{X}, \mathbf{N}, \mathbf{K}) &= p(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}}, \mathbf{L}_t^{ir} \in B_i) \\ &= \frac{p(\mathbf{L}_t^{ir}, \mathbf{L}_t^{ir} \in B_i|\boldsymbol{x}_t^{K_t^{ir}})}{p(\mathbf{L}_t^{ir} \in B_i|\boldsymbol{x}_t^{K_t^{ir}})} \\ &= \frac{h(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}})}{h^i\left(\boldsymbol{x}_t^{K_t^{ir}}\right)}, \end{aligned} \tag{4.22}$$

where $\mathbf{L}_t^{ir}$ is the precise location of the r^{th} shot which fell in pixel i at time t, K_t^{ir} specifies the known component index and $\boldsymbol{x}_t^{K_t^{ir}}$ specifics the spatial mean associated with the r^{th} shot in pixel i at time t. Also, recall that $h(\cdot)$ is the component psf, and $h^i(\cdot)$ is the per-pixel probability and can be calculated for a target or clutter component using (4.9) and (4.10), respectively. Assuming independence of shots, we can express the first term on the rhs of (4.21) as a product across time t, pixels i, and shots r such that

$$p(\mathbf{L}|\mathbf{X}, \mathbf{N}, \mathbf{K}) = \prod_{t=1}^{T} \prod_{i=1}^{I} \prod_{r=1}^{\mathbf{n}_t^i} \frac{h(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}})}{h^i\left(\boldsymbol{x}_t^{K_t^{ir}}\right)}. \tag{4.23}$$

The second term $p(\mathbf{K}, \mathbf{N}|\mathbf{X})$ in (4.21) is the probability of the shot counts and assignments ignorant of the precise locations $\mathbf{L}$ of where the shots fell in the pixel, but given full knowledge of the position of all the component states. Note that this expression contains redundant terms as $\mathbf{N}$ is completely determined by associations $\mathbf{K}$. Given the component state estimates are known, consider evaluating $P(\mathbf{K}|\mathbf{N}, \mathbf{X})$, that is, we want to extract the shots within each pixel that originated from component m. This probability also follows a multinomial distribution, however the categories consist of both pixel i and component indices m. Rather than assuming a multinomial distribution with categories $\{i, m\}$, we can consider the multinomial probability of drawing a shot successfully from category m within each pixel separately. Let $f_t^i = f^i(\boldsymbol{x}_t^{1:M}; \pi_t^{0:M})$, then each draw from this multinomial distribution has probability

$$p(K_t^{ir} = m|\mathbf{X}_t, \mathbf{N}_t) = \frac{\pi_t^m h^i(\boldsymbol{x}_t^m)}{f_t^i}, \tag{4.24}$$

which represents the proportion of energy in the pixel that came from component m scaled by the overall mixture density for pixel i. By substituting in (4.16) and (4.24), and assuming independence across time and pixels, we can write

$$\begin{aligned} p(\mathbf{K},\mathbf{N}|\mathbf{X}) &= p(\mathbf{K}|\mathbf{X},\mathbf{N})p(\mathbf{N}|\mathbf{X}) \\ &= \prod_{t=1}^{T}\left[\prod_{i=1}^{I}\prod_{r=1}^{\mathbf{n}_t^i} p(K_t^{ir}|\mathbf{X}_t,\mathbf{N}_t)\right]p(\mathbf{N}_t|\mathbf{X}_t) \\ &= \prod_{t=1}^{T}\gamma_t\prod_{i=1}^{I}\prod_{r=1}^{\mathbf{n}_t^i}\left[\frac{\pi_t^{K_t^{ir}}h^i(\boldsymbol{x}_t^{K_t^{ir}})}{f_t^i}\times\frac{f_t^i}{F_t}\right] \\ &= \prod_{t=1}^{T}\gamma_t\prod_{i=1}^{I}\prod_{r=1}^{\mathbf{n}_t^i}\pi_t^{K_t^{ir}}\frac{h^i(\boldsymbol{x}_t^{K_t^{ir}})}{F_t}. \end{aligned} \tag{4.25}$$

where $F_t = \sum_{j=1}^{S} f_t^j$, is the summation of the pixel probabilities over both the observed and unobserved measurement space. An expression for the complete data likelihood can be found by substituting (4.23) and (4.25) into (4.21) to give,

$$p(\mathbf{X},\mathbf{N},\mathbf{L},\mathbf{K}) = p(\mathbf{X})\prod_{t=1}^{T}\gamma_t\prod_{i=1}^{I}\prod_{r=1}^{\mathbf{n}_t^i}\pi_t^{K_t^{ir}}\frac{h(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}})}{F_t}. \tag{4.26}$$

4.2.3.3 Auxiliary Function

An expression for the auxiliary function $Q^{(H)}$ can now be found by substituting the complete data likelihood (4.26) and the prior (5.10) into (4.14) to give,

$$\begin{aligned} Q^{(H)}(\mathbf{X},\Pi|\mathbf{X}',\Pi') = \sum_{\mathbf{N}^C,\mathbf{K}}\int\Bigg[\sum_{m=0}^{M}\left(\log\left\{p(\boldsymbol{x}_0^m)\right\}+\sum_{t=1}^{T}||\mathbf{N}_t^{total}||\ \log\left\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\right\}\right) \\ +\sum_{t=1}^{T}\log\{\gamma_t\}+\sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{r=1}^{\mathbf{n}_t^i}\log\left\{\pi_t^{K_t^{ir}}\right\}-\sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{r=1}^{\mathbf{n}_t^i}\log\{F_t\}+\sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{r=1}^{n_t^i}\log\left\{h(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}})\right\}\Bigg] \\ \times\quad p(\mathbf{N}^c,\mathbf{L},\mathbf{K}|\mathbf{X}',\mathbf{N})\,d\mathbf{L}. \end{aligned} \tag{4.27}$$

The last term on the rhs of (4.27) is the conditional density of the missing data (4.15) and can be simplified as follows,

$$\begin{aligned} p(\mathbf{N}^c,\mathbf{L},\mathbf{K}|\mathbf{X}',\mathbf{N}) &= \frac{p(\mathbf{N}^c|\mathbf{X}',\mathbf{N},\mathbf{L},\mathbf{K})p(\mathbf{X}',\mathbf{N},\mathbf{L},\mathbf{K})}{p(\mathbf{X}',\mathbf{N})} \\ &= p(\mathbf{N}^c|\mathbf{X}',\mathbf{N})p(\mathbf{L}|\mathbf{X}',\mathbf{N},\mathbf{K})p(\mathbf{K}|\mathbf{X}',\mathbf{N}). \end{aligned} \tag{4.28}$$

It can be further simplified by substituting (4.23) and (4.24) to give,

$$p(\mathbf{N}^c, \mathbf{L}, \mathbf{K}|\mathbf{X}', \mathbf{N}) = \prod_{t=1}^{T} p(\mathbf{N}_t^c|\mathbf{X}', \mathbf{N}_t) \prod_{i=1}^{I} \prod_{r=1}^{\mathbf{n}_t^i} \pi_t^{K_t^{ir}{}'} \frac{h\left(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}{}'}\right)}{f_t^{i'}}. \tag{4.29}$$

Substituting (5.26) into (4.27) and re-inserting $\mathbf{N}^c$ terms, we can now apply the expectation across the missing data to the relevant terms to simplify the auxiliary function to,

$$\begin{aligned} Q^{(H)}(\mathbf{X}, \Pi|\mathbf{X}', \Pi') = \sum_{m=0}^{M} \Bigg\{ & \log\{p(\boldsymbol{x}_0^m)\} + \sum_{t=1}^{T} \log\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\} \sum_{\mathbf{N}_t^c} ||\mathbf{N}_t^{total}|| p(\mathbf{N}_t^c|\mathbf{X}_t', \mathbf{N}_t) \Bigg\} \\ & + \sum_{\mathbf{N}^c, \mathbf{K}} \left[\sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{\mathbf{n}_t^i} \log\left\{ \pi_t^{K_t^{ir}} \right\} \right] p(\mathbf{K}|\mathbf{X}', \mathbf{N}) \\ & + \sum_{\mathbf{N}^c, \mathbf{K}} \int \left[\sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{\mathbf{n}_t^i} \log\left\{ h(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}}) \right\} \right] p(\mathbf{L}, \mathbf{K}|\mathbf{X}', \mathbf{N}) d\mathbf{L}. \end{aligned} \tag{4.30}$$

Note that the γ_t term has been removed from the auxiliary function as it is not dependent on $\mathbf{X}$ or Π. Recall that F_t approximates the integral of the mixture density (4.7) and can be evaluated using (4.20). If we assume that the psf $h(\cdot)$ is a pdf with intensity concentrated in a region interest with decaying tails, e.g. Gaussian, then the contribution from the unobserved pixels can be assumed to be small and F_t is close to unity. In such a case, the F_t term in (4.27) can be ignored as it is a scaling constant when the auxiliary function is optimised.

We now simplify each line in (4.30) separately. In the first line, the component states are independent from the missing data and we only need to consider the expectation over the unobserved counts $\mathbf{N}^c$ to derive an expected value. We retain terms dependent on $\mathbf{N}^c$ and all other terms are marginalised out. To evaluate this line, first define the quantity $\bar{n}_t^i$, for all observed and unobserved pixels $i = 1, \ldots, S$ such that

$$\begin{aligned} \bar{n}_t^i &= E_{\mathbf{N}^c}[\mathbf{n}_t^i|\mathbf{X}', \mathbf{N}] \\ &= \sum_{\mathbf{N}^c} \mathbf{n}_t^i \prod_{\tilde{t}=1}^{T} p(\mathbf{N}_{\tilde{t}}^c|\mathbf{X}_{\tilde{t}}', \mathbf{N}_{\tilde{t}}). \end{aligned} \tag{4.31}$$

Above, the sum of products $\sum_{\mathbf{N}^c} \prod_{t=1}^{T} p(\mathbf{N}_t^c|\mathbf{X}_t', \mathbf{N}_t)$ can be written as a product of sums such that,

$$\sum_{\mathbf{N}^c} \prod_{t} p(\mathbf{N}_t^c|\mathbf{X}_t', \mathbf{N}_t) = \left(\sum_{\mathbf{N}_1^c} p\left(\mathbf{N}_1^c|\mathbf{X}_1', \mathbf{N}_1\right) \right) \ldots \left(\sum_{\mathbf{N}_T^c} p\left(\mathbf{N}_T^c|\mathbf{X}_T', \mathbf{N}_T\right) \right), \tag{4.32}$$

where each of the product terms on the rhs (4.32) is a probability density function (pdf) which sums to unity. Given this, the expectation in (4.31) can be simplified in two ways. If pixel i is

part of the set of observed pixels $\mathcal{O}$, then $\bar{n}_t^i = \mathbf{n}_t^i$, as the observed shots $\mathbf{n}_t^i$ do not contribute to the summation over $\mathbf{N}^c$. If however, pixel i is part of the set of unobserved pixels $\bar{\mathcal{O}}$,

$$\bar{n}_t^i = \sum_{\mathbf{N}_t^c} \mathbf{n}_t^i \, p(\mathbf{N}_t^c | \mathbf{X}'_t, \mathbf{N}_t) \left(\sum_{\mathbf{N}^c \setminus \mathbf{N}_t^c} \prod_{\substack{\tilde{t}=1 \\ \tilde{t} \neq t}}^{T} p(\mathbf{N}_{\tilde{t}}^c | \mathbf{X}'_{\tilde{t}}, \mathbf{N}_{\tilde{t}}) \right), \tag{4.33}$$

where the term in the brackets of (4.33) is equal to unity by (4.32). Thus for all unobserved pixels i,

$$\bar{n}_t^i = \sum_{\mathbf{N}_t^c} \mathbf{n}_t^i \, p(\mathbf{N}_t^c | \mathbf{X}'_t, \mathbf{N}_t). \tag{4.34}$$

Streit assumes that the probability of seeing the unobserved counts $\mathbf{N}_t^c$ conditioned on the observed counts $\mathbf{N}_t$ can be modelled using a negative multinomial distribution. The negative multinomial distribution generalises the negative binomial distribution to more than two outcomes and is commonly used to model problems which feature missing data.

Suppose we have an experiment with $S+1 \geq 2$ outcomes where $X_0, \ldots, X_S$ provides a count of each outcome, and $p_0, \ldots, p_S$ denotes the probability of each outcome, respectively. Clearly, the outcomes $X_0, \ldots, X_S$ can be modelled using a multinomial distribution. However, if we were to stop the experiment when the count in X_0 reached some predetermined number k_0, then the distribution of the remaining counts $X_1, \ldots, X_S$ follows a negative multinomial distribution with mean given by,

$$E[X_i] = \frac{k_0}{p_0} p, \quad i = 1, \ldots, S. \tag{4.35}$$

where $p = \sum_{i=1}^{S} p_i$ is the probability of observing outcomes $X_1, \ldots, X_S$.

The negative multinomial distribution can be used to model the number of successes before a certain number of failures are observed. In the context of the TkBD problem, let the event of a shot falling into an unobserved pixel constitute a 'success' and a shot falling into an observed pixel as a 'failure'. Define $\mathbf{N}_t^0$ to encompass all events in which a shot is drawn from an observed pixel i.e. $\mathbf{N}_t^0 = \{\mathbf{N}_t^1, \ldots, \mathbf{N}_t^I\}$. Then, the negative multinomial distribution can be used to model the expected number of unobserved shots (or successes) $\mathbf{N}_t^{I+1}, \ldots, \mathbf{N}_t^S$ before $||\mathbf{N}_t||$ observed shots (failures) are drawn.

Define the probability of success i.e. a shot being drawn from an unobserved pixel $j = I+1, \ldots, S$ as,

$$p(\mathbf{N}_t^j) = \frac{f^j}{F_t}. \tag{4.36}$$

Let the number of predefined failures be given by $||\mathbf{N}_t||$ and the probability of failure i.e. a shot being drawn from an observed pixel $i = 1, \dots I$ be given by,

$$\begin{aligned} p(\mathbf{N}_t^0) &= \sum_{i=1}^{I} p(\mathbf{N}^i t) \\ &= \sum_{i=1}^{I} \frac{f_t^i}{F_t}. \end{aligned} \tag{4.37}$$

Then by (4.35), we can calculate the expected count in the unobserved pixels $j = I+1, \dots, S$ as following:

$$\begin{aligned} E[\mathbf{N}_t^j] &= ||\mathbf{N}_t|| \frac{\frac{f^j}{F_t}}{\sum_{i=1}^{I} \frac{f_t^i}{F_t}} \\ &= ||\mathbf{N}_t|| \frac{f^j}{F_t^{\mathcal{O}}}. \end{aligned} \tag{4.38}$$

Thus, we can simplify (4.31) as follows,

$$\bar{n}_t^i = \begin{cases} \mathbf{n}_t^i, & i \in \mathcal{O}, \\ ||\mathbf{N}_t|| \frac{f_t^{i'}}{F_t^{\mathcal{O}'}}, & i \in \bar{\mathcal{O}}, \end{cases} \tag{4.39}$$

where and $F_t^{\mathcal{O}'} = \sum_{i=1}^{I} f_t^i$ is the summation of pixel probabilities over the observed measurement space.

We can now write the following expression:

$$\begin{aligned} \sum_{\mathbf{N}^c} ||\mathbf{N}_t^{total}||\, p(\mathbf{N}^c|\mathbf{X}', \mathbf{N}) &= ||\mathbf{N}_t|| + \sum_{\mathbf{N}^c} ||\mathbf{N}_t^c||\, p(\mathbf{N}^c|\mathbf{X}', \mathbf{N}) \\ &= ||\mathbf{N}_t|| + \sum_{i=I+1}^{S} \bar{n}_t^i \\ &= ||\mathbf{N}_t|| + \sum_{i=I+1}^{S} ||\mathbf{N}_t|| \frac{f_t^{i'}}{F_t^{\mathcal{O}'}} \\ &= \frac{||\mathbf{N}_t|| F_t^{\mathcal{O}'} + \sum_{i=I+1}^{S} ||\mathbf{N}_t|| f_t^{i'}}{F_t^{\mathcal{O}'}} \\ &= \frac{||\mathbf{N}_t||}{F_t^{\mathcal{O}'}} \left\{ \sum_{i=1}^{I} f_t^{i'} + \sum_{i=I+1}^{S} f_t^{i'} \right\} \\ &= \frac{||\mathbf{N}_t||}{F_t^{\mathcal{O}'}}. \end{aligned} \tag{4.40}$$

Note that $\sum_{i=1}^{I} f_t^{i'} + \sum_{i=I+1}^{S} f_t^{i'} = 1$ as the summation encompasses the entire observation space. Thus, the first line in (4.30) can be expressed as

$$\begin{aligned} & \sum_{m=0}^{M} \left\{ \log\{p(\boldsymbol{x}_0^m)\} + \sum_{t=1}^{T} \log\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\} \sum_{\mathbf{N}_t^c} ||\mathbf{N}_t^{total}|| p(\mathbf{N}_t^c|\mathbf{X}_t', \mathbf{N}_t) \right\} \\ = & \sum_{m=0}^{M} \left\{ \log\{p(\boldsymbol{x}_0^m)\} + \sum_{t=1}^{T} \frac{||\mathbf{N}_t||}{F_t^{\mathcal{O}'}} \log\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\} \right\} \end{aligned} \tag{4.41}$$

Consider simplifying the second line on the rhs (4.30). The inner summand of $\left[\sum\sum\sum \log \pi^{K_t^{ir}}\right]$ depends only on one assignment, namely K_t^{ir}. In a similar way to (4.32), we assume independence of the K_t^{ir} term and write,

$$\begin{aligned} & \sum_{\mathbf{K}} \left[\sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{\mathbf{n}_t^i} \log\left\{\pi_t^{K_t^{ir}}\right\} \right] p(\mathbf{K}|\mathbf{X}', \mathbf{N}) \\ = & \left\{ \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{\mathbf{n}_t^i} \sum_{K_t^{ir}=0}^{M} \log\left\{\pi_t^{K_t^{ir}}\right\} p(K_t^{ir}|\mathbf{X}_t', \mathbf{N}_t) \right\} \sum_{K \setminus K_t^{ir}} p(K \setminus K_t^{ir}|\mathbf{X}_t', \mathbf{N}). \end{aligned} \tag{4.42}$$

The right-most term in (4.42) is the marginal probability of all the assignments except K_t^{ir} and is summed over all possible values. It equates to unity. Furthermore, by substituting in the prior for associations $\mathbf{K}$ derived in (4.24), (4.42) simplifies to,

$$\begin{aligned} & \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{\mathbf{n}_t^i} \sum_{K_t^{ir}=0}^{M} \log\left\{\pi_t^{K_t^{ir}}\right\} p(K_t^{ir}|\mathbf{X}_t', \mathbf{N}_t) \\ = & \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{\mathbf{n}_t^i} \sum_{K_t^{ir}=0}^{M} \frac{\pi_t^{K_t^{ir'}} h^i(\boldsymbol{x}_t^{K_t^{ir'}})}{f_t^{i'}} \log\left\{\pi_t^{K_t^{ir}}\right\} \\ = & \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{K_t^{ir}=0}^{M} \sum_{r=1}^{\mathbf{n}_t^i} \frac{\pi_t^{K_t^{ir'}} h^i(\boldsymbol{x}_t^{K_t^{ir'}})}{f_t^{i'}} \log\left\{\pi_t^{K_t^{ir}}\right\} \end{aligned} \tag{4.43}$$

$$= \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{m=0}^{M} \frac{\pi_t^{m'} \bar{n}_t^i \, h^i(\boldsymbol{x}_t^{m'})}{f_t^{i'}} \log\{\pi_t^m\}. \tag{4.44}$$

Note that the summand in (4.43) is independent of shots r and will be identical for each shot r due to the IID assumption. As a result, we can replace the summation over shots r by a single multiplication of the $\bar{n}_t^i$ term. Also note, that in (4.44), the summation over associations K_t^{ir} converts to a summation over all targets m to simplify notation.

We now consider the last term on the rhs of (4.30). As before, the internal summand depends on only one shot so the expectation simplifies to the marginal expectation over that specific

shot. Substituting (4.23) and (4.24) into the last term on the rhs of (4.30) results in

$$\begin{aligned}
&\sum_{\mathbf{K}} \int \left[\sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{n_t^i} \log \left\{ h(\mathbf{L}_t^{ir} | \boldsymbol{x}_t^{K_t^{ir}}) \right\} \right] p(\mathbf{L}, \mathbf{K} | \mathbf{X}', \mathbf{N}) d\mathbf{L} \\
= & \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{n_t^i} \sum_{K_t^{ir}=0}^{M} \int \log \left\{ h(\mathbf{L}_t^{ir} | \boldsymbol{x}_t^{K_t^{ir}}) \right\} p(\mathbf{L}_t^{ir} | \mathbf{X}', \mathbf{N}, \mathbf{K}) p(\mathbf{K}_t^{ir} | \mathbf{X}', \mathbf{N}) d\mathbf{L}_t^{ir} \\
= & \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{m=0}^{M} \frac{\pi_t^{m'} \bar{n}_t^i}{f_t^{i'}} \int_{B_i} h(\tau | \boldsymbol{x}_t^{m'}) \log \left\{ h(\tau | \boldsymbol{x}_t^{m}) \right\} d\tau. \qquad (4.45)
\end{aligned}$$

Again we have substituted the summation over shots r by a multiplication of a $\bar{n}_t^i$ term and converted the summation over associations K_t^{ir} to a summation over all targets m. Note that although (4.45) contains two $h(\cdot)$ terms, only the $\log\{h(\cdot)\}$ term depends on the state we are maximising over: the other term is a function of the previous estimate. Finally, by substituting (4.41), (4.44) and (4.45) into (4.30), the auxiliary function simplifies to,

$$\begin{aligned}
Q^{(H)}(\mathbf{X}, \Pi | \mathbf{X}', \Pi') = \sum_{m=0}^{M} \Bigg\{ & \log\{p(\boldsymbol{x}_0^m)\} + \sum_{t=1}^{T} \frac{||\mathbf{N}_t||}{F_t^{\mathcal{O}'}} \log\{p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m)\} \\
& + \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{\pi_t^{m'} \bar{n}_t^i h^i(\boldsymbol{x}_t^{m'})}{f_t^{i'}} \log\{\pi_t^m\} \\
& + \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{\pi_t^{m'} \bar{n}_t^i}{f_t^{i'}} \int_{B_i} h(\tau | \boldsymbol{x}_t^{m'}) \log \left\{ h(\tau | \boldsymbol{x}_t^{m}) \right\} d\tau \Bigg\}. \qquad (4.46)
\end{aligned}$$

Recall that the use of the quantised measurement $\mathbf{N}_t$ rather than the original measurement $\mathbf{Z}_t$ is only an intermediate step in the derivation of the algorithm. Having defined an EM algorithm at a prescribed quantisation, the limit of the auxiliary function $Q^{(H)}$ is taken in the final step of the derivation. This leads to the H-PMHT which no longer uses quantisation.

4.2.4 Taking the Limit of the Quantisation

Recall that in the first step of the H-PMHT derivation, the measurement data is quantised according to some arbitrary quantisation level c^2. In the final step of the derivation, the limit of the auxiliary function is taken, such that, $Q^{(H)}$ is replaced with $\lim_{c^2 \to 0} c^2 Q^{(H)}$. Note that only the parameters $\bar{n}_t^i$ and $||\mathbf{N}_t||$ will be affected by the limit of the quantisation. As $\bar{n}_t^i$ depends on $\mathbf{n}_t^i$ by (4.39), we can use (4.5) to write the following,

$$\begin{aligned}
\lim_{c^2 \to 0} c^2 \mathbf{n}_t^i &= \lim_{c^2 \to 0} c^2 \left\lfloor \frac{\boldsymbol{z}_t^i}{c^2} \right\rfloor \\
&= \lim_{c^2 \to 0} \left\{ c^2 \frac{\boldsymbol{z}_t^i}{c^2} - \epsilon \right\} \\
&= \boldsymbol{z}_t^i, \qquad (4.47)
\end{aligned}$$

where $\epsilon < c^2$. Using (4.47), we can also write

$$\begin{aligned}\lim_{c^2\to 0} c^2||\mathbf{N}_t|| &= \lim_{c^2\to 0}\left\{c^2\sum_i\left\lfloor\frac{\boldsymbol{z}_t^i}{c^2}\right\rfloor\right\}\\ &= \sum_{i=1}^{I}\lim_{c^2\to 0}\left\{c^2\frac{\boldsymbol{z}_t^i}{c^2}-\epsilon_i\right\}\\ &= \sum_{i=1}^{I}\boldsymbol{z}_t^i\\ &= ||\mathbf{Z}_t||. \end{aligned} \tag{4.48}$$

It is assumed that floor function is sufficiently stable such that the summation over pixels i can be taken outside the limit. When the limit of $Q^{(H)}$ is taken, we can substitute (4.47) and (4.48) into (4.46) to give the following expression for the auxiliary function,

$$\begin{aligned}Q^{(H)}(\mathbf{X},\Pi|\mathbf{X}',\Pi') = \sum_{m=0}^{M}\Bigg\{&\log\{p(\boldsymbol{x}_0^m)\}\left[\sum_{t=1}^{T}\frac{||\mathbf{Z}_t||}{F_t^{\mathcal{O}'}}\log\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\}\right]\\ &+\sum_{t=1}^{T}\sum_{i=1}^{I}\frac{\pi_t^{m'}\bar{z}_t^i}{f_t^{i'}}h^i(\boldsymbol{x}_t^{m'})\log\{\pi_t^m\}\\ &+\sum_{t=1}^{T}\sum_{i=1}^{I}\frac{\pi_t^{m'}\bar{z}_t^i}{f_t^{i'}}\int_{B_i}h(\tau|\boldsymbol{x}_t^{m'})\log\left\{h(\tau|\boldsymbol{x}_t^m)\right\}d\tau\Bigg\}. \end{aligned} \tag{4.49}$$

where

$$\bar{z}_t^i = \begin{cases}\boldsymbol{z}_t^i & i\in\ \mathcal{O},\\ \\ ||\mathbf{Z}_t||\dfrac{f^i}{F_t^{\mathcal{O}}} & i\in\ \bar{\mathcal{O}}.\end{cases} \tag{4.50}$$

We can further decompose (4.49) into two separate expressions for individually estimating the target states $\boldsymbol{x}_t^m$ and mixing proportions π_t^m such that,

$$Q^{(H)}(\mathbf{X},\Pi|\mathbf{X}',\Pi') = \sum_{m=0}^{M}Q_X^m + \sum_{t=1}^{T}Q_{t\pi}, \tag{4.51}$$

where

$$\begin{aligned}Q_X^m = \log\left\{p(\boldsymbol{x}_0^m)\right\} &+ \sum_{t=1}^{T}\frac{||\mathbf{Z}_t||}{F_t^{\mathcal{O}'}}\log\left\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\right\}\\ &+\sum_{t=1}^{T}\sum_{i=1}^{I}\frac{\pi_t^{m'}\bar{z}_t^i}{f_t^{i'}}\int_{B_i}h(\tau|\boldsymbol{x}_t^{m'})\log\left\{h(\tau|\boldsymbol{x}_t^m)\right\}d\tau, \end{aligned} \tag{4.52}$$

$$Q_{t\pi} = \sum_{m=0}^{M}\sum_{i=1}^{I}\frac{\pi_t^{m'}\bar{z}_t^i}{f_t^{i'}}h^i(\boldsymbol{x}_t^{m'})\log\{\pi_t^m\}. \tag{4.53}$$

4.2.5 M-Step

In the maximisation step of the EM procedure, each component of the auxiliary function can be maximised separately,

$$\hat{\mathbf{X}}^m = \arg\max Q_X^m, \tag{4.54}$$

$$\hat{\Pi} = \arg\max \sum_{t=1}^{T} Q_{t\pi}. \tag{4.55}$$

First, consider maximising the state auxiliary function (4.54). Upon initial examination, the state auxiliary function appears to be quite daunting to solve. However by regrouping terms, we can write the auxiliary function as follows:,

$$\begin{aligned} Q_X^m &= \log\left\{p(\boldsymbol{x}_0^m)\right\} \\ &\quad + \sum_{t=1}^{T}\left[\log\left\{\psi_t^m(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\right\} + \log\left\{\zeta_t^m(\boldsymbol{z}_t^i, \boldsymbol{x}_t^m)\right\}\right], \end{aligned} \tag{4.56}$$

where $\psi_t^m(\cdot)$ can be interpreted as a time-varying process model and $\zeta_t^m(\cdot)$, as a time varying measurement function. The auxiliary function for the H-PMHT can be reposed into an equivalent measurement tracking problem, consisting of a target dynamics model and a measurement likelihood model. Thus, if an appropriate measurement estimator exists for the problem under consideration, it can be used to perform the state estimation component of the H-PMHT algorithm.

Although we have derived the H-PMHT in batch form, it is also possible to adapt the H-PMHT to allow for time-recursive filtering of the component state estimates. A discussion of both batch processing and time-recursive filters for the H-PMHT is presented in Section 4.3.

We now consider maximising the mixing proportion auxiliary function (4.55). An analytic expression for the mixing proportion estimate can be found by employing the Lagrangian multiplier method to maximise (4.55) subject to the normalisation constraint $\sum_{m=0}^{M} \pi_t^m = 1$. Define $\tilde{\lambda}_t$ to be the Lagrangian multiplier and let the Lagrange function be given by,

$$\begin{aligned} L_{t,\pi} &= Q_{t\pi} + \tilde{\lambda}_t\Big(1 - \sum_{m=0}^{M} \pi_t^m\Big) \\ &= \sum_{m=0}^{M}\sum_{i=1}^{I} \frac{\pi_t^{m'}\bar{z}_t^i}{f_t^{i'}} h^i(\boldsymbol{x}_t^{m'}) \log\left\{\pi_t^m\right\} + \tilde{\lambda}_t\Big(1 - \sum_{m=0}^{M} \pi_t^m\Big). \end{aligned} \tag{4.57}$$

Differentiating with respect to π_t^m and solving for stationary points results in the update,

$$\pi_t^m = \frac{\pi_t^{m'}}{\tilde{\lambda}_t} \sum_{i=1}^{I} \frac{\pi_t^{m'} \bar{z}_t^i}{f_t^{i'}} h^i(\boldsymbol{x}_t^{m'}). \tag{4.58}$$

By summing (4.58) over all components m and making use of the normalisation constraint, the Lagrangian multiplier can be expressed as,

$$\tilde{\lambda}_t = \sum_{m=1}^{M} \pi_t^{m'} \sum_{i=1}^{I} \frac{\pi_t^{m'} \bar{z}_t^i}{f_t^{i'}} h^i(\boldsymbol{x}_t^{m'}) \tag{4.59}$$

Substituting (4.59) into (4.58) results in a final updated expression for the mixing terms,

$$\pi_t^m = \frac{p_t^m}{\sum_{m=0}^{M} p_t^m}, \tag{4.60}$$

where

$$p_t^m = \pi_t^{m'} \sum_{i=1}^{I} \frac{\bar{z}_t^i h^i(\boldsymbol{x}_t^{m'})}{f_t^{i'}} \tag{4.61}$$

4.3 Implementations

EM convergence is determined in terms of the auxiliary function Q_X^m, however the computation of this term can be expensive, and it is only used to determine convergence. To streamline computations, we constructed a convergence test based around the estimates of the target state. A track was considered converged when the difference between track estimates from one EM iteration to the next dropped below some threshold.

In the batch version of the H-PMHT, the entire target trajectory is updated within each EM iteration, which allows for a smoothed estimate of the target trajectory as the maximisation is performed over both target states and time. However this does not allow for real time processing which can be useful in some applications. As mentioned earlier, the H-PMHT can be easily adapted to allow for time-recursive filtering. The advantage of a time-recursive filter is the ability to process measurements sequentially as they become available at each time scan. The disadvantage is that the state estimates are unsmoothed so the trajectory formed by linking the filtering estimates across time may violate the constraints imposed by the target dynamics model and the estimation error will be higher.

The original presentation of the H-PMHT algorithm [117] showed that for a linear Gaussian psf $h(\tau|\boldsymbol{x}_t^m)$, the EM auxiliary function is equivalent to the log-likelihood of a point-measurement

filtering problem. This result allows for simple implementation and has thus led to the Kalman-based H-PMHT filter [35].

The H-PMHT is not restricted to linear Gaussian applications. In the case when the psf is non-Gaussian and non-linear, there is no analytic way of reducing the auxiliary function (4.52) to an equivalent point measurement problem with the same likelihood. Nevertheless, the EM procedure can still be employed to return estimates of the mixing proportions and target states using non-linear state estimation methods such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) or PF [36]. In the following subsections, we give a brief review of how the KF and PF may be applied to solve for the target states estimates (4.54) for the H-PMHT algorithm. A batch solution to the H-PMHT via the Viterbi algorithm is also possible but has not been demonstrated in the literature. A description for implementing a Viterbi-based H-PMHT is presented in subsection 4.3.3 and is the key contribution of this chapter [133].

4.3.1 Kalman Filter Implementation

For a linear Gaussian psf,

$$h(\tau|\boldsymbol{x}_t^m) = \mathcal{N}(\tau; \mathsf{H}\boldsymbol{x}_t^m, \mathsf{R}), \tag{4.62}$$

the EM target auxiliary function is equivalent to the log-likelihood of a point-measurement filtering problem with associated synthetic measurements. If the target dynamics are also linear and Gaussian then the estimator used to obtain the target states is a KF [117]. Under these assumptions, analytic expressions for the per-cell contribution for each component (4.8) and for the synthetic measurements can be derived. The procedure for calculating these integrals is based on a software implementation presented in [35].

To derive the equivalent synthetic point measurement for the KF implementation, consider one time slice of the measurement component of the auxiliary function (4.52). By factorising the measurement component and completing the square, we can show that the measurement component is a sum of quadratics in the target state which when collected into a single quadratic, is equivalent to the log of a normal distribution:

$$
\begin{aligned}
&\sum_i \frac{\pi_t^{m'}\bar{z}_t^i}{f_t^{i'}} \int_{B_i} h\left(\tau|\boldsymbol{x}_t^{m'}\right) \log\left\{h\left(\tau|\boldsymbol{x}_t^{m}\right)\right\} d\tau \\
&= \quad \mathsf{const} - \frac{\pi_t^{m'}}{2} \sum_i \frac{\bar{z}_t^i}{f_t^{i'}} \int_{B_i} h\left(\tau|\boldsymbol{x}_t^{m'}\right) \left(\tau - \mathsf{H}\boldsymbol{x}_t^m\right)^{\mathsf{T}} \mathsf{R}^{-1} \left(\tau - \mathsf{H}\boldsymbol{x}_t^m\right) d\tau, \\
&= \quad \mathsf{const} - \frac{\pi_t^{m'}}{2} \sum_i \frac{\bar{z}_t^i}{f_t^{i'}} \int_{B_i} h\left(\tau|\boldsymbol{x}_t^{m'}\right) \left(\mathsf{H}\boldsymbol{x}_t^m\right)^{\mathsf{T}} \mathsf{R}^{-1} \mathsf{H}\boldsymbol{x}_t^m d\tau \\
&\qquad\qquad + \pi_t^{m'} \sum_i \frac{\bar{z}_t^i}{f_t^{i'}} \int_{B_i} h\left(\tau|\boldsymbol{x}_t^{m'}\right) \left(\mathsf{H}\boldsymbol{x}_t^m\right)^{\mathsf{T}} \mathsf{R}^{-1} \tau d\tau, \\
&= \quad \mathsf{const} - \frac{\pi_t^{m'}}{2} (\mathsf{H}\boldsymbol{x}_t^m)^{\mathsf{T}} \mathsf{R}^{-1} \mathsf{H}\boldsymbol{x}_t^m \sum_i \frac{\bar{z}_t^i}{f_t^{i'}} h^i\left(\boldsymbol{x}_t^m\right) + \pi_t^{m'} (\mathsf{H}\boldsymbol{x}_t^m)^{\mathsf{T}} \mathsf{R}^{-1} \sum_i \frac{\bar{z}_t^i}{f_t^{i'}} \int_{B_i} \tau h\left(\tau|\boldsymbol{x}_t^{m'}\right) d\tau, \\
&= \quad \mathsf{const} - \frac{1}{2} \left(\tilde{\boldsymbol{z}}_{t,m}^{(m,H)} - \mathsf{H}\boldsymbol{x}_t^m\right)^{\mathsf{T}} \tilde{\mathsf{R}}_{t,m}^{(H)^{-1}} \left(\tilde{\boldsymbol{z}}_{t,m}^{(m,H)} - \mathsf{H}\boldsymbol{x}_t^m\right). \qquad (4.63)
\end{aligned}
$$

Note that in (4.63), all terms that are not dependent on $\boldsymbol{x}_t^m$ are absorbed into the constant term, whose value can change from line to line. The resulting synthetic mean and covariance is given by,

$$
\tilde{\boldsymbol{z}}_{t,m}^{(m,H)} = \frac{\pi_t^{m'}}{p_t^m} \sum_i \frac{\bar{z}_t^i}{f_t^{i'}} \int_{B_i} \tau h\left(\tau|\boldsymbol{x}_t^{m'}\right) d\tau, \qquad (4.64)
$$

$$
\tilde{\mathsf{R}}_t^{(m,H)} = \frac{1}{p_t^m} \mathsf{R}. \qquad (4.65)
$$

Substituting (4.64) and (4.65) into (4.52), we obtain

$$
\zeta_t^m(\boldsymbol{z}_t^i, \boldsymbol{x}_t^m) = \mathcal{N}\left(\tilde{\boldsymbol{z}}_t^{(m,H)}; \mathsf{H}\boldsymbol{x}, \tilde{\mathsf{R}}_{t,m}^{(H)}\right). \qquad (4.66)
$$

The TkBD problem can be reposed into an equivalent point-measurement problem consisting of the same target state model and of a measurement model, which is scaled by the TkBD psf. It is important to emphasize that this is not an approximation. By completing the square, analytic expressions are obtained for a point measurement that has the same likelihood as the sensor image (to within a constant) and is assumed equivalent for the purposes of state estimation.

Next we develop analytic expressions to evaluate the target and clutter per-cell contributions defined in (4.9) and (4.10) respectively. First, we make the assumption that the state space grid is uniform resulting in the spatial extent for every B_i being the same. The contribution of each target within each pixel is simply the area under the pdf in pixel i. For a Gaussian function, this can be evaluated using *erf* functions, however this is generally expensive to compute. For scenarios in which the target contribution to the image is assumed to be changing slowly from pixel to pixel, we can make the approximation that the function $h\left(\tau|\boldsymbol{x}_t^m\right)$ is constant over the area B_i. This can be done by assuming that the function across B_i is equal to the psf evaluated

at the centre of the pixel,

$$\begin{aligned} h^i\left(\boldsymbol{x}_t^m\right) &= \int_{B_i} h\left(\tau|\boldsymbol{x}_t^m\right) d\tau \\ &\approx \frac{\Delta_\tau}{|2\pi\mathsf{R}|} \exp\left\{-\frac{1}{2}(\mathsf{H}\boldsymbol{x}_t^m - \bar{\tau}_i)^\mathsf{T}\mathsf{R}^{-1}(\mathsf{H}\boldsymbol{x}_t^m - \bar{\tau}_i)\right\} \end{aligned} \tag{4.67}$$

where $\bar{\tau}_i$ denotes centre of pixel i and Δ_τ the cell area. In order to evaluate the clutter per-cell contribution, assume that the clutter model is uniform across the measurement image:

$$h^i\left(\emptyset\right) = \frac{1}{I} \tag{4.68}$$

To evaluate the integral in the synthetic measurement (4.64), consider the derivative of the psf $h\left(\tau|\boldsymbol{x}_t^m\right)$. As the psf is Gaussian,

$$\frac{d\, h\left(\tau|\boldsymbol{x}_t^m\right)}{d\tau} = h\left(\tau|\boldsymbol{x}_t^m\right)\left(\mathsf{H}\boldsymbol{x}_t^m - \tau\right)^\mathsf{T}\mathsf{R}^{-1}. \tag{4.69}$$

We can then write

$$\tau h\left(\tau|\boldsymbol{x}_t^m\right) = \mathsf{H}\boldsymbol{x}_t^m h\left(\tau|\boldsymbol{x}_t^m\right) - \mathsf{R}\frac{d\, h\left(\tau|\boldsymbol{x}_t^m\right)}{d\tau}. \tag{4.70}$$

Substituting (4.70) back into the synthetic measurement gives

$$\begin{aligned} \tilde{\boldsymbol{z}}_t^{(m,H)} &= \frac{\pi_t^{m'}}{p_t^m}\sum_i \bar{\boldsymbol{z}}_t^i \int_{B_i} \tau h\left(\tau|\boldsymbol{x}_t^{m'}\right) \\ &= \frac{\pi_t^{m'}}{p_t^m}\sum_i \bar{\boldsymbol{z}}_t^i \frac{\mathsf{H}\boldsymbol{x}_t^m}{h^i(\boldsymbol{x}_t^m)} \int_{B_i} h(\tau|\boldsymbol{x}_t^m)d\tau - \frac{R}{h^i(\boldsymbol{x}_t^m)}\left(\frac{d}{d\tau}\int_{B_i} h\left(\tau|\boldsymbol{x}_t^m\right) d\tau\right) \\ &= \frac{\pi_t^{m'}}{p_t^m}\sum_i \bar{\boldsymbol{z}}_t^i \left[\mathsf{H}\boldsymbol{x}_t^m - \frac{R}{h^i(\boldsymbol{x}_t^m)} h\left(\tau|\boldsymbol{x}_t^m\right)\right] \\ &= \frac{\pi_t^{m'}}{p_t^m}\sum_i \bar{\boldsymbol{z}}_t^i \left[\mathsf{H}\boldsymbol{x}_t^m - \frac{R}{h^i(\boldsymbol{x}_t^m)}\left\{h\left(\bar{\tau}_i + \frac{\Delta_\tau}{2}\middle|\boldsymbol{x}_t^m\right) - h\left(\bar{\tau}_i - \frac{\Delta_\tau}{2}\middle|\boldsymbol{x}_t^m\right)\right\}\right]. \end{aligned} \tag{4.71}$$

In the single target case, let $\pi_0^\emptyset$ and π_0^x denote the initial mixing proportion terms for the clutter and target components, respectively, such that,

$$\pi_0 = [\ \pi_0^\emptyset \quad \pi_0^x\ ], \tag{4.72}$$

If there are no targets detected at the beginning of the scenario, let $\pi_0^\emptyset = 1$ and $\pi_0^x = \emptyset$. When a target is detected, let $\pi_0^x = 0.01$ and π_t is renormalised accordingly.

In the multi-target case, priority is given to detections and tracks that have higher SNR. In the initialisation step, the mixing terms for each new target are given a initial mixing proportion value of $\pi_{tim}^m = 0.01$, appended to the vector π_t, and π_t is re-normalised. This initialisation

procedure guarantees that each time scan, tracks that are confirmed first are given higher probabilities of existence, and are given preference over tracks which are confirmed later.

The KF implementation of the H-PMHT algorithm is summarised in Algorithm 5 on page 86. In [36], Davey extended this procedure to account for a Gaussian psf where the peak is a non-linear function of the target state,

$$h(\tau|\boldsymbol{x}) = \mathcal{N}(\tau; g(\boldsymbol{x}_t^m), \mathsf{R}). \tag{4.73}$$

Davey demonstrated how this may also be rearranged as an equivalent point-measurement likelihood by substituting in the non-linear Gaussian psf, expanding and then completing the square to result in the equivalent point measurement function of the form,

$$\zeta_t^m(\boldsymbol{z}_t^i, \boldsymbol{x}_t^m) = \mathcal{N}\left(\tilde{\boldsymbol{z}}_t^{(m,H)}; g(\boldsymbol{x}_t^m), \tilde{\mathsf{R}}_t^{(m,H)}\right). \tag{4.74}$$

As the estimation problem is now non-linear, an EKF, UKF or PF could be used to obtain the state estimates. The extension of the H-PMHT to accommodate an EKF or UKF is relatively simple, however these implementations may diverge under highly non-linear conditions and fail to represent multi-modality. The PF performance is generally more robust in highly non-linear scenarios and its implementation for the H-PMHT is discussed in the next section.

4.3.2 Particle Filter Implementation

The PF uses the sampling approach to solve the dynamical state estimation problem. The PF is founded upon Sequential Monte Carlo estimation (SMC) and is based on the idea that a probability density can be represented by a set of random samples or 'particles' with associated weights. In the limit, as the number of samples become very large, the particle representation of the pdf becomes the equivalent functional form of the pdf, and the particle filter approaches the optimal Bayesian estimator.

To avoid degeneracy, in which all but one particle has a negligible normalised weight after a few recursions [26], we employ resampling to discard particles of negligible weight and duplicate those particles which contribute most to the approximation of the posterior density.

In the general non-linear non-Gaussian case, the process model $\psi_t(\cdot)$ given by (4.56) is a non-linear function with non-Gaussian noise and the measurement function $\zeta_t(\cdot)$ is the sum of logarithms of arbitrary functions with no closed form. Nevertheless, Davey demonstrated that a solution is still possible using a PF implementation for the H-PMHT state estimation [36].

The initialisation step for the target states is similar to that required for a PF TkBD method (e.g. see [108]) and only requires that the initial sample spread has enough diversity to support

Algorithm 5 H-PMHT KF

1. Initialise the EM algorithm: For each target $m = 1 \ldots, M$, initialise the algorithm from a known $p(\boldsymbol{x}_0^m)$.

2. Initialise the mixing proportion estimate using an initial estimate for $\boldsymbol{\pi}^m$.

3. For time scans $t = 1, \ldots, T$,

   (a) Prediction Step: For each target component m, compute the predicted state estimate $\boldsymbol{x}_{tt-1}^m$ by applying the dynamics model to the previous EM state estimates $\boldsymbol{x}_t^{m'}$.

   (b) For each target component m, construct $h^i(\boldsymbol{x}_{t|t-1}^m)$, the probability that a shot due to the target falls in pixel i. Note that this step may require numerical integration.

   (c) Evaluate $f^i\left(\boldsymbol{x}_{t|t-1}^{1:M}; \boldsymbol{\pi}_{t|t-1}^{0:M}\right)$, the overall probability that a shot (due to target or clutter) falls in pixel i.

   (d) Refine the mixing proportion estimates $\boldsymbol{\pi}_t^{0:M}$ using the above values of $h^i(\boldsymbol{x}_{t|t-1}^m)$ and $f^i\left(\boldsymbol{x}_{t|t-1}^{1:M}; \boldsymbol{\pi}_{t|t-1}^{0:M}\right)$.

   (e) Update Step

      i. Compute the updated state estimates $\boldsymbol{x}_{t|t}^m$ using synthetic measurements $\tilde{z}_t^{(m,H)}$ in (4.64).

      ii. Compute the synthetic covariance $\tilde{\mathsf{R}}_t^{(m,H)}$ via (4.65).

   (f) Repeat Step 3.b) ... 3.e) until convergence using the updated EM state estimates.

the range of possible target states. The prediction step is also similar to the PF TkBD and involves predicting the posterior particle set forward in time using the target dynamics model. The initialisation procedure for the mixing terms is the same as for the KF implementation of the H-PMHT.

For the state estimation step, we are required to calculate the likelihood of the sensor data given the value of the state for every particle. For notational convenience, we suppress the target index m in the following equations. Let $\boldsymbol{x}_t^j$ denote the j^{th} sampled state for a given target m. Then, the likelihood for target m in pixel i is defined by (4.52) as

$$\begin{aligned} \log\{w_t^{ij}\} &= \sum_i \pi_t' \frac{\bar{z}_t^i}{f_t^{i'}} \int_{B_i} h\Big(\tau|\boldsymbol{x}_t'\Big) \log\left\{ h\Big(\tau|\boldsymbol{x}_t^j\Big) \right\} d\tau, \\ &= \sum_i \pi_t' \frac{\bar{z}_t^i}{f_t^{i'}} W_i\left(\boldsymbol{x}_t', \boldsymbol{x}_t^j\right). \end{aligned} \tag{4.75}$$

where w_t^{ij} denotes the weight for j^{th} particle in pixel i and

$$W_i\left(\boldsymbol{x}_t', \boldsymbol{x}_t^j\right) = \int_{B_i} h\Big(\tau|\boldsymbol{x}_t'\Big) \log\left\{ h\Big(\tau|\boldsymbol{x}_t^j\Big) \right\} d\tau. \tag{4.76}$$

Also, observe that the state estimate $\boldsymbol{x}_t'$ and the π_t' term are the result of the previous EM iteration and is common for all particles generated for target m. The sampled state for the jth particle enters through the logarithm term.

Unfortunately, even though we have gone to a particle representation, the evaluation of the integral remains intractable. However, if we treat the particle state $\boldsymbol{x}_t^j$ as a known constant, this allows us to numerically evaluate the integrand of (4.75) [36]. Let the area of cell i (i.e. B_i) be broken up into D contiguous subcells. The integrand can be approximated by a piecewise constant function and the integral can be approximated with a Reimann sum. Let the area of subcell d be denoted by Δ_d and the centre of subcell d as τ_d, then we can make the following approximation,

$$W_i\left(\boldsymbol{x}_t', \boldsymbol{x}_t^j\right) \approx \sum_{d=1}^{D} h\Big(\tau_d|\boldsymbol{x}_t'\Big) \log\left\{ h\left(\tau_d|\boldsymbol{x}_t^j\right) \right\} \Delta_d. \tag{4.77}$$

The limit as $D \to \infty$ of (4.77) is the Reimann definition of the integral in (4.75). The expression in (4.77) is of a form readily implemented in software. Note that the first term in (4.77) is not dependent on the particle state, thus to ensure an efficient implementation, these terms can be calculated outside the particle loop. Finally, the target state estimates can be calculated using,

$$\hat{\boldsymbol{x}} = \sum_{j=1}^{N_p} \sum_{i=1}^{I} w^{ij} \boldsymbol{x}_t^j. \tag{4.78}$$

Note that the EM process requires a maximum a posteriori (MAP) estimate on each iteration however it is difficult to extract this from a set of particles. That being the case, we have assumed that the conditional mean estimate produced by the particle filter implementation is a sufficient approximation of the MAP estimate required by the EM. The recursive form of the PF implementation for the H-PMHT algorithm is summarised in Algorithm 6.

Algorithm 6 H-PMHT PF

1. Initialise the EM algorithm: For each target $m = 1 \ldots, M$ draw N_p independent sample positions from some prior distribution $p(\boldsymbol{x}_0^m)$ and set the initial weights to be uniform.
2. Initialise the mixing proportion estimate using an initial estimate for $\boldsymbol{\pi}^m$.
3. For time scans $t = 1, \ldots, T$, perform steps 3.(a) - 3.(d) from Algorithm 5.
   (e) Update Step:
      i. Re-weight particles by evaluating (4.75).
      ii. Define a new state estimate based on the weighted particle set
   (f) Repeat Step 3.b) ... 3.e) until convergence using the updated EM state estimates.
   (g) Resample particles for each target.

4.3.3 Viterbi Algorithm Implementation

Up until this point, we have presented a summary of the derivation of the H-PMHT and described how it can be implemented using a KF for linear Gaussian problems. We also showed that in the general non-linear non-Gaussian case, the PF can be used as a numerical approximation to solve for the state estimates. However, there is nothing preventing us from using other numerical approximations based on fixed grid techniques such as the Viterbi algorithm. We now describe how the general Viterbi algorithm can be applied to solve the optimisation given in (4.54). The ideas introduced in this section comprise the key contribution for this chapter [133]. For a review of the Viterbi algorithm, refer to subsection 3.4.1.

The PF implemented in previous work is a recursive filter and produces a conditional mean state estimate after each EM iteration through use of a dynamic grid. On the other hand, the Viterbi algorithm is a batch estimator and produces a maximum likelihood sequence estimate. It does so by performing a discretisation over the entire state space resulting in a fixed grid with

time. For the Viterbi implementation, we now define $\boldsymbol{x}_t^j$ to be a distinct state in the Viterbi grid.

The advantage of using a fixed-grid over the particle filtering approach is that the method is more robust to clutter outliers that arise in real-world data [35] since it does not suffer from sample degeneracy. Also unlike the PF implementation, which outputs a conditional mean estimate, the Viterbi implementation naturally generates a MAP estimate, as required by the EM procedure. Finally if batch processing is desired, the Viterbi algorithm can return a smoothed sequence, whereas the particle approach presented in [36] is only suited to filtering.

The initialisation step is similar to that required for the standard TkBD version [133], and requires the evaluation of an initial likelihood. Alternatively, initialisation can also be performed via a peak detection process where a threshold is applied to the measurement image at every time scan and a track is initiated for every point which falls above that threshold. This threshold can be set by the user and is dependent on the scenario under consideration. Note that after initialisation via the peak detection process, track updates are performed using the raw measurement image. The H-PMHT batch algorithm requires an initial estimate of the entire state sequence rather than just a state estimate at time $t = 0$. Hence, an initialisation procedure is required for all peak detections for all time points $1, \ldots, T$. This can be done by running a standard Viterbi algorithm across the image to produce an initial sequence estimate for each peak detection, which can be used as input into the batch form H-PMHT. The initialisation procedure for the mixing terms is the same as for the KF implementation of the H-PMHT.

The H-PMHT employs the Viterbi algorithm to perform the state estimation component of the EM procedure. Define $\zeta_t(\boldsymbol{x}_t^j)$ to be the likelihood for state $\boldsymbol{x}_t^j$ given the sensor data such that the log-likelihood via (4.52) is given by

$$\begin{aligned} \log\{\zeta_t(\boldsymbol{x}_t^j)\} &= \sum_i \pi_t' \frac{\bar{z}_t^i}{f_t^{i\prime}} \int_{B_i} h\Big(\tau|\boldsymbol{x}_t'\Big) \log\left\{ h\Big(\tau|\boldsymbol{x}_t^j\Big) \right\} d\tau \\ &= \sum_i \pi_t' \frac{\bar{z}_t^i}{f_t^{i\prime}} W_i\left(\boldsymbol{x}_t', \boldsymbol{x}_t^j\right), \end{aligned} \tag{4.79}$$

where W_i is given by (4.76) and the quantity $\boldsymbol{x}_t^j$ in the logarithm term now denotes a distinct state in the Viterbi grid. Again, the state estimate $\boldsymbol{x}_t'$ and mixing proportion term π_t' take their values from the previous EM iteration and remain the same when iterating though all states in the Viterbi grid. By assuming that the grid point $\boldsymbol{x}_t^j$ is a known constant, the integrand in (4.79) can be evaluated numerically in the same way as (4.77).

This Viterbi algorithm finds the best path through the trellis generated by the fixed grid. It does so by using the likelihood to evaluate the cost of transitioning from one state to the next at each time point. Recall that the cost function $C_t(\boldsymbol{x}_t^j)$ is defined to be the best score over all

paths that end in state $\boldsymbol{x}_t^j$ from time $1, \ldots, t$ and the back-pointer array, $\theta_t(\boldsymbol{x}_t^j)$ stores the state that maximised the cost function for each time step. Given these functions, the batch form of the Viterbi H-PMHT algorithm is summarised in Algorithm 7.

For multi-target tracking, fixed-grid approximation techniques can grow exponentially in computational complexity due to the large number of discrete states required. However the H-PMHT provides data association weights that allows each target to be estimated independently using a bank of single-target Viterbi estimators.

However, the performance of the single-target Viterbi algorithm is still highly dependent on the discretisation of the state space. For a discretisation that results in a fine resolution grid, performance can be costly as we are required to calculate the likelihood for every state in the grid. This is inefficient as the likelihood is negligible except for grid points that are near the previous EM's estimate for $\boldsymbol{x}_t'$. To reduce computational cost at each time scan, we can evaluate the likelihood for a subset of grid points near the previous EM's estimate $\boldsymbol{x}_t'$. In effect, this restricts the target updates to only close neighbouring states in the grid. The subset of grid points will obviously change as the target is updated at each time scan. This results in an *adaptive grid* from one EM iteration to the next, in which the total subset of grid points considered to update the entire target trajectory will change accordingly as each individual target state is updated at each time scan. In particular, the use of an adaptive grids may prove necessary in multi-target scenarios to reduce the overall computational expense.

In the next section, a comparison of the Viterbi-based H-PMHT performance with KF and PF H-PMHT implementations is shown for several simulated scenarios.

4.4 Simulations

The batch Viterbi H-PMHT will be now compared with the PF H-PMHT and KF H-PMHT through some idealised single target simulation scenarios. Two cases are considered: a linear target motion with a Gaussian psf and a linear target motion with a non-linear non-Gaussian psf. For brevity, the three algorithms will be referred to as H-PMHT-V, H-PMHT-P and H-PMHT-K from herein.

It is clear that the first case satisfies the assumptions leading to the H-PMHT-K so we would expect it to perform well. The second simulation scenario will demonstrate a simple example where the H-PMHT-V will be shown to give some benefit over the H-PMHT-K. Some insight into the performance of differing non-linear estimators as applied to the H-PMHT algorithm is also shown. The performance for all three algorithms will be evaluated using the root mean square (RMS) position accuracy.

Algorithm 7 H-PMHT Viterbi

1. Initialise the EM algorithm: For each target $m = 1 \ldots, M$, initialise the EM algorithm by generating an initial state sequence estimate $p(\boldsymbol{x}_{1:T}^m)$ using the likelihood or peak detection process.

2. Initialise the mixing proportion estimate using an initial estimate for $\boldsymbol{\pi}^m$.

3. For time scans $t = 1, \ldots, T$, perform Steps 3(a) - (d) from Algorithm 5.

   (e) Update Step:

      i. Calculate the likelihood for each state $\boldsymbol{x}_t^j$ in the Viterbi grid using (4.79).

      ii. Calculate scoring quantities $C_t(\boldsymbol{x}_t^j)$ and back pointer array $\theta_t(\boldsymbol{x}_t^j)$ for each state grid point $\boldsymbol{x}_t^j$.

4. Find the most likely state estimate at final time T.

5. Using the back pointer array $\theta_t(\boldsymbol{x}_t^j)$, reconstruct the most likely state sequence.

6. Repeat Steps 3 - 5 until sequence converges using the updated EM sequence estimates.

For each algorithm, we assume a four element state consisting of position and velocity in the plane,

$$\boldsymbol{x}_t = \begin{bmatrix} x_t & \dot{x}_t & y_t & \dot{y}_t \end{bmatrix}^{\mathsf{T}}. \tag{4.80}$$

In the case of the Viterbi-based implementation, we assume a uniform grid and specify the mapping of index space into physical space using the vector $\boldsymbol{x}_t^j = [j_x \; j_{\dot{x}} \; j_y \; j_{\dot{y}}]'$ such that

$$\boldsymbol{x}_t^j \;=\; \begin{bmatrix} \triangle_x j_x & \frac{\triangle_x}{\triangle_t} j_{\dot{x}} & \triangle_y j_y & \frac{\triangle_y}{\triangle_t} j_{\dot{y}} \end{bmatrix}', \tag{4.81}$$

where $\triangle_t$ refers to duration between time scans, and $\triangle_x$ and $\triangle_y$ denote the size of the cells in the X and Y dimensions respectively.

For each of the scenarios, the sensor collected a 100×100 pixel image of a single target moving in the plane for $T = 40$ time frames. The frames were collected at a uniform rate of one per second. We assume a point-scatterer target such that the target contribution to the measurement image can be described purely in terms of the sensor psf, $h(\boldsymbol{x}_t)$,

$$\mathbf{Z}_t = A_t \; h(\boldsymbol{x}_t) + \boldsymbol{w}_t, \tag{4.82}$$

where $\boldsymbol{w}_t$ is additive i.i.d. sensor noise process representing uncertainties in the measurement process and A_t denotes the amplitude at time t. The sensor noise was complex-Gaussian with

unit variance. However, the H-PMHT algorithm assumes non-negative real values for each pixel, so the envelope (absolute value) of the image was supplied to the tracking algorithm. Thus the noise-only pixels were Rayleigh distributed with unit variance and the pixels containing a target contribution followed a Rician distribution with unit variance and a mean dependent on the target location and SNR. Note that the psf is a property of the sensor and is the same for all targets, but can vary with different sensors. In subsequent sections, the performance of the H-PMHT is demonstrated in scenarios featuring a Gaussian and non-Gaussian psf. Note that as the measurement function is only dependent on position, the likelihoods for each state grid point are the same for each velocity component.

The computational cost and performance of the H-PMHT-V will be highly dependent on the resolution of the Viterbi grid. In the simulations, a uniform unit integer grid in position and velocity, in both dimensions was assumed. Thus in the following scenarios, the Viterbi grid size consisted of approximately 40 position elements and two velocity elements in both the X and Y direction i.e. $N_x = N_y = 40$ and $N_{\dot{x}} = N_{\dot{y}} = 2$, where $\dot{x}_t = \dot{y}_t = \{0, 1\}$, resulting in a total grid size of 6400 points. For the simulations considered in this chapter, it was found that ten iterations was adequate to ensure EM convergence.

The H-PMHT-P algorithm employs random sampling and as such, the total number of particles is generally dependent on how non-linear a problem is. For the following scenarios, it was found to be sufficient to test the PF method with 1000 particles.

4.4.1 Linear Gaussian Scenario

The first scenario consists of a target under constant velocity motion with the assumption of an isotropic Gaussian psf. Figure 4.1 shows the target trajectory for the scenario. The target starts near the corner of the image at approximately (20,20) and moves with a constant speed of one pixel per frame with a heading of 45 degrees.

The psf $h(\boldsymbol{x}_t)$ forms an ellipse based on a two dimensional Gaussian function centred at (x_t^*, y_t^*) such that,

$$h(\boldsymbol{x}_t) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{ -\left[\left(\frac{x_t - x_t^*}{2\sigma_x^2} \right)^2 + \left(\frac{y_t - y_t^*}{2\sigma_y^2} \right)^2 \right] \right\}, \tag{4.83}$$

where $\sigma_x = 4$ and $\sigma_y = 6$ denote the standard deviation in the X and Y direction, respectively.

As we are using a four element state space consisting of position and velocity in the plane and a nearly-constant velocity target model, let $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \sim \mathcal{N}(\boldsymbol{x}_t; \mathsf{F}\boldsymbol{x}_{t-1}, \mathsf{Q})$. The parameters of the model are given by,

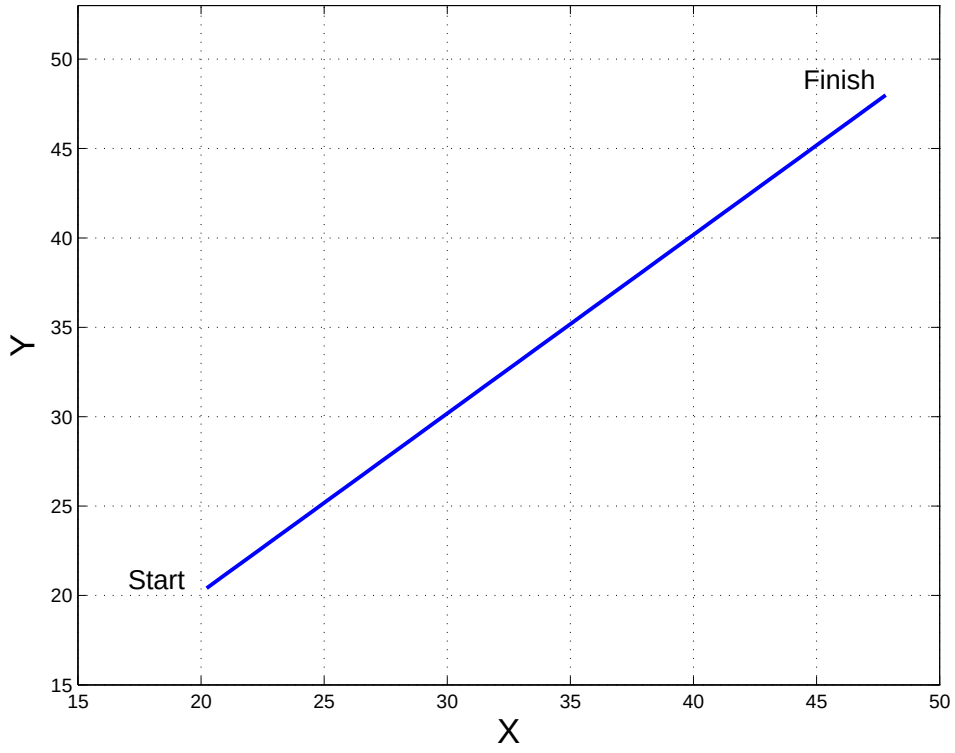


Figure 4.1: Linear Gaussian scenario

$$\mathsf{F} = \begin{bmatrix} 1 & \triangle_t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \triangle_t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathsf{Q} = 0.1 \begin{bmatrix} \frac{\triangle_t^3}{3} & \frac{\triangle_t^2}{2} & 0 & 0 \\ \frac{\triangle_t^2}{2} & \triangle_t & 0 & 0 \\ 0 & 0 & \frac{\triangle_t^3}{3} & \frac{\triangle_t^2}{2} \\ 0 & 0 & \frac{\triangle_t^2}{2} & \triangle_t \end{bmatrix},$$

where $\triangle_t = 1$ second.

For efficient implementation, an adaptive grid for the Viterbi algorithm was employed so that at each time scan, the likelihood was evaluated for only a 10×10 subset of grid points around the previous EM target estimates. This restricts the possible target updates within each EM iteration to only close neighbouring states. The H-PMHT-K and H-PMHT-P use the true target dynamics model directly; the H-PMHT-V must use a discretised version of it. This is achieved by direct sampling, that is, the transitions from state S_p to state S_q are given by $a_{pq} = \mathcal{N}(S_q - FS_p; \mathsf{0}, \mathsf{Q})$. To improve computation costs, the support of the transition kernel was limited to only adjacent states. Thus the transition matrix F was renormalised so that $a_{pq} = 0$ for non-adjacent states.

Figure 4.2 shows the RMS position estimation accuracy for the three implementations of H-PMHT averaged over 100 Monte Carlo trials. All three algorithms perform well but as ex-

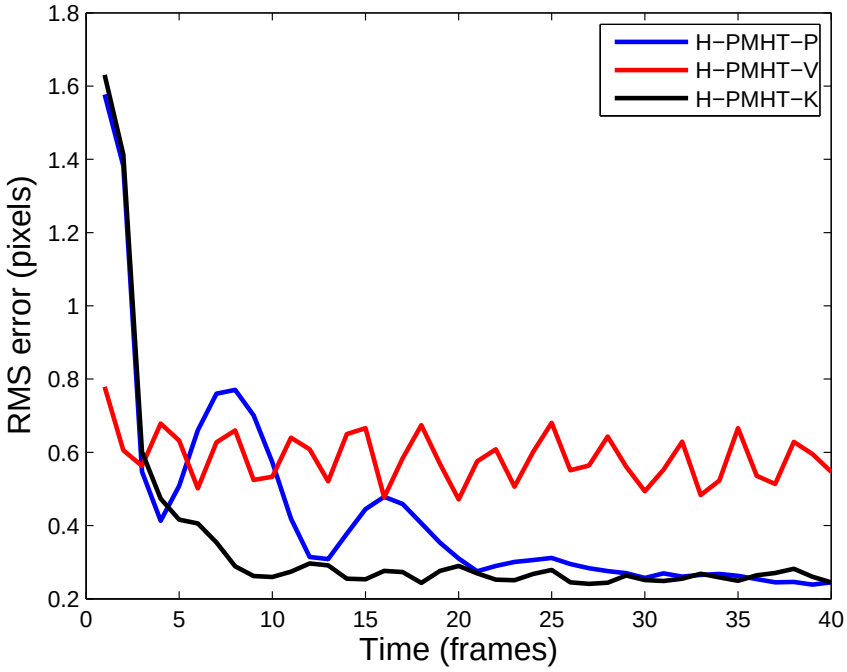


Figure 4.2: Localisation accuracy, linear Gaussian scenario

pected, the H-PMHT-K gives the best performance as it is the optimal Bayesian estimator for the linear Gaussian case. The H-PMHT-V does not perform as well as the H-PMHT-P or the H-PMHT-K and its RMS curve shows distinct fluctuations in error with time. In the Viterbi implementation, the errors between the true state and the Viterbi estimate are limited to the resolution of the Viterbi grid. As the Viterbi algorithm produces a smoothed trajectory, these differences are expected to fluctuate over time as the Viterbi state estimates do not always select the state grid point which is closest to the true state. These fluctuations in errors are a characteristic of the Viterbi algorithm and will vary in size depending on the resolution of the Viterbi state space. The observed fluctuations in error have a magnitude of approximately half a pixel, which are acceptable in light of the approximation of the state space to a unit integer grid. Large state estimation errors can also arise as the Viterbi algorithm is required to conform to a target dynamics model. In some cases, a physical transition to the the grid point closest to the true state maybe be infeasible under the target dynamics model if the target estimate has deviated significantly from the true state. This scenario verifies that the H-PMHT-V gives an appropriate answer for the simplest case.

4.4.2 Linear Non-Gaussian Scenario

In the second scenario, the target motion model remained the same as the first, but a non-Gaussian psf was selected. For this scenario, a psf somewhat resembling the letter 'C' was chosen. Specifically, the psf in polar coordinates (r^p, θ) is given by

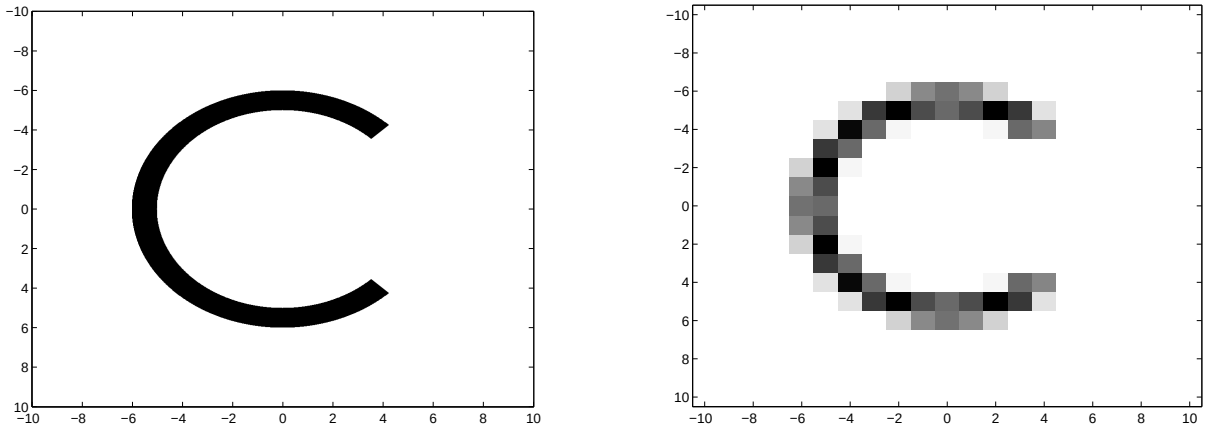
$$h(r^p, \theta) = \begin{cases} A & \text{if } 5 \leq r^p \leq 6 \text{ and } |\theta| > \frac{\pi}{4}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.84}$$

where A is a normalising constant. This response is shown in Figure 4.3(a). A approximation of the psf can be used to compute likelihoods across the pixel image, where the contribution of the target to each pixel is the integral of $h(r^p, \theta)$ over that pixel. An example of this is shown in Figure 4.3(b).

This artificial psf was chosen because it is asymmetrical and the mean of the distribution is at a location of very low density. For the H-PMHT-K implementation, the psf was approximated by a Gaussian with covariance $\mathsf{diag}(18.4, 9.2)$, which is the covariance of $h(r^p, \theta)$. Note that because the psf is asymmetrical, the mean of the psf is not coincident with the target state. The H-PMHT-K can easily compensate for this since the offset is fixed and known: it can be treated as a bias.

The H-PMHT-P and H-PMHT-V have a more difficult problem to overcome. The psf has a discontinuity and is also identically zero over most of the measurement space. This will result in a log-likelihood value of $\log\{0\}$ for each particle or grid-point that was not part of the original pixel support at the previous EM iteration. The log-likelihood associated with such a psf is unbounded and the EM will struggle to optimise over this function. Some regularisation steps proposed by [36] can be used to overcome this problem. First, we can convolve the psf with a Gaussian with relatively small variance to effectively "blur" the psf and remove the discontinuity. A uniform pedestal can also be added to the psf to prevent numerical problems associated with evaluating $\log\{0\}$.

Figure 4.4 on page 97 shows the overall RMS errors for the three algorithms averaged over 100 Monte Carlo trials for the linear non-Gaussian scenario. Again observe that the H-PMHT-P gives very good state estimation performance. The H-PMHT-V RMS does not perform as well as the H-PMHT-P, and again its performance tends to fluctuate with time. As mentioned earlier, this saw-tooth characteristic can be attributed to the discretisation of the state space. Note that if the target truth lies outside the trellis path generated by the state discretisation step, then clearly the H-PMHT-V smoother will be unable to estimate target states as well as the PF as it approximates the target state to the nearest state grid point. This results in a lower bound on estimation performance. In this case, an improvement in the H-PMHT-V performance can



(a) Continuous-domain target psf

(b) Target response per pixel

Figure 4.3: Non-Gaussian point spread function

possibly be achieved with the use of a finer resolution state grid, at the additional expense of computational complexity.

It is clear that the H-PMHT-K is biased even with compensation for the asymmetry in the psf. Figure 4.5 and Figure 4.6 on pages 97 and 98, respectively, show the errors (in terms of pixels) in the X and Y component respectively. It can be seen that the bias is attributed to the Y component. This bias is not present in the H-PMHT-P or the H-PMHT-V and leads to higher estimation error for H-PMHT-K.

4.5 Limitations

The H-PMHT's unique definition of the measurement model provides data association weights that accounts for the interaction between closely-spaced targets, which results in a TkBD multi-target algorithm that retains linear complexity with the number of targets. Other multi-target algorithms retain linear complexity by assuming that the targets are either well separated or deal with the interaction between closely-spaced targets in a heuristic manner. The H-PMHT's ability to compute fast estimates in the multi-target context is one of its most attractive properties however the H-PMHT algorithm does have its limitations. These limitations are discussed in the following subsections.

4.5.1 Unsmoothed Mixing Proportion Estimate

At each time step, the EM procedure for the multinomial mixing proportions is initialised using the estimate calculated at the previous time point. However the EM update of the mixing

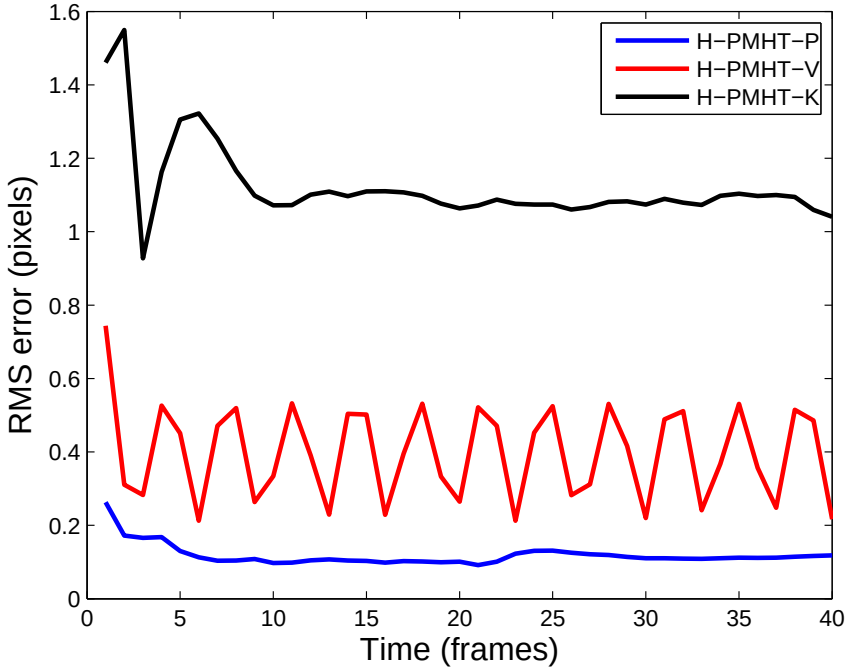


Figure 4.4: Localisation accuracy, linear non-Gaussian scenario

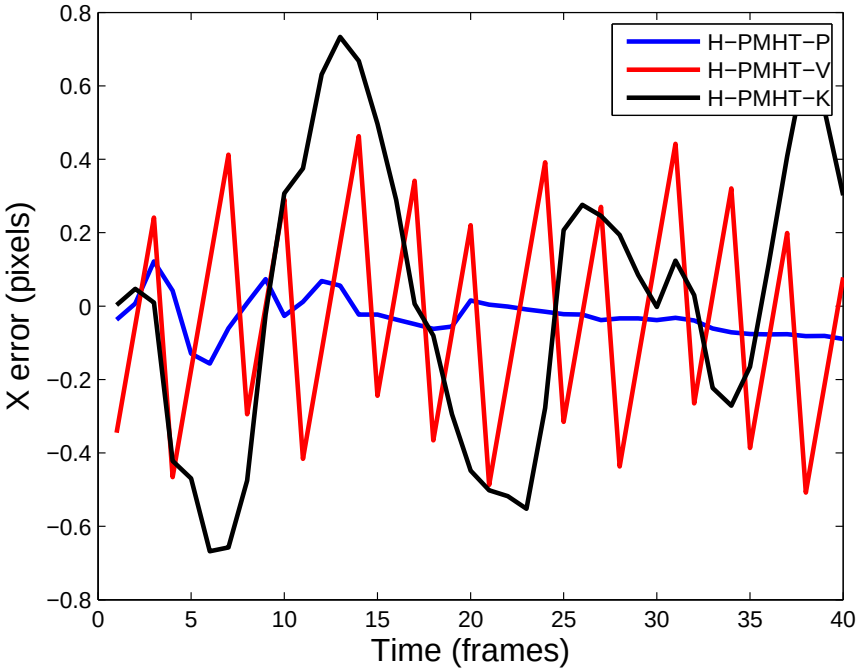


Figure 4.5: Localisation error in X component, linear non-Gaussian scenario

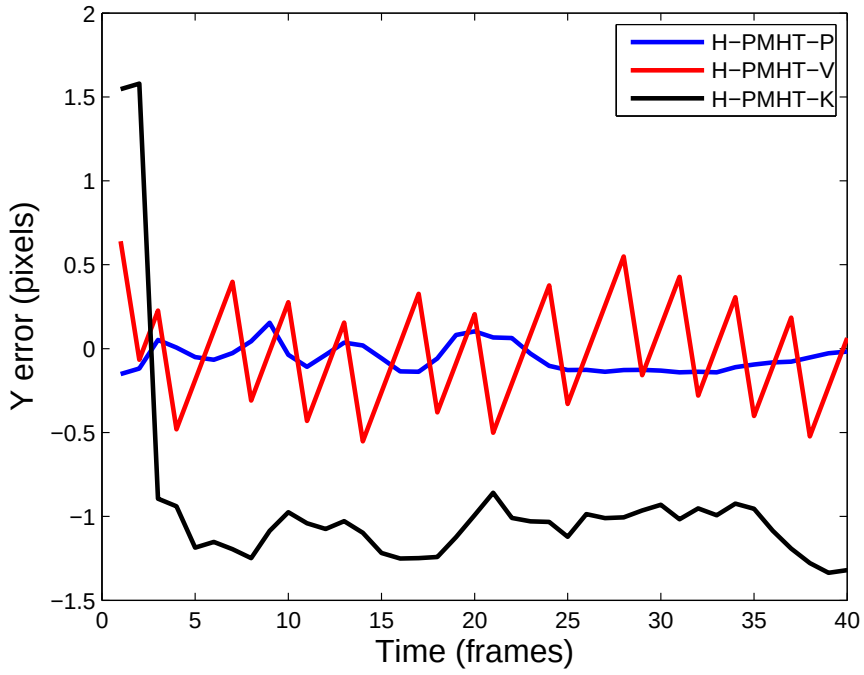


Figure 4.6: Localisation error in Y component, linear non-Gaussian scenario

proportion values given by (4.60), is clearly only dependent on the mixing proportion value at the previous EM iteration. This results in an unsmoothed estimate of the component mixing proportions as they are uncorrelated with time. For a scenario featuring a target signal with high fluctuations, estimates for the underlying average target SNR using the H-PMHT would be difficult to obtain as the target mixing proportions are highly sensitive to the fluctuations in the observed target signal.

The component mixing proportions are also dependent on each other through a normalising constant, hence estimates for the mixing proportions must be determined jointly. The consequence of this is exponential complexity if one attempts to apply a state model directly to the mixing proportions [37]. Finally, note that the mixing proportions give an estimate of the target SNR relative to the measurement image size. Although this is not an issue in most scenarios, nevertheless, this requires the π values to be recalculated when the measurement size or resolution changes. This limitation is addressed in Chapter 5, in which an alternative measurement model based on a Poisson assumption on the quantised measurement counts is proposed. Under this new model, the H-PMHT can be rederived to incorporate a time-correlated estimate for the component mixing proportions, resulting in an improved measure for track quality.

4.5.2 Quantisation issues

The H-PMHT algorithm is a TkBD method which returns tracks by accessing the intensity map data directly. Generally, there exists no standard multi-target TkBD measurement model that can be used to describe the intensity map image. As a result, it is impossible to perform filtering in the usual sense without applying an approximation to reduce the intensity map data to a set of discrete point measurements.

The H-PMHT algorithm makes an approximation to the intensity map by performing a quantisation over the measurement image, and assuming a multinomial distribution on the resulting counts from the integer-valued image. This gives rise to the assumption of independence between shot measurements, which is questionable as the counts are derived from quantised intensities. The quantisation level c^2 influences the counts, and under the independence assumption, dictates the amount of information created in the synthetic measurement data generation, yet the true information content is not related to c^2 at all. The arbitrary choice of the quantisation level combined with the assumption of independence between shots results in an infinite amount of synthetically generated data when the limit of the quantisation is taken to zero. This is clearly undesirable as there is only a finite amount of information in the measurement data.

The process of taking the quantisation limit to zero also has other consequences when a Bayesian model is adopted: the infinite amount of synthetically generated data will overwhelm any prior. In the standard H-PMHT, this problem is solved by employing a modified Bayesian prior (5.10) to ensure that the prior has sufficient influence on the target estimates. However, this modification introduces some irregularities. At the end of the H-PMHT derivation, this modification results in the prior being scaled by a $||\mathbf{Z}_t||$ term in (4.52), which can have some dubious effects during filtering. For example, if the sensor surveillance region were to increase from one time scan to the next, this would result in an increase in $||\mathbf{Z}_t||$, the total energy received from the image. Under these circumstances, the data dependent prior also becomes larger, which affects the overall state estimation process. Thus, considering extra and possibly irrelevant pixels in the image can result in a change in the target estimation performance, even though the observed target itself has not changed.

The introduction of a data dependent prior can also have some adverse affects on the process noise covariance. Under the assumption of Gaussian process noise, the data dependent prior in (4.52) evaluates to,

$$\|\mathbf{Z}_t\| \log\{p(\boldsymbol{x}_t|\boldsymbol{x}_{t|t-1})\} \quad = \quad C - \frac{1}{2}(\boldsymbol{x}_t - \mathsf{F}\boldsymbol{x}_{t|t-1})^{\mathsf{T}}||\mathbf{Z}_t||\mathsf{Q}^{-1}(\boldsymbol{x}_t - \mathsf{F}\boldsymbol{x}_{t|t-1}) \qquad (4.85)$$

where C is a constant, and F and Q are the state transition and process noise covariance matrices, respectively. Assuming Gaussian noise results in a $||\mathbf{Z}_t||$ scaling term on the covariance

Q:

$$\tilde{\mathsf{Q}} = \frac{\mathsf{Q}}{||\mathbf{Z}_t||} \tag{4.86}$$

where $\tilde{\mathsf{Q}}$ is defined to be the synthetic process noise covariance matrix. It is clear that an increase in the total energy in the image would result in a decrease in the synthetic process covariance $\tilde{\mathsf{Q}}$. This is potentially problematic as an increase in $||\mathbf{Z}_t||$ can be attributed to an increase in target SNR. In this particular case, (4.86) dictates that the process covariance matrix will become smaller. A smaller $\tilde{\mathsf{Q}}$ implies that a filter will encounter difficulties when tracking a maneuvering target, which is counter-intuitive as a target with higher SNR should be easier to track.

In Chapter 5, we propose an alternative state prior density, which under Gaussian assumptions, which results in a more regularised form for the synthetic process noise matrix. In Chapter 6, we extend the work in Chapter 5 by extending the H-PMHT derived under a Poisson measurement model to an interpolated Poisson measurement model, which removes the requirement for quantisation and thus the need for a data dependent prior.

4.6 Summary

The H-PMHT algorithm is an efficient multi-target parametric mixture-fitting approach to the TkBD problem. In the past, the H-PMHT has been demonstrated to give performance comparable to numerical approximations to the Bayesian filter at a fraction of the processing cost. In this chapter, we provided a detailed review of the H-PMHT derivation and outlined the steps for an implementation via the KF for linear Gaussian scenarios. A review of the PF implementation of the H-PMHT for non-linear non-Gaussian applications was also presented.

The key contribution of this chapter was the first description for *evaluating a Viterbi-based H-PMHT algorithm*. Its performance was verified against the KF and PF implementations of the H-PMHT for both a linear Gaussian and a linear non-Gaussian scenario featuring a single target. The Viterbi-based implementation was shown to provide significant performance advantages over the commonly used KF implementation on scenarios featuring a non-Gaussian target. However the algorithm did not perform as well when compared to a PF H-PMHT implementation.

For scenarios in which the target dynamics is constrained to a set of fixed locations, a Viterbi-based implementation may be advantageous. The main drawback to a Viterbi-based implementation lies in the dependence of the algorithm's complexity on the discretisation of the state space. Theoretically, the complexity of the Viterbi algorithm increases linearly with the

number of observations and quadratically with the number of state space points. As a result, any extension to a multi-target scenario requires the use of adaptive grids, which only consider states in the fixed grid with non-negligible likelihoods. This ensures that the H-PMHT's linear complexity with the number of targets is retained.

The limitations of the H-PMHT algorithm were also discussed. These include possible adverse consequences on tracking performance when the quantisation limit is taken to zero, resulting from the H-PMHT's arbitrary choice in quantisation level and questionable independence assumptions. We also discussed the benefits of extending the H-PMHT algorithm to include a smoothed estimate for the component mixing proportions. A discussion of how these limitations can be addressed will be the focus of the following two chapters.

Chapter 5

H-PMHT with a Poisson Measurement Model

In this chapter, we address the main limitations of the H-PMHT algorithm discussed at the end of Chapter 4, namely, the lack of an unsmoothed mixing proportion estimate and the potential for inconsistent tracking performance after quantisation due to questionable independence assumptions.

The standard H-PMHT algorithm assumes a multinomial distribution on the quantised measurement counts, which results in an uncorrelated estimate of the component mixing proportions with time. We propose to address this limitation by assuming an alternative measurement model based on a Poisson distribution. The H-PMHT algorithm can then be re-derived to incorporate a time-correlated estimate of the component mixing terms, allowing for an improved measure for track quality. We also propose an alternative H-PMHT state prior density that depends only on the properties of that target: in the Gaussian case this allows for SNR independent filtering performance. The key contributions of this chapter are:

- the first derivation of the H-PMHT assuming a target generates a Poisson distributed random number of measurements, and
- the proposal of an alternate prior density for the H-PMHT that results in more consistent tracking performance.

The main contributions of this chapter are summarised in conference article [136] and journal submission [135].

5.1 Introduction

Recall that the quantisation step in the standard H-PMHT converts the continuous-valued measurement data in every pixel to an integer-value and interprets this integer as a count of a realisation of a point process. The counts in each pixel are modelled using a multinomial distribution and are compiled to create a synthetic histogram. The H-PMHT employs EM methods to estimate the target states as well as each target's contribution to the overall mixture model. The target mixing proportion terms can be interpreted as the received power from each target, and can be used as a natural test statistic for track quality to help distinguish true targets from clutter.

Under the multinomial model, the mixing proportion estimates are modelled as unknown parameters that can be time-varying or constant with time. The standard H-PMHT also assumes that the number of targets is known and remains invariant with time. In most practical applications, these assumptions are too restrictive as targets commonly appear and disappear from the surveillance region. The target SNR can also fluctuate with time making it difficult to estimate a target's underlying signal strength from a single scan. The inclusion of a smoothed mixing term estimate and a detection process for the birth and death of targets in the H-PMHT algorithm is desirable.

In conventional target tracking, a time-correlated estimate for the target SNR was introduced into the Probabilistic Multi-Hypothesis Tracker (PMHT) by imposing a dynamics model on the component mixing proportions [38]. Although an improvement in tracking accuracy was observed, the coupling of the mixing terms resulted in an exponential complexity with the number of targets. Recently, Davey [37] proposed an alternative derivation of the PMHT based on a Poisson distribution on the number of point measurements. Under a Poisson assignment model, Davey derived a time-correlated estimate based on [60], to more accurately estimate track existence. An important feature of this new algorithm is that it also retains linear complexity with the number of targets.

In this chapter, we extend Davey's work to TkBD applications by incorporating a Poisson measurement model into the H-PMHT algorithm. As the multinomial assumption in the standard H-PMHT is consistent with a Poisson Point Process [118], it is possible to re-derive the H-PMHT with a Poisson assumption on the number of quantised measurements generated by an individual target. This allows us to capitalize on the Poisson model's ability to impose a dynamics model on the component mixing rates, while still maintaining the original H-PMHT's linear complexity with the number of targets.

Another key limitation of the H-PMHT algorithm is the potential for inconsistent target track-

ing as the synthetic process noise covariance (4.86) derived for Gaussian targets is data dependent. This limitation is due to the assumption of independence between measurement shots and the modification to the state prior density to include a $||\mathbf{N}_t^{total}||$ term to ensure the prior is not overwhelmed by the measurement data. In this chapter, we propose to resolve this issue by modifying the H-PMHT state prior density to include a power to the $||\mathbf{N}_t^m||$ term. The important difference is that the state prior density for each component m, for $m = 0, \ldots, M$, now only incorporates the power from that target, not the total image power. When the limit of the quantisation is taken, the synthetic target process noise covariance derived under Gaussian assumptions is shown to have the same scaling factor as the synthetic measurement noise covariance. This new form for the process noise covariance ensures filtering performance is SNR independent and consistent performance is maintained when the sensor surveillance region is increased.

The key contributions of this chapter are the *first derivation and implementation of the H-PMHT assuming a Poisson measurement model and the proposal of an alternate state prior density to ensure more consistent tracking performance.*

It is possible to remove the measurement quantisation step completely and model the measurement image directly by assuming an interpolated Poisson distribution on the measured target energy. This idea is explored later in Chapter 6.

5.2 Relationship between Multinomial and Poisson Distributions

The standard H-PMHT algorithm quantises the received energy in the measurement and models the resulting counts (of measurement shots) as a multinomial distribution. The counts, defined in (4.4), are given by,

$$\mathbf{N}_t = \{\mathbf{n}_t^i\}_{i=1}^I, \tag{5.1}$$

where $\mathbf{n}_t^i$ denotes the total number of shots in pixel i and I is the total number of observed image pixels. In the previous chapter, we also defined the quantity $||\mathbf{N}_t|| = \sum_{i=1}^I \mathbf{n}_t^i$ to be the total number of shots observed at time t. However, $||\mathbf{N}_t||$ can also be interpreted as the total number of independent trials in the multinomial context, where the result of each trial leads to a shot falling into the i^{th} pixel, for $i = 1, \ldots, I$ with probability $f^i\left(\boldsymbol{x}_t^{1:M}; \pi_t^{0:M}\right)$. See subsection 4.2 for further details.

In the TkBD context, the total number of shots $||\mathbf{N}_t||$ is observable, and therefore known. This

leads to the multinomial assumption that is the basis for the standard H-PMHT algorithm. However, if we choose to assume that sample size $||\mathbf{N}_t||$ is unknown and random, the counts $\mathbf{n}_t^i$ can be modelled using a Poisson distribution. This is important in the TkBD context as the Poisson distribution has the important properties of superposition and thinning [52].

Define $Poiss(\cdot\ ;\ \lambda)$ to be the Poisson distribution with rate parameter λ, then superposition means that for $\mathfrak{Z} = \mathfrak{Z}^1 + \mathfrak{Z}^2$ with $\mathfrak{Z}^j \sim Poiss(\mathfrak{Z}^j; \lambda^j)$, the combined energy is also Poisson distributed with $\mathfrak{Z} \sim Poiss(\mathfrak{Z}; \lambda^1 + \lambda^2)$. Thinning is essentially the reverse of superposition: splitting one Poisson variable into two components leads to two Poisson variables. Surprisingly, these two variables are independent [80]. Superposition and thinning, also otherwise known as the colouring theorem, can generally be applied to an arbitrary number of variables [80].

Under the thinning property of the Poisson model, the H-PMHT counts $\mathbf{n}_t^i$ are assumed to be mutually independent Poisson random variables with rate parameters $\lambda_t f^i$. Alternatively, if we define $||\mathbf{N}_t^m||$ to be the total number of measurement shots received from each component $m = 0, \ldots, M$ at time t, then by the thinning property, the counts $||\mathbf{N}_t^m||$ can also be assumed to to be mutually independent Poisson random variables across the pixel space with rate parameters λ_t^m.

It is also possible to show how a multinomial distribution can be derived from a Poisson distribution with counts $||\mathbf{N}_t^m||$ and rate parameters λ_t^m [53]. Consider the conditional distribution of random variables $\mathbf{N}_t$ given the sample size $||\mathbf{N}_t||$. This distribution is multinomial with categories m and parameters $\Pi_t = \{\pi_t^0, \ldots, \pi_t^M\}$ such that,

$$\pi_t^m = \frac{\lambda_t^m}{\sum_{s=0}^{M} \lambda_t^s}. \tag{5.2}$$

That is, the multinomial probabilities π_t^m are simply the vector of Poisson rate parameters λ_t^m normalised to sum to unity. We can see that the Poisson and multinomial distributions are closely connected such that if we impose a fixed number of successes in several Poisson populations, the result is a multinomial distribution. This is important as it implies that the unconditional distribution of the counts $\mathbf{N}_t$ in the H-PMHT measurement image can be factored into two distributions:

- **Multinomial Distribution** for the counts $\mathbf{N}_t = \{\mathbf{n}_t^i\}_{i=1}^I$ across pixel categories i given that the sample size $||\mathbf{N}_t||$ is known,

$$\mathbf{N}_t \sim Multin\Big(||\mathbf{N}_t||, \Pi_t\Big), \tag{5.3}$$

where $Multin(\cdot)$ is the multinomial distribution and $\Pi_t = \{\pi_t^0, \ldots, \pi_t^M\}$ is the multinomial mixing proportions.

- **Poisson Distribution** for the overall total number of counts,

$$||\mathbf{N}_t|| \sim Poiss\Big(||\mathbf{N}_t||\ ;\ \lambda_t = \lambda_t^0 + \ldots + \lambda_t^M\Big), \tag{5.4}$$

where the total number of measurement shots received from each component m is Poisson distributed by the thinning property, i.e. $||\mathbf{N}_t^m|| \sim Poiss(\lambda_t^m)$.

We can show that the H-PMHT derived under a Poisson measurement model is consistent with the multinomial measurement model in the standard H-PMHT.

5.3 Derivation

In this section, we outline in detail the alternative derivation of the H-PMHT algorithm under a Poisson measurement model. The Poisson assumption, combined with the introduction of an alternative state prior density in Subsection 5.3.1, comprises the key contributions for this chapter [135, 136].

Again assume a scenario in which a sensor observing M targets collects images at discrete time intervals $t = 1, \ldots, T$, where the energy in the observed pixels $\mathbf{Z}_t$ and the energy in unobserved pixels $\mathbf{Z}_t^c$, are given by equations (4.1) and (4.2), respectively. We also again assume that the total observed energy received across the image $||\mathbf{Z}_t||$ given by (4.3) is the $L1$-norm of the sensor image.

Again, let $\boldsymbol{x}_t^m$ denote the state of component m at time t for $m = 0 \ldots M$. A component can be attributed to either a clutter or target object, therefore component $m = 0$ denotes the clutter contribution, which is assumed to be an empty set for all time scans, i.e. $\boldsymbol{x}_t^0 = \emptyset$. Assume that the remaining components $m = 1, \ldots, M$ are target objects that evolve according to a known process that may be non-linear and stochastic, and let $\mathbf{X} = \boldsymbol{x}_{1:T}^{0:M}$ denote the collection of all component states at all time scans.

The original H-PMHT derivation quantises the sensor image and assumes the resulting integer-valued image is multinomial distributed. For the derivation under a Poisson measurement model, we again make these assumptions such that the quantised energy in the observed pixels $\mathbf{N}_t$ and quantised energy in the unobserved pixels $\mathbf{N}_t^c$, are given by (4.4) and (4.11), respectively.

The standard H-PMHT also makes the implicit assumption that the number of shots $||\mathbf{N}_t||$ or multinomial trials, given by (4.6), is known. As shown in the previous subsection, this is

consistent with assuming that the number of discrete measurements from each component m is Poisson distributed. We can then re-derive the H-PMHT algorithm using an alternative measurement model by assuming that $||\mathbf{N}_t^m||$, the total number shots from component m, is Poisson distributed with rate intensity parameter λ_t^m. Note that when the measurement data is quantised, the Poisson rates will be quantised according to some arbitrary quantisation level. Define $\widetilde{\lambda}_t^m$ to be the quantised Poisson intensity rate parameter for component m such that,

$$\widetilde{\lambda}_t^m = \frac{\lambda_t^m}{c^2}, \tag{5.5}$$

where recall that $c^2 > 0$ is some arbitrary quantisation level. Also define $\mathbf{\Lambda} = \widetilde{\lambda}_{1:T}^{0:M}$ to be the collection of all $M+1$ component mixing rates across all time scans.

Given the quantised measurement data, we now assume that $||\mathbf{N}_t^m||$ follows a Poisson distribution with measurement rate $\widetilde{\lambda}_t^m$ such that,

$$Poiss(||\mathbf{N}_t^m||; \widetilde{\lambda}_t^m) = \exp(-\widetilde{\lambda}_t^m) \frac{\left[\widetilde{\lambda}_t^m\right]^{||\mathbf{N}_t^m||}}{(\ ||\mathbf{N}_t^m||\ )!}. \tag{5.6}$$

By employing the thinning properties of Poisson distributions, the counts from target m in each pixel can be assumed to be mutually independent Poisson random variables with individual rate parameters $\widetilde{\lambda}_t^{im}$, where the total intensity across the image at time t is given by $\widetilde{\lambda}_t^m = \sum_{i=1}^{I} \widetilde{\lambda}_t^{im}$. Thus, denoting the counts in pixel i due to component m as $\boldsymbol{n}_t^{im}$, thinning also results in

$$p(\boldsymbol{n}_t^{im} | \boldsymbol{x}_t^m, \widetilde{\lambda}_t^m) = \exp\left\{-\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right\} \frac{\left[\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right]^{\boldsymbol{n}_t^{im}}}{(\boldsymbol{n}_t^{im})!}, \tag{5.7}$$

where $\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)$ can be interpreted as the rate of measurements from component m in pixel i at time t.

The quantised sensor image again provides a count of the number of shots in each pixel, where each shot is assumed to be an independent identically distributed (IID) random variable. The shots have a distribution defined by the *intensity* function, $\mathbb{f}_t(\tau | \boldsymbol{x}_t^{1:M}; \widetilde{\lambda}_t^{0:M})$. In a similar way to the original H-PMHT mixture density (4.8), an expression for the underlying intensity in terms of a mixture model can be formed:

$$\mathbb{f}_t(\tau | \boldsymbol{x}_t^{1:M}; \widetilde{\lambda}_t^{0:M}) = \widetilde{\lambda}_t^0 G_0(\tau) + \sum_{m=1}^{M} \widetilde{\lambda}_t^m h\left(\tau | \boldsymbol{x}_t^m\right), \tag{5.8}$$

where $G_0(\tau)$ denotes the clutter contribution and $h\left(\tau | \boldsymbol{x}_t^m\right)$ is the psf for the target components. The intensity of shots falling into pixel i, for $i = 1, \ldots, S$ is now given by,

$$\mathbb{f}_t^i\left(\boldsymbol{x}_t^{1:M}; \widetilde{\lambda}_t^{0:M}\right) = \widetilde{\lambda}_t^0 h^i(\emptyset) + \sum_{m=1}^{M} \widetilde{\lambda}_t^m h^i(\boldsymbol{x}_t^m), \tag{5.9}$$

where $h^i(\boldsymbol{x}_t^m)$ and $h^i(\emptyset)$ are the target and clutter per-pixel shot probabilities defined in (4.9) and (4.10), respectively.

We can see that the multinomial mixing proportion π_t^m in the original H-PMHT mixture density has been replaced with the quantised Poisson mixing rate $\widetilde{\lambda}_t^m$. Recall that the multinomial mixing proportions π_t^m in the standard H-PMHT were interpreted as the proportion of power from component m for the given measurement image size. In contrast, the new Poisson mixing rates λ_t^m can be interpreted as the *absolute average power* received from component m, regardless of the image size. The multinomial mixing proportions π_t^m were also required to form a probability vector. This is not necessary for $\widetilde{\lambda}_t^m$ due to the independence assumptions that are the result of the Poisson thinning property. This is critical as it implies that the algorithm complexity will remain linear with the number of targets. We show later in subsection 5.3.5, that the new Poisson mixing rates $\widetilde{\lambda}_t^m$ can also be estimated dynamically.

Under the Poisson measurement model, the procedure for evaluating the probability of missing data under the EM is the same as in the original H-PMHT, except that the density (4.7) and per pixel probability (4.8) in the standard H-PMHT are now replaced with the intensity function (5.8) and per-pixel shot intensity (5.9). Also, as the component mixing terms are now assumed to follow a Poisson distribution, the mixing terms $\boldsymbol{\Lambda}$ are now unknown random variables that need to bee estimated. Thus under a Poisson measurement model, the variables to be estimated are the component states $\mathbf{X} = \boldsymbol{x}_{1:T}^{0:M}$ and their associated Poisson mixing rates $\boldsymbol{\Lambda} = \widetilde{\lambda}_{1:T}^{0:M}$, for all components m and time scans t. The unobserved measurement shots $\mathbf{N}^c$, assignments $\mathbf{K}$ of shots to components, and the precise location $\mathbf{L}$ of each shot in each pixel are still considered to be missing data.

5.3.1 Prior Density

As the H-PMHT derived under a Poisson measurement model still performs a quantisation over the measurement image, a modified Bayesian state prior is again required to ensure that the overabundance of synthetic data generated after quantisation does not overwhelm the state prior. In the previous chapter, the state prior was modified to include a $||\mathbf{N}_t^{total}||$ power term, effectively allowing an instance of the prior density to be applied to each individual shot at every time scan. As the $||\mathbf{N}_t^{total}||$ term is obtained from the data, this resulted in a data dependent prior.

Under Gaussian assumptions, this modification also results in a synthetic process noise covariance (4.86) that is scaled by a data dependent term. As discussed in subsection 4.5.2 this can lead to some irregularities, namely, when the image size is increased, the process noise

decreases. This is counter-intuitive as increasing the image size should have no effect on target tracking performance if the resolution of the image remains unchanged. To address this, we propose an alternative Bayesian prior, in which we only modify the prior to factor in shots from that individual target. Under Gaussian assumptions, this new state prior density results in a scaling factor in the synthetic process noise covariance matrix that is consistent with the scaling factor derived for the synthetic measurement noise covariance matrix. This allows for SNR independent filtering performance and is one of the key contributions of this chapter.

For the new state prior, we again use a first-order Markov Model as it is sufficiently non-diffuse to compensate for the overabundance of information in the likelihood function. Assuming independence with time and targets, we propose the following alternative Markov prior:

$$p(\mathbf{X}) = \prod_{m=0}^{M} \left[p(\boldsymbol{x}_0^m) \prod_{t=1}^{T} \left\{ p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m) \right\}^{||\mathbf{N}_t^m||} \right], \tag{5.10}$$

where $||\mathbf{N}_t^m||$ is a Poisson distributed random integer with an unknown rate (that can be estimated). The new prior is now only dependent on the shots for the current target. This modification results in an instance of the prior density for every shot that originated from target m.

We also impose a dynamics model on the component mixing terms $\boldsymbol{\Lambda}$, however as the Poisson mixing rates are also quantised, no modification to the prior is required. Like the state prior, we can again assume a first-order Markov Model for the mixing term prior:

$$p(\boldsymbol{\Lambda}) = \prod_{m=0}^{M} \left[p(\widetilde{\lambda}_0^m) \prod_{t=1}^{T} p(\widetilde{\lambda}_t^m | \widetilde{\lambda}_{t-1}^m) \right]. \tag{5.11}$$

5.3.2 Expectation-Maximisation

The observer for the H-PMHT algorithm with a Poisson measurement model is given by $\mathbf{O} : \{\mathbf{X}, \mathbf{K}, \boldsymbol{\Lambda}, \mathbf{N}\}$, consisting of component states $\mathbf{X} = \boldsymbol{x}_{1:T}^{0:M}$, Poisson mixing rates $\boldsymbol{\Lambda} = \widetilde{\lambda}_{1:T}^{0:M}$, and set of assignments $\mathbf{K} = \{K_t^{i,r}\}$ of components to measurement shots $\mathbf{N}$, for $i = 1, \ldots, I, r = 1, \ldots, \mathbf{n}_t^i$ and $t = 1, \ldots, T$.

The observer $\mathbf{O} : \{\mathbf{X}, \mathbf{K}, \boldsymbol{\Lambda}, \mathbf{N}\}$ is unknown and the calculation of the ML estimate for the states $\mathbf{X}$ and prior $\boldsymbol{\Lambda}$ remains infeasible due to the exponential complexity of enumerating over all assignments $\mathbf{K}$ of shots to components. Hence, we still require the EM algorithm to calculate the ML solution for $\mathbf{X}$ and $\boldsymbol{\Lambda}$, under the missing data $\{\mathbf{N}^c, \mathbf{L}, \mathbf{K}\}$.

5.3.3 E-Step

At every iteration of the EM algorithm, the goal is to find estimates for the component states $\mathbf{X}$ and Poisson mixing rates $\mathbf{\Lambda}$ conditioned on the current observed measurements $\mathbf{N}$ and the previous iteration's estimates $\mathbf{X}'$ and $\mathbf{\Lambda}'$. The E-step of the EM algorithm evaluates the conditional expectation of the logarithm of the complete data likelihood with respect to the missing data $\mathbf{L}, \mathbf{K}$ and $\mathbf{N}^c$. This is given by the auxiliary function $Q^{(P)}$ at the current iteration:

$$\begin{aligned} Q^{(P)}(\mathbf{X}, \mathbf{\Lambda} | \mathbf{X}', \mathbf{\Lambda}') &= E_{\mathbf{N}^c \mathbf{L}\, \mathbf{K}} \left[\log p(\mathbf{X}, \mathbf{\Lambda}, \mathbf{N}, \mathbf{N}^c, \mathbf{L}, \mathbf{K}) | \mathbf{X}', \mathbf{\Lambda}', \mathbf{N} \right] \\ &= \sum_{\mathbf{N}^C \mathbf{K}} \int_{\mathbf{L}} \log p(\mathbf{X}, \mathbf{\Lambda}, \mathbf{N}, \mathbf{N}^c, \mathbf{L}, \mathbf{K}) p(\mathbf{N}^c, \mathbf{L}, \mathbf{K} | \hat{\mathbf{X}}', \mathbf{\Lambda}', \mathbf{N}) d\mathbf{L}. \quad (5.12) \end{aligned}$$

where $E_{\mathbf{N}^c, \mathbf{L}, \mathbf{K}}$ denotes the expectation with respect to the missing data. If we compare (5.12) to the original H-PMHT auxiliary function (4.14), we observe that the differences are minor; the functions are identical except that the multinomial mixing proportions Π have been replaced with the Poisson mixing rates $\mathbf{\Lambda}$, and the expectation is also conditioned on the rates $\mathbf{\Lambda}'$.

The first term in (5.12) is the complete data likelihood and the second term is the conditional density of the missing data given by,

$$p(\mathbf{N}^c, \mathbf{L}, \mathbf{K} | \mathbf{X}', \mathbf{\Lambda}', \mathbf{N}) = \frac{p(\mathbf{X}', \mathbf{\Lambda}', \mathbf{N}, \mathbf{N}^c, \mathbf{L}, \mathbf{K})}{p(\mathbf{X}', \mathbf{\Lambda}', \mathbf{N})}. \quad (5.13)$$

The numerator and the denominator in (5.13) are exactly the complete and incomplete data likelihoods, respectively. We now derive expressions for both likelihoods.

5.3.3.1 Incomplete Data Likelihood

Consider evaluating $p(\mathbf{N} | \mathbf{X}', \mathbf{\Lambda}')$, the probability of drawing $\mathbf{n}_t^i$ shots from each pixel i given that the source for each shot is unknown:

$$p(\mathbf{N} | \mathbf{X}', \mathbf{\Lambda}') = \prod_{t=1}^{T} \prod_{i=1}^{I} p(\mathbf{n}_t^i | \mathbf{X}_t'). \quad (5.14)$$

As discussed in Section 5.2, the counts $\mathbf{n}_t^i$ can be assumed to follow a Poisson distribution with mean $\mathbb{f}_t^i$. We can now write,

$$\begin{aligned} p(\mathbf{N}_t | \mathbf{X}_t', \mathbf{\Lambda}_t') &= \prod_{i=1}^{I} \exp\left\{ -\mathbb{f}_t^{i'} \right\} \frac{\left[\mathbb{f}_t^{i'} \right]^{\mathbf{n}_t^i}}{\mathbf{n}_t^i!} \\ &= \exp\left\{ -\sum_{i=1}^{I} \mathbb{f}_t^{i'} \right\} \prod_{i=1}^{I} \frac{1}{\mathbf{n}_t^i!} \left[\mathbb{f}_t^{i'} \right]^{\mathbf{n}_t^i} \\ &= \frac{\gamma_t^*}{\exp\{\mathbb{F}'_t\}} \prod_{i=1}^{I} \left[\mathbb{f}_t^{i'} \right]^{\mathbf{n}_t^i}, \quad (5.15) \end{aligned}$$

where $\gamma_t^* = \dfrac{1}{\prod_{i=1}^{I} \mathbf{n}_t^i!}$, $\mathbb{f}_t^{i\prime} = \mathbb{f}_t^i\left(\left[\boldsymbol{x}_t^{0:M}\right]'; \left[\widetilde{\lambda}_t^{0:M}\right]'\right)$ and $\mathbb{F}_t' = \sum_{i=1}^{I} \mathbb{f}_t^{i\prime}$. Note that $\mathbb{F}_t'$ is defined only across the observed measurement space and is calculated using the estimates of the state $\left[\boldsymbol{x}_t^{0:M}\right]'$ and of the Poisson mixing rates $\left[\widetilde{\lambda}_t^{0:M}\right]'$ from the previous EM iteration. The $\mathbb{F}_t'$ term can also be approximated by,

$$\begin{aligned}
\mathbb{F}_t' &= \sum_{i=1}^{I} \mathbb{f}_t^{i\prime} \\
&= \widetilde{\lambda}_t^0 \sum_{i=1}^{I} h^i(\emptyset) + \sum_{m=1}^{M} \widetilde{\lambda}_t^m \sum_{i=1}^{I} h^i(\boldsymbol{x}_t^m), \\
&\approx \widetilde{\lambda}_t^0 + \sum_{m=1}^{M} \widetilde{\lambda}_t^m, \\
&= \widetilde{\lambda}_t. \qquad (5.16)
\end{aligned}$$

In (5.16), we have assumed that the unobserved measurement space is small or empty. As a result, we can approximate the sum over the entire observation space with only the sum over the observed pixels such that, $\sum_{i=1}^{S} h(\cdot) \approx \sum_{i=1}^{I} h(\cdot) = 1$ as the sum is equivalent to the integral over the whole psf. We can see that $\mathbb{F}_t'$ can be interpreted to be the total power in the image.

Substituting in (5.15), the incomplete data likelihood is then given as,

$$\begin{aligned}
p(\mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N}) &= p(\mathbf{N}|\mathbf{X}', \boldsymbol{\Lambda}')p(\mathbf{X}')p(\boldsymbol{\Lambda}') \\
&= p(\mathbf{X}')p(\boldsymbol{\Lambda}') \prod_{t=1}^{T} p(\mathbf{N}_t|\mathbf{X}_t', \boldsymbol{\Lambda}_t') \\
&\propto p(\mathbf{X}')p(\boldsymbol{\Lambda}') \prod_{t=1}^{T} \frac{1}{\exp\{\widetilde{\lambda}_t\}} \prod_{i=1}^{I} \left[\mathbb{f}_t^{i\prime}\right]^{\mathbf{n}_t^i}, \qquad (5.17)
\end{aligned}$$

where $\mathbf{X}'$ and $\boldsymbol{\Lambda}'$ are assumed to be independent of each other. The γ_t^* term is not dependent on $\mathbf{X}$ or $\boldsymbol{\Lambda}$, and in the context of the optimisation problem it can be treated as a scaling constant and ignored. We can see that (5.17) is similar in form to the complete data likelihood (4.18) derived under the multinomial measurement model, however the densities have been replaced with intensity functions and the denominator in (5.17) now features an exponent term.

5.3.3.2 Complete Data Likelihood

Under a Poisson measurement model, it is possible to derive the complete data likelihood in a different way from the steps outlined in subsection 4.2.3.2 for the standard H-PMHT. The assignment indices $\mathbf{K}$ can be conceptually divided into two pieces of information: the number of shots due to each component and the particular permutation of individual shot

sources. In the Poisson context, it is useful to explicitly recognise this by adding a redundant variable $\mathbf{N}^k = \{\mathbf{n}_t^{im}\}$ that specifies the number of shots in each pixel from each component, for $m = 1, \ldots, M$, $i = 1, \ldots, I$ and $t = 1, \ldots T$. It is important to note that the variable $\mathbf{N}^k$ is implied by $\mathbf{K}$, and that $\mathbf{N}$ is completely determined by $\mathbf{N}^k$. By the thinning property, $\mathbf{N}^k$ also follows a Poisson distribution. Removing the unobserved data term $\mathbf{N}^c$ as it does not contribute to the conditional probability of $\{\mathbf{L}, \mathbf{K}\}$, we have the following expression,

$$\begin{aligned} p(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{N}, \mathbf{N}^k, \mathbf{L}, \mathbf{K}) &= p(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{N}^k, \mathbf{L}, \mathbf{K}) \\ &= p(\mathbf{L}|\mathbf{X}, \mathbf{N}^k, \mathbf{K}) p(\mathbf{K}, \mathbf{N}^k|\mathbf{X}, \boldsymbol{\Lambda}) p(\mathbf{X}) p(\boldsymbol{\Lambda}), \end{aligned} \tag{5.18}$$

as the precise locations $\mathbf{L}$ of each shot in a pixel is independent of the Poisson mixing rates $\boldsymbol{\Lambda}$. We can see that (5.18) is similar in form to the complete data likelihood (4.21) formed under multinomial assumptions. The first term in (5.18) is the probability of drawing the locations $\mathbf{L}$ conditioned on the target states, measurement shots and assignments. This probability is again given by (4.23) and remains unchanged under the Poisson measurement model as it is a function of the psf $h(\cdot)$ only, which is independent of mixing terms. The second term in (5.18), $p(\mathbf{K}, \mathbf{N}^k|\mathbf{X}, \boldsymbol{\Lambda})$, extracts all the shots that originate from a particular component and can be simplified as follows,

$$p(\mathbf{K}, \mathbf{N}^k|\mathbf{X}, \boldsymbol{\Lambda}) = p(\mathbf{K}|\mathbf{N}^k, \mathbf{X}, \boldsymbol{\Lambda}) p(\mathbf{N}^k|\mathbf{X}, \boldsymbol{\Lambda}). \tag{5.19}$$

The first term on the rhs of (5.19) is the probability of the particular permutation of assignments conditioned on knowing the total number of shots that originated from each source. We can evaluate this probability by first considering the number of ways that $\mathbf{n}_t^{im}$ shots from components $m = 0, \ldots, M$ can be selected from $\mathbf{n}_t^i$ total shots in pixel i. Clearly, this is given by the multinomial coefficient,

$$\binom{\mathbf{n}_t^i}{\mathbf{n}_t^{i0}, \ldots, \mathbf{n}_t^{iM}} = \frac{\mathbf{n}_t^i!}{\prod_{m=0}^{M} \mathbf{n}_t^{im}!}. \tag{5.20}$$

Define K_t^i to be the set of components associated with the $\boldsymbol{n}_t^i$ shots in pixel i. The probability of observing a certain set of components m can be expressed as,

$$\begin{aligned} p(\mathbf{K}|\mathbf{N}^k, \mathbf{X}, \boldsymbol{\Lambda}) &= \prod_{t=1}^{T} \prod_{i=1}^{I} p(K_t^i|\mathbf{n}_t^{i0}, \ldots, \mathbf{n}_t^{iM}, \mathbf{X}, \boldsymbol{\Lambda}) \\ &= \prod_{t=1}^{T} \prod_{i=1}^{I} \frac{1}{\binom{\mathbf{n}_t^i}{\mathbf{n}_t^{i0}, \ldots, \mathbf{n}_t^{iM}}} \\ &= \prod_{t=1}^{T} \prod_{i=1}^{I} \frac{\prod_{m=0}^{M} \mathbf{n}_t^{im}!}{\mathbf{n}_t^i!}. \end{aligned} \tag{5.21}$$

Note that in (5.21), the dependence of $\mathbf{K}$ on component states $\mathbf{X}$ is redundant. Also note that the number of shots from each component depends on the mixing rates, therefore when we condition on $\mathbf{N}^k$, it is redundant to also condition on $\mathbf{\Lambda}$.

The second term in (5.19), $p(\mathbf{N}^k|\mathbf{X},\mathbf{\Lambda})$, is the probability of drawing $\boldsymbol{n}_t^{im}$ shots from components m in pixel i. In this chapter, we assume that $\boldsymbol{n}_t^{im}$ follows a Poisson distribution with rate parameter $\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)$ such that,

$$\begin{aligned}
p(\mathbf{N}^k|\mathbf{X},\mathbf{\Lambda}) &= \prod_{t=1}^{T}\prod_{i=1}^{I}\prod_{m=0}^{M} p(\boldsymbol{n}_t^{im}|\mathbf{X}_t,\mathbf{\Lambda}_t') \\
&= \prod_{t=1}^{T}\prod_{i=1}^{I}\prod_{m=0}^{M} \exp\Big\{-\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\Big\}\frac{\left[\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right]^{\mathbf{n}_t^{im}}}{\mathbf{n}_t^{im}!} \\
&= \prod_{t=1}^{T}\exp\Big\{-\sum_{i=1}^{I}\sum_{m=0}^{M}\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\Big\}\prod_{m=0}^{M}\prod_{i=1}^{I}\frac{\left[\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right]^{\mathbf{n}_t^{im}}}{\mathbf{n}_t^{im}!} \\
&= \prod_{t=1}^{T}\exp\Big\{-\mathbb{F}_t\Big\}\prod_{m=0}^{M}\prod_{i=1}^{I}\frac{\left[\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right]^{\mathbf{n}_t^{im}}}{\mathbf{n}_t^{im}!}, \qquad (5.22)
\end{aligned}$$

where $\mathbb{F}_t = \sum_{i=1}^{I}\mathbb{f}_t^i$ is the integral of the intensity function $\mathbb{f}_t$ over the observed measurement space. Substituting (5.21) and (5.22) into (5.19) results in

$$\begin{aligned}
p(\mathbf{K},\mathbf{N}^k|\mathbf{X},\mathbf{\Lambda}) &= \prod_{t=1}^{T}\exp\Big\{-\mathbb{F}_t\Big\}\prod_{i=1}^{I}\left[\prod_{m=0}^{M}\left(\frac{\left\{\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right\}^{\mathbf{n}_t^{im}}}{\mathbf{n}_t^{im}!}\right)\times\frac{\prod_{m=0}^{M}\mathbf{n}_t^{im}!}{\mathbf{n}_t^{i}!}\right] \\
&= \prod_{t=1}^{T}\exp\Big\{-\mathbb{F}_t\Big\}\left[\prod_{i=1}^{I}\prod_{m=0}^{M}\left\{\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right\}^{\mathbf{n}_t^{im}}\right]\left[\prod_{i=1}^{I}\frac{1}{\prod_{m=0}^{M}\mathbf{n}_t^{im}!}\times\frac{\prod_{m=0}^{M}\mathbf{n}_t^{im}!}{\mathbf{n}_t^{i}!}\right] \\
&= \prod_{t=1}^{T}\frac{1}{\exp\Big\{\mathbb{F}_t\Big\}}\prod_{i=1}^{I}\vartheta_t^i\prod_{m=0}^{M}\left[\widetilde{\lambda}_t^m h^i\left(\boldsymbol{x}_t^m\right)\right]^{\mathbf{n}_t^{im}} \\
&= \prod_{t=1}^{T}\frac{1}{\exp\Big\{\mathbb{F}_t\Big\}}\prod_{i=1}^{I}\vartheta_t^i\prod_{r=1}^{\mathbf{n}_t^{i}}\widetilde{\lambda}_t^{K_t^{ir}} h^i\left(\boldsymbol{x}_t^{K_t^{ir}}\right), \qquad (5.23)
\end{aligned}$$

where $\vartheta_t^i = \dfrac{1}{\mathbf{n}_t^i!}$. In the last line of (5.23), the product over the components m has been changed to an equivalent product over the shots r, which results in a unique term for each shot in the pixel. This eliminates the power term as the inner product is no longer grouped according to its source. Finally, by substituting (5.23) and the probability of drawing the locations $\mathbf{L}$ (4.23)

into (5.18), an expression for the complete data likelihood can be found,

$$\begin{aligned} p(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{N}, \mathbf{L}, \mathbf{K}) &= p(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{N}^k, \mathbf{L}, \mathbf{K}) \\ &= p(\mathbf{X})\, p(\boldsymbol{\Lambda}) \prod_{t=1}^{T} \frac{1}{\exp\left\{\mathbb{F}_t\right\}} \prod_{i=1}^{I} \vartheta_t^i \prod_{r=1}^{\mathbf{n}_t^i} \widetilde{\lambda}_t^{K_t^{ir}} h^i\left(\boldsymbol{x}_t^{K_t^{ir}}\right) \times \frac{h\left(\mathbf{L}_t^{ir} | \boldsymbol{x}_t^{K_t^{ir}}\right)}{h^i\left(\boldsymbol{x}_t^{K_t^{ir}}\right)} \\ &= p(\mathbf{X})\, p(\boldsymbol{\Lambda}) \prod_{t=1}^{T} \frac{1}{\exp\left\{\mathbb{F}_t\right\}} \prod_{i=1}^{I} \vartheta_t^i \prod_{r=1}^{\mathbf{n}_t^i} \widetilde{\lambda}_t^{K_t^{ir}} h\left(\mathbf{L}_t^{ir} | \boldsymbol{x}_t^{K_t^{ir}}\right). \end{aligned} \tag{5.24}$$

Comparing the above complete data likelihood with its counterpart (4.26) in the original H-PMHT derivation, we clearly see some similarities between the two. Both likelihoods feature a mixing term scaled by the psf centred at $\mathbf{L}_t^{ir}$, and a term that approximates the integral to their respective mixture models.

Although it is possible to express the complete data likelihood in terms of $\mathbf{N}^k$ under both Poisson and multinomial measurement models, explicit expressions for $p(\mathbf{N}^k|\mathbf{X}, \cdot)$ can only be formulated in the Poisson case. This is because the multinomial model implicitly assumes that the quantised counts are conditioned on $||\mathbf{N}_t||$. The standard H-PMHT assumes $||\mathbf{N}_t||$ is a known constant and no guidance is given on how to evaluate $p(\mathbf{N}^k|\mathbf{X}, \cdot)$.

5.3.3.3 Auxiliary Function

An expression for the auxiliary function $Q^{(P)}$ can be found by substituting the prior (5.10) and complete data likelihood (5.24) into (5.12) to give,

$$\begin{aligned} Q^{(P)}(\mathbf{X}, \boldsymbol{\Lambda} | \mathbf{X}', \boldsymbol{\Lambda}') = \sum_{\mathbf{N}^C, \mathbf{K}} \int_{\mathbf{L}} \Bigg[& \sum_{m=0}^{M} \left(\log\left\{p(\boldsymbol{x}_0^m)\right\} + \sum_{t=1}^{T} ||\mathbf{N}_t^m||\ \log\left\{p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m)\right\} \right. \\ & \left. + \log\left\{p(\widetilde{\lambda}_0^m)\right\} + \sum_{t=1}^{T} \log\left\{p(\widetilde{\lambda}_t^m | \widetilde{\lambda}_{t-1}^m)\right\} \right) + \sum_{t=1}^{T} \sum_{i=1}^{I} \log\{\vartheta_t^i\} - \sum_{t=1}^{T} \mathbb{F}_t \\ & + \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{\mathbf{n}_t^i} \log\left\{\widetilde{\lambda}_t^{K_t^{ir}}\right\} + \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{r=1}^{n_t^i} \log\left\{h(\mathbf{L}_t^{ir} | \boldsymbol{x}_t^{K_t^{ir}})\right\} \Bigg] \times p(\mathbf{N}^c, \mathbf{L}, \mathbf{K} | \mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N})\, d\mathbf{L}. \end{aligned} \tag{5.25}$$

The last term on the rhs of (5.25) is the conditional density of the missing data and can be simplified as follows,

$$p(\mathbf{N}^c, \mathbf{L}, \mathbf{K} | \mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N}) = p(\mathbf{N}^c | \mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N}) p(\mathbf{L} | \mathbf{X}', \mathbf{N}, \mathbf{K}) p(\mathbf{K} | \mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N}). \tag{5.26}$$

Again, we assume that the contribution from the unobserved pixels is small such that the sum of the psf over the observed pixels $\sum_{i=1}^{I} h^i(\boldsymbol{x}_t^m) \approx 1$ as the sum is equivalent to the integral over

the whole psf. By (5.16), we can make the following simplification:

$$\mathbb{F}_t = \widetilde{\lambda}_t. \tag{5.27}$$

Substituting (5.26) and (5.27) into (5.25) and re-inserting the $\mathbf{N}^c$ terms, we can now apply the expectation across the missing data to the relevant terms to result in

$$\begin{aligned} Q^{(P)}(\mathbf{X}, \boldsymbol{\Lambda}|\mathbf{X}', \boldsymbol{\Lambda}') = \sum_{m=0}^{M}\Bigg[\log\left\{p(\boldsymbol{x}_0^m)\right\} + \sum_{t=1}^{T}\sum_{\mathbf{N}^c,\mathbf{K}} ||\mathbf{N}_t^m|| \log\left\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\right\} p(\mathbf{N}^c, \mathbf{K}|\mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N})\Bigg] \\ + \sum_{m=0}^{M}\left[\log\left\{p(\widetilde{\lambda}_0^m)\right\} + \sum_{t=1}^{T}\log\left\{p(\widetilde{\lambda}_t^m|\widetilde{\lambda}_{t-1}^m)\right\}\right] - \sum_{t=1}^{T}\widetilde{\lambda}_t \\ + \sum_{\mathbf{N}^c,\mathbf{K}}\left[\sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{r=1}^{\mathbf{n}_t^i}\log\left\{\widetilde{\lambda}_t^{K_t^{ir}}\right\}\right] p(\mathbf{K}, \mathbf{N}^c|\mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N}) \\ + \sum_{\mathbf{N}^c,\mathbf{K}}\int_{\mathbf{L}}\left[\sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{r=1}^{\mathbf{n}_t^i}\log\left\{h(\mathbf{L}_t^{ir}|\boldsymbol{x}_t^{K_t^{ir}})\right\}\right] p(\mathbf{N}^c, \mathbf{L}, \mathbf{K}|\mathbf{X}', \boldsymbol{\Lambda}', \mathbf{N}) d\mathbf{L}. \end{aligned} \tag{5.28}$$

We can ignore all terms that are not dependent on $\mathbf{X}$ or $\boldsymbol{\Lambda}$, therefore ϑ_t^i can be removed as it is simply a scaling constant. Observe that the second line in (5.28) remains unchanged when the expectation is taken over the missing data as it only depends on the Poisson mixing rates $\widetilde{\lambda}_t$, which are independent of the missing data.

The new auxiliary function (5.28) looks very similar to the original H-PMHT auxiliary function (4.30) derived under the multinomial model. However there are a few obvious differences; the multinomial mixing proportions π have been replaced with the Poisson mixing rates $\widetilde{\lambda}$, and a dynamics model has now been applied to $\widetilde{\lambda}$. In the first line of (5.28), the component states and Poisson mixing rates are independent of the locations $\mathbf{L}$. However, due to the new component state prior, the expectation is now taken over both the unobserved counts $\mathbf{N}^c$ and associations $\mathbf{K}$.

We now evaluate each line in (5.28) separately. In the first line of (5.28), the term $\sum_{\mathbf{N}^c,\mathbf{K}} ||\mathbf{N}_t^m|| p(\mathbf{N}^c, \mathbf{K}|\mathbf{X}_t', \boldsymbol{\Lambda}_t', \mathbf{N}_t)$ can be interpreted as the expected number of shots due to component m. As the shots are independent across pixels, we can separate the term into the expected number of shots from the observed pixels and the expected number of shots from the unobserved pixels. Define $\mathbf{n}_t^{ci}$ as the measurement count from unobserved pixels $i = I+1, \ldots, S$.

We can then write the following expression,

$$\sum_{\mathbf{N}_t^c,\mathbf{K}_t} ||\mathbf{N}_t^m||p(\mathbf{N}_t^c,\mathbf{K}_t|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t) = \sum_{i=1}^{I}\sum_{|K_t^i|}\mathbf{n}_t^{im}|K_t^i|\ p(|K_t^i|\Big|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t)$$
$$+\sum_{i=I+1}^{S}\sum_{\mathbf{n}_t^{ci}=1}^{\infty}\sum_{|K_t^i|}\mathbf{n}_t^{im}|K_t^i|\ p(|K_t^i|\Big|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t)p(\mathbf{n}_t^{ci}|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t). \tag{5.29}$$

where $|K_t^i|$ is defined as the cardinality of the set of components associated with the $\boldsymbol{n}_t^i$ shots in pixel i. Note that the expectation over the unobserved shots $\mathbf{N}_t^c$ is only applicable to the unobserved measurement space. The first term on the rhs of (5.29) is the mean number of shots due to component m in the observed pixel i given $\mathbf{n}_t^i$, and the previous EM estimates $\mathbf{X}'$ and $\mathbf{\Lambda}'$. As the total number of shots has been defined and following the discussion in Section 5.2, this mean count follows a multinomial distribution where the probability of a shot falling into an observed pixel is given by,

$$\widetilde{\mu}_t^{im} = \frac{\widetilde{\lambda}_t^{m'}h^i\left(\boldsymbol{x}_t^{m'}\right)}{\mathbb{r}_t^{i'}}. \tag{5.30}$$

This contribution from the observed pixels is then given by,

$$\sum_{K_t^i}\mathbf{n}_t^{im}K_t^i\ p(K_t^i|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t) \quad = \quad \widetilde{\mu}_t^{im}\mathbf{n}_t^i. \tag{5.31}$$

The μ_t^{im} term can be thought of as an association weight and represents the relative proportion of power from component m in pixel i. The second term on the rhs of (5.29) represents the contribution from the unobserved pixels. Substituting (5.31) into the second line of (5.29) results in

$$\begin{aligned}
&\sum_{\mathbf{n}_t^{ci}=1}^{\infty}\sum_{K_t^i}\mathbf{n}_t^{im}K_t^i\ p(K_t^i|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t)p(\mathbf{n}_t^{ci}|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t)\\
&= \quad \widetilde{\mu}_t^{im}\sum_{\mathbf{n}_t^{ci}=1}^{\infty}\mathbf{n}_t^{ci}p(\mathbf{n}_t^{ci}|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t)\\
&= \quad \widetilde{\mu}_t^{im}\ \mathbb{r}_t^{i'}.
\end{aligned} \tag{5.32}$$

Above, the summand on the second line of (5.32) is simply the mean of $\mathbf{n}_t^{ci}$, which we have assumed to be Poisson distributed with mean $\mathbb{r}_t^{i'}$. Unlike the standard H-PMHT, we have assumed that the counts in the observed and unobserved measurement space follow the same distribution. The first line of the auxiliary function can be expressed as,

$$\sum_{\mathbf{N}^c,\mathbf{K}}||\mathbf{N}_t^m||p(\mathbf{N}^c,\mathbf{K}_t|\mathbf{X}_t',\mathbf{\Lambda}_t',\mathbf{N}_t) = \sum_{i=1}^{S}\widetilde{\mu}_t^{im}\widetilde{n}_t^i, \tag{5.33}$$

where

$$\widetilde{n}_t^i = \begin{cases} \mathbf{n}_t^i, & i \in \mathcal{O}, \\ \mathbb{I}_t^{i'}, & i \in \bar{\mathcal{O}}. \end{cases} \tag{5.34}$$

The evaluation of the final lines in the auxiliary function (5.28) follows the same procedure as in the standard H-PMHT. Note that the auxiliary function (5.28) is conditioned on the term $\mathbf{\Lambda}'$, however as mentioned in subsection 5.3.3.2, conditioning on the measurements $\mathbf{N}$ makes conditioning on $\mathbf{\Lambda}'$ redundant. Therefore, expressions for the third and fourth lines of the auxiliary function can be found by taking the equivalent terms under the standard H-PMHT, given by (4.44) and (4.45), respectively, and replacing the original mixing proportions π_t^m with the new Poisson mixing rates $\widetilde{\lambda}_t^m$, and the densities f_t^i with the intensities $\mathbb{I}_t^{i'}$. This results in the following expression for the auxiliary function,

$$\begin{aligned} Q^{(P)}(\mathbf{X}, \mathbf{\Lambda} | \mathbf{X}', \mathbf{\Lambda}') = \sum_{m=0}^{M} \Bigg[& \log\Big\{ p(\boldsymbol{x}_0^m) \Big\} + \sum_{t=1}^{T} \sum_{i=1}^{S} \widetilde{\mu}_t^{im} \widetilde{n}_t^i \log\Big\{ p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m) \Big\} \\ & + \log\Big\{ p(\widetilde{\lambda}_0^m) \Big\} + \sum_{t=1}^{T} \log\Big\{ p(\widetilde{\lambda}_t^m | \widetilde{\lambda}_{t-1}^m) \Big\} - \sum_{t=1}^{T} \lambda_t \\ & + \sum_{t=1}^{T} \sum_{i=1}^{I} \widetilde{\mu}_t^{im} \mathbf{n}_t^i \log\Big\{ \widetilde{\lambda}_t^m \Big\} \\ & + \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{\widetilde{\lambda}_t^{m'} \mathbf{n}_t^i}{\mathbb{I}_t^{i'}} \int_{B_i} h(\tau | \boldsymbol{x}_t^{m'}) \log\Big\{ h(\tau | \boldsymbol{x}_t^m) \Big\} d\tau \Bigg]. \end{aligned} \tag{5.35}$$

We point out that the original auxiliary function in the standard H-PMHT contains a term for the expected quantised measurement $\bar{n}_t^i$. This term, defined by (4.34) is required in the standard H-PMHT as the expectation over the unobserved counts $\mathbf{N}^c$ differs under the observed and unobserved measurement space. This term is not necessary here as under a Poisson measurement model, $\bar{n}_t^i = \mathbf{n}_t^i$ for the entire measurement space.

5.3.4 Taking the Limit of the Quantisation

In the final step of the derivation, the limit of the auxiliary function is taken, such that $Q^{(P)}$ is replaced with $\lim_{c^2 \to 0} c^2 \, Q^{(P)}$. Note that the variables $\widetilde{\lambda}_t^m$ and $\mathbf{n}_t^i$ will be affected by the limit of the quantisation. By (4.47), it is known that the $\lim_{c^2 \to 0} c^2 \, \mathbf{n}_t^i = \boldsymbol{z}_t^i$. Using the definition for $\widetilde{\lambda}_t^m$ given in (5.5), we can also write the following,

$$\begin{aligned} \lim_{c^2 \to 0} c^2 \widetilde{\lambda}_t^m &= \lim_{c^2 \to 0} c^2 \left\lfloor \frac{\lambda_t^m}{c^2} \right\rfloor \\ &= \lim_{c^2 \to 0} \left\{ c^2 \frac{\lambda_t^m}{c^2} - \epsilon \right\} \\ &= \lambda_t^m, \end{aligned} \tag{5.36}$$

where $\epsilon < c^2$. As the variable $\widetilde{n}_t^i$ defined in (5.34) also depends on $\mathbf{n}_t^i$, taking the limit of the quantisation results in

$$\lim_{c^2 \to 0} \widetilde{n}_t^i = \widetilde{z}_t^i, \tag{5.37}$$

where

$$\widetilde{z}_t^i = \begin{cases} \boldsymbol{z}_t^i & i \in \mathcal{O}, \\ \mathbb{f}_t^{i'}, & i \in \bar{\mathcal{O}}, \end{cases} \tag{5.38}$$

where the per-pixel intensity $\mathbb{f}_t^{i'}$ is now no longer a function of the quantised Poisson mixing rates $\widetilde{\lambda}_t^m$:

$$\mathbb{f}_t^{i'} \quad = \quad \lambda_t^{0'} h^i(\emptyset) + \sum_{m=1}^{M} \lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'}). \tag{5.39}$$

It is assumed that floor function is sufficiently stable such that the summation over pixels i can be taken outside the limit. When the limit of $Q^{(P)}$ is taken, we can substitute (4.47), (5.36) and (5.37) into (5.35) to give the following expression for the auxiliary function under the Poisson measurement model:

$$\begin{aligned} Q^{(P)}(\mathbf{X}, \boldsymbol{\Lambda} | \mathbf{X}', \boldsymbol{\Lambda}') = \sum_{m=0}^{M} \Bigg[& \log \left\{ p(\boldsymbol{x}_0^m) \right\} + \sum_{t=1}^{T} \sum_{i=1}^{S} \mu_t^{im} \widetilde{z}_t^i \ \log \left\{ p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m) \right\} \\ & + \log \left\{ p(\lambda_0^m) \right\} + \sum_{t=1}^{T} \ \log \left\{ p(\lambda_t^m | \lambda_{t-1}^m) \right\} - \sum_{t=1}^{T} \lambda_t \\ & + \sum_{t=1}^{T} \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \log \left\{ \lambda_t^m \right\} \\ & + \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{\lambda_t^{m'} \boldsymbol{z}_t^i}{\mathbb{f}_t^{i'}} \int_{B_i} h(\tau | \boldsymbol{x}_t^{m'}) \log \left\{ h(\tau | \boldsymbol{x}_t^m) \right\} d\tau \Bigg]. \end{aligned} \tag{5.40}$$

We can decompose (5.40) into two separate expressions for estimating the Poisson mixing rates λ_t^m and the component states $\boldsymbol{x}_t^m$ such that,

$$Q^{(H)}(\mathbf{X}, \boldsymbol{\Lambda} | \mathbf{X}', \boldsymbol{\Lambda}') = \sum_{m=0}^{M} Q_X^m + \sum_{m=0}^{M} Q_\lambda^m, \tag{5.41}$$

where

$$\begin{aligned} Q_X^m = & \log \left\{ p(\boldsymbol{x}_0^m) \right\} + \sum_{t=1}^{T} \sum_{i=1}^{S} \mu_t^{im} \widetilde{z}_t^i \log \left\{ p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m) \right\} \\ & + \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{\lambda_t^{m'} \boldsymbol{z}_t^i}{\mathbb{f}_t^{i'}} \int_{B_i} h(\tau | \boldsymbol{x}_t^{m'}) \log \left\{ h(\tau | \boldsymbol{x}_t^m) \right\} d\tau, \end{aligned} \tag{5.42}$$

$$Q_\lambda^m = \log \left\{ p(\lambda_0^m) \right\} + \ \sum_{t=1}^{T} \left[\log \{ p(\lambda_t^m | \lambda_{t-1}^m) \} + \log \left\{ \lambda_t^m \right\} \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i - \lambda_t \right]. \tag{5.43}$$

Unlike the standard H-PMHT that assumes the component mixing proportions π_t^m are uncorrelated with time, under the new derivation we have imposed a dynamics model on the Poisson mixing rates λ_t^m to generate a smoothed time estimate.

We now discuss how the introduction of the new modified prior density affects the dynamics noise covariance. For a linear Gaussian dynamics model, the data dependent prior in (5.42) evaluates to,

$$\sum_{i=1}^{S} \mu_t^{im} \widetilde{z}_t^i \log \left\{ p(\boldsymbol{x}_t | \boldsymbol{x}_{t|t-1}) \right\} = C - \frac{1}{2} (\boldsymbol{x}_t - \mathsf{F}\boldsymbol{x}_{t|t-1})^{\mathsf{T}} \left(\bar{\mathsf{Q}}_t^{im} \right)^{-1} (\boldsymbol{x}_t - \mathsf{F}\boldsymbol{x}_{t|t-1}), \tag{5.44}$$

where C is some constant and $\bar{\mathsf{Q}}_t^{im}$ is the modified synthetic dynamics covariance matrix:

$$\bar{\mathsf{Q}}_t^{im} = \frac{\mathsf{Q}}{\sum_{i=1}^{S} \mu_t^{im} \widetilde{z}_t^i}. \tag{5.45}$$

Under the new state prior density and a Gaussian target model, the noise covariance matrix is scaled according to the received measurements from each pixel and the μ_t^{im} term, which represents the relative proportion of power from component m in each pixel. The new noise covariance is now both pixel and target dependent. Under a KF implementation, the synthetic measurement noise covariance matrix can be derived in a similar way to (4.65) to give,

$$\begin{aligned} \tilde{\mathsf{R}}_t^{(m,P)} &= \left(\frac{\lambda_t^{m'} \sum_{i=1}^{I} \boldsymbol{z}_t^i h^i(\boldsymbol{x}_t^{m'})}{\mathbb{f}_t^{i'}} \right)^{-1} \mathsf{R} \\ &= \frac{\mathsf{R}}{\sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i}. \end{aligned} \tag{5.46}$$

As $\widetilde{z}_t^i = \boldsymbol{z}_t^i$ for observed pixels, we can see that the scaling factor for the synthetic process noise covariance matrix $\bar{\mathsf{Q}}_t^{im}$ is consistent with the scaling factor derived for the synthetic measurement noise covariance matrix $\mathsf{R}^{(m,P)}$. As a result, the KF performance is no longer a function of SNR. By introducing a new modified state prior density, we are able to correct one of the key limitations of the H-PMHT that resulted in inconsistent tracking performance whenever the image size was changed.

5.3.5 M-Step

In the previous subsection, we showed that by adopting a Poisson measurement model, we can estimate the received power for each component m using the Poisson mixing rates λ_t^m. The Poisson model also naturally imposes a dynamics model on the component mixing rates λ_t^m, which allows a smoothed estimate with time.

In this subsection, we describe how to calculate the new smoothed Poisson mixing rates λ_t^m based on a procedure by Granström [60]. The extension of the H-PMHT to allow for a dynamic mixing term through the assumption of a Poisson measurement model is one of the key contributions of this chapter [135, 136].

Under the Poisson measurement model, the procedure for maximising the component states $\mathbf{X}$ remains the same as in the standard H-PMHT. That is, for a Gaussian psf $h(\cdot)$, the $\log h(\cdot)$ is clearly a quadratic, and the integral evaluation becomes trivial. For non-Gaussian psfs, the estimation of the component states $\mathbf{X}$ can be performed through an appropriate non-linear filter [39]. See Subsection 4.3 for further details.

To estimate the Poisson measurement rate λ_t^m, consider a model for the prior $p(\lambda_0^m)$ and its evolution with time, $p(\lambda_t^m|\lambda_{t-1}^m)$. The well known conjugate prior for the Poisson distribution is the gamma distribution. That is, if the prior distribution for λ is a gamma distribution with parameters α, β, then the posterior distribution is also a gamma distribution. Hence, let the prior for the Poisson measurement rate be a gamma distribution with parameters α and β, $p(\lambda_0) \sim Gamma(\lambda_0, \alpha, \beta)$ where $\alpha, \beta > 0$, but not necessarily integers. Then, the prior at time t for the Poisson rate λ_t is given as,

$$\begin{aligned} p(\lambda_t; \alpha_{t-1}, \beta_{t-1}) &= Gamma\left(\lambda_t, ; \alpha_{t|t-1}, \beta_{t|t-1}\right) \\ &\propto \lambda_t^{\alpha_{t|t-1}-1} \exp\left\{-\beta_{t|t-1}\lambda_t\right\}, \end{aligned} \tag{5.47}$$

Note that we are considering only the m^{th} component, and thus the component superscript has been suppressed for convenience. At time t, assume that the total number of received measurements $||\mathbf{N}_t||$ is Poisson distributed with measurement rate λ_t. Let the likelihood of observing $||\mathbf{N}_t||$ measurements be given by,

$$\begin{aligned} p(||\mathbf{N}_t||; \lambda_t) &= Poiss(||\mathbf{N}_t||; \lambda_t) \\ &\propto \lambda_t^{||\mathbf{N}_t||} \exp\{-\lambda_t\}. \end{aligned} \tag{5.48}$$

It follows that the posterior distribution for λ_t is proportional to the product of the prior (5.47) and likelihood (5.48) such that,

$$\begin{aligned} p(\lambda_t|\mathbf{N}_t) &\propto Gamma\left(\lambda_t, ; \alpha_{t|t-1}, \beta_{t|t-1}\right) \times Poiss(||\mathbf{N}_t||; \lambda_t) \\ &= \lambda_t^{\alpha_{t|t-1}+||\mathbf{N}_t||-1} \exp\Big\{-(\beta_{t|t-1}+1)\lambda_t\Big\} \\ &= Gamma(\lambda_t; \alpha_{t|t-1}+||\mathbf{N}_t||, \beta_{t|t-1}+1). \end{aligned} \tag{5.49}$$

Thus the posterior distribution for λ_t is given by a gamma distribution with parameters $\alpha_t = \alpha_{t|t-1} + ||\mathbf{N}_t||$ and $\beta_t = \beta_{t|t-1} + 1$. It is important to realise that under this formulation, λ_t is *not* gamma distributed with parameters α and β, but the values of λ_t that maximise the auxiliary

function (5.43) are the expected values of a gamma distribution, with the following mean and variance,

$$E[\lambda_t] = \frac{\alpha_t}{\beta_t}, \tag{5.50}$$

$$\text{Var}[\lambda_t] = \frac{\alpha_t}{\beta_t^2}. \tag{5.51}$$

For the H-PMHT algorithm, the EM performs a maximisation and thus it is more appropriate to consider the mode of the gamma distribution:

$$\text{argmax}\, p(\lambda_t) = \frac{\alpha_t - 1}{\beta_t}. \tag{5.52}$$

It is important to note that we have not explicitly defined a model for the dynamics of the Poisson mixing rates $p(\lambda_t|\lambda_{t-1})$. Instead, we have assumed that the predicted density $p(\lambda_t|N_1, \ldots, N_{t-1})$ can be estimated through the posterior density at the previous time scan,

$$\begin{aligned} p(\lambda_t|N_1, \ldots, N_{t-1}) &= \int p(\lambda_t|\lambda_{t-1}) p(\lambda_{t-1}|N_1, \ldots, N_{t-1}) d\lambda_{t-1} \\ &\approx p(\lambda_{t-1}|N_1, \ldots, N_{t-1}), \end{aligned} \tag{5.53}$$

where $p(\lambda_{t-1}|N_1, \ldots, N_{t-1})$ is assumed to be Gamma distributed (5.47). This prior assumption for the Poisson mixing rates is sufficient for Bayesian estimation, however it is unclear how to resolve this in the batch case. The difficulty lies in the fact that there exists no conjugate distribution $p(\lambda_t|\lambda_{t-1})$ that ensures that the predicted density $p(\lambda_t|N_1, \ldots, N_{t-1})$ is also a Gamma distribution. In the case of batch smoothing, it may be necessary to approximate $p(\lambda_t|\lambda_{t-1})$ numerically through dynamic sampling methods using particles.

Granström [60] provides a framework for the prediction and update for the parameters of the measurement rate λ_t. In a multi-target context, let $\alpha^m_{t|t-1}$ and $\beta^m_{t|t-1}$ denote the predicted gamma parameters for component m where,

$$\begin{aligned} \alpha^m_{t|t-1} &= \exp\left\{\frac{-\delta_t}{\eta}\right\} \alpha^m_{t-1|t-1}, \\ \beta^m_{t|t-1} &= \exp\left\{\frac{-\delta_t}{\eta}\right\} \beta^m_{t-1|t-1}, \end{aligned} \tag{5.54}$$

and δ_t is the duration between time scans $t-1$ and t, and η is a time constant, which for convenience has been chosen to be the same for all components. Let $\alpha^m_{t|t}$ and $\beta^m_{t|t}$ denote the updated estimates for the gamma parameters for component m such that,

$$\begin{aligned} \alpha^m_{t|t} &= \alpha^m_{t|t-1} + ||\mathbf{N}^m_t||, \\ \beta^m_{t|t} &= \beta^m_{t|t-1} + 1. \end{aligned} \tag{5.55}$$

To calculate the updated estimate $\alpha_{t|t}^m$, we require an estimate for $||\mathbf{N}_t^m||$. Even though $||\mathbf{N}_t^m||$ is assumed to be an integer, it can be approximated via the following expression,

$$||\mathbf{N}_t^m|| \approx \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i, \tag{5.56}$$

where μ_t^{im} is given by (5.30). It is clear that we are approximating the integer $||\mathbf{N}_t^m||$ by a continuous value and thus, $\alpha_{t|t}$ is also not an integer. However, this is acceptable as there is no requirement for $\alpha_{t|t}$ to be an integer.

Note that the updated estimate β_t converges to an equilibrium given sufficient batch length. However, the prediction $\beta_{t|t-1}$ can be initialised such that the update β_t remains in a steady state for all time scans. By (5.55), the steady state for the $\beta_{t|t-1}^*$ term can be calculated for a given η and δ_t as following,

$$\beta_t^* = \frac{1}{1 - \exp\left(\frac{-\delta_t}{\eta}\right)}. \tag{5.57}$$

In the prediction stage, the constant η acts as a forgetting factor as it determines how much weight is applied to the past estimates of α and β. For large values of η, the predictions place more weighting on past estimates and the updated estimates will be highly correlated with time. In the case when the limit of $\eta \to 0$, the predicted gamma parameters $\alpha_{t|t-1}^m$ and $\beta_{t|t-1}^m \to 0$, and the updated estimates will be effectively uncorrelated with time. In this case, the Poisson mixing rates are estimated by,

$$\begin{aligned} \lambda_t^m &= \frac{\alpha_{t|t}^m}{\beta_{t|t}^m} \\ &= ||\mathbf{N}_t^m|| \\ &\approx \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \\ &= \lambda_t^{m'} \sum_{i=1}^{I} \frac{\boldsymbol{z}_t^i h^i(\boldsymbol{x}_t^{m'})}{\mathbb{f}_t^{i'}}. \end{aligned} \tag{5.58}$$

Observe that if we replace the intensity function $\mathbb{f}_t^{i'}$ with the probability $f_t^{i'}$, (5.58) is equivalent to the un-normalised multinomial mixing proportions p_t^m given by (4.61) in the standard H-PMHT. We can see that the H-PMHT algorithm formed under a Poisson measurement model generalises the standard H-PMHT under multinomial assumptions through the parameter η.

As we have imposed a dynamics model on the Poisson mixing rates, it is expected that the smoothed estimates of λ_t^m will provide a more robust measure of track quality than its multinomial counterpart π_t^m. Note that the multinomial mixing proportions π_t^m are dependent through a normalisation process, while the Poisson mixing rates λ_t^m are calculated independently due to

the Poisson thinning property. This is important for implementation, as it allows the H-PMHT algorithm under the Poisson measurement model to retain linear complexity with the number of targets.

5.4 Implementation

As discussed in subsection 5.3.5, the state estimation component for the H-PMHT under a Poisson measurement model can be implemented using an appropriate point measurement estimator. See subsection 4.3 for further details on target state estimation for both linear Gaussian and non-Gaussian psfs. For convenience, we now refer to the H-PMHT derived under a Poisson measurement model as the Poisson H-PMHT.

Initialisation can be performed via a peak detection process where a threshold is applied to the measurement image at every time scan and a track is initiated for every point that falls above that threshold. To initialise the Poisson mixing rates λ_0^m, we extract an estimated amplitude value for each peak detection from the measurement data. Assuming a steady state value for β, we can calculate values for α_0^m and β_0^m from λ_t^0. A summary for the implementation of the Poisson H-PMHT is provided in Algorithm 8. Comparing Algorithm 8 with the KF implementation of the standard H-PMHT in Algorithm 5, we can see that the steps in each algorithm are very similar, except that the Poisson H-PMHT now dynamically estimates both the target states and mixing component terms.

5.5 Simulations

This section demonstrates and verifies the performance of the proposed Poisson H-PMHT for a simulated linear Gaussian scenario, featuring a single target under four different target amplitude model assumptions. In each scenario, the state estimation errors and average target SNR estimates for the proposed Poisson H-PMHT is compared with the standard H-PMHT.

Define A_t to be the target amplitude at time t. The single target linear Gaussian scenario previously considered in Subsection 4.4.1 is again used here assuming four different target amplitude models:

- **Constant Amplitude Model**: Assumes a non-fluctuating target amplitude model such that A_t is given by the average target amplitude A,

$$A_t = A. \tag{5.59}$$

Algorithm 8 Poisson H-PMHT

1. Initialise the EM algorithm: For each target $m = 1 \ldots, M$, initialise the algorithm from a known $p(\boldsymbol{x}_0^m)$.

2. Initialise the Poisson mixing rates λ_0^m using an initial estimate for α_0^m and β_0^m for each component m.

3. For time scans $t = 1, \ldots, T$,

   (a) Prediction Step: For each target component m, compute the

   - predicted state estimates $\boldsymbol{x}_{t|t-1}^m$ by applying the dynamics model to the previous posterior state estimates,
   - Poisson mixing rates $\lambda_{t|t-1}^m$ by evaluating (5.52) and (5.54).

   (b) Using the predicted state estimate, construct $h^i(\boldsymbol{x}_{t|t-1}^m)$, the probability that a shot due to the target falls in pixel i using (4.9).

   (c) Update Step: For each target component m, compute

   - the updated state estimates $\boldsymbol{x}_{t|t}^m$ using an appropriate point measurement estimator,
   - Poisson mixing rates $\lambda_{t|t}^m$ by evaluating (5.52) and (5.55).

4. Repeat Step 3 b) ... 3 c) until convergence using the updated EM state estimates.

- **Slowly Varying Amplitude Model**: Assumes a target amplitude that fluctuates slowly between a minimum amplitude A_{min} and maximum amplitude A_{max}.

- **Swerling I Model**: Assumes the target amplitude follows an exponential distribution with mean A. An exponential random variable can be generated from a uniform random variable u_t by the following transformation [58]:

$$A_t = -A \log\{u_t\}. \tag{5.60}$$

- **Step Function with Gaussian Noise**: Assumes a target amplitude model based on a one-step function with additional Gaussian noise. Note that this noise is added to the amplitude component of the TkBD measurement model given in (2.11), and is different from the measurement noise $\boldsymbol{w}_t$ added to each image pixel.

Recall that the multinomial mixing proportions π_t^m can be interpreted as the average power from component m in the given image, while the Poisson mixing rates λ_t^m are interpreted as the absolute average power (average power in the image regardless of image size) from component m. However both terms can also be interpreted as the SNR from component m as the measurement model assumes Gaussian noise such that $\sigma_n^2 = 1$. In this case, power and SNR are equivalent. Under the Poisson measurement model, the absolute average SNR for component m can be calculated in dB follows,

$$SNR_t^{(m,P)} = 20 \log_{10}\{\lambda_t^m\}. \tag{5.61}$$

To compare SNR estimates from both algorithms, an absolute average target SNR value for the multinomial mixing proportions π_t^m is also required. If we multiply (5.58) by the quantity $\frac{\sum_s \lambda_t^s}{\sum_s \lambda_t^s}$, and following on from (5.2), we can see that the Poisson mixing rates λ_t^m is equivalent to p_t^m (4.61), the un-normalised value of π_t^m. Given this, an absolute average SNR for component m for the standard H-PMHT in dB can be calculated:

$$SNR_t^{(m,H)} = 20 \log_{10}\{p_t^m\}. \tag{5.62}$$

As we are only considering a linear Gaussian scenario, it is sufficient to verify the Poisson H-PMHT using a KF for the the target state estimation step. An KF implementation can be easily integrated into the algorithm in the same way as in the standard H-PMHT. See subsection 4.3.1 for more details. In the following simulations, 100 Monte Carlo runs were performed with randomised measurements with a forgetting factor $\eta = 10$ assumed for the Poisson H-PMHT. For both algorithms, it was found that ten iterations was sufficient to ensure EM convergence.

For the constant amplitude and Swerling I models, the average target amplitude was set to $A = 7$. For the slowly varying target amplitude model, the minimum and maximum amplitudes was set to $A_{min} = 5$ and $A_{max} = 10$, respectively. For the case when the target amplitude is characterised by a one-step function with Gaussian noise, the noise variance was set $\sigma^2 = 5$, with an initial constant target amplitude of 7 that changed to 12 midway through the scenario.

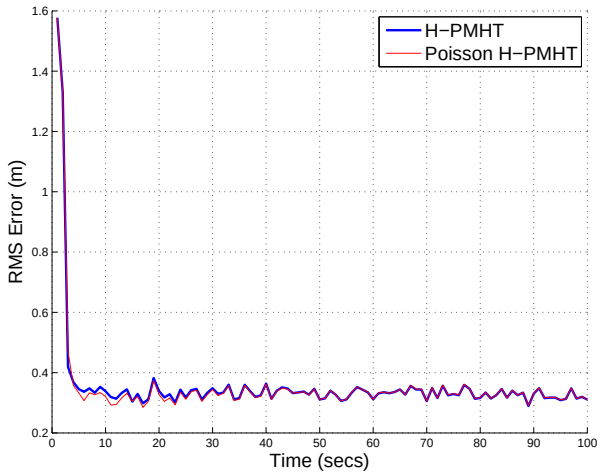
Figure 5.1 shows the root mean square (RMS) state estimation error for the Poisson H-PMHT and standard H-PMHT under each target amplitude model. In all four target amplitude scenarios, the state estimation performance of the Poisson H-PMHT (red line) is comparable to the standard H-PMHT (blue line). This is not unexpected as the state estimation component for each algorithm follows the same EM procedure, with different estimation procedures for the component mixing terms. Note that in Figure 5.1 (c), both algorithms seem to diverge slightly at approximately time scan 80 for the Swerling I model case.

As stated earlier, the key difference between the two algorithms lies with their calculations for the component mixing terms. Figure 5.2 shows the estimated SNR performance with time for each target amplitude model averaged over the 100 Monte Carlo runs. The true average target SNR is shown as a solid cyan line. As the results are averaged over 100 Monte Carlo runs, it is difficult to see the advantage of the Poisson mixing rates over the multinomial mixing proportions estimates; only in the Swerling I model case, do we observe that the Poisson H-PMHT outperforms the standard H-PMHT.

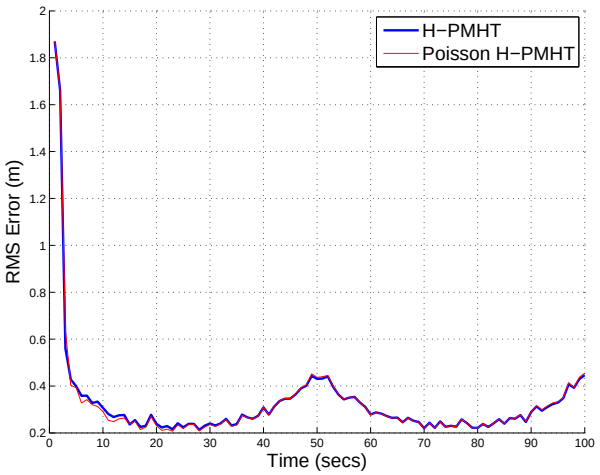
The differences in SNR performance between the two algorithms is more evident if we only consider a single Monte Carlo run. Figures 5.3 - 5.6 shows the estimated target SNR for each algorithm and target amplitude model for a single run. The true average target SNR is again shown as a solid cyan line and the instantaneous measured target SNR$= \log_{20} A_t$, is in dashed green. For each target amplitude model, the Poisson H-PMHT was implemented using a forgetting factor of $\eta = 1, 3$ and 10 to investigate the effect of varying the contribution of past estimates on updated estimates. Note that the performance of the standard H-PMHT is independent of the forgetting factor η.

Figure 5.3 (a) compares the estimated target SNR outputs from each algorithm for the constant amplitude scenario for a forgetting factor $\eta = 1$. We observe that the performance of the Poisson H-PMHT is very similar to the standard H-PMHT. This is expected as small values of η correspond to a high forgetting factor in the Poisson H-PMHT, resulting in SNR estimates that are largely time independent. As the standard H-PMHT estimates are also time independent, we expect the performance of two algorithms will be similar. This verifies that the Poisson H-PMHT generalises the standard H-PMHT through the forgetting factor η.

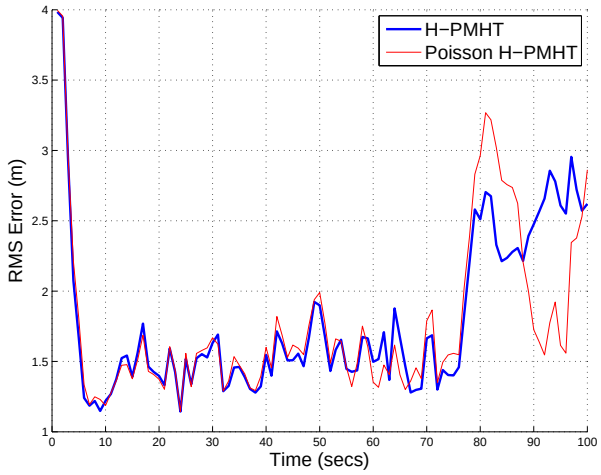
In Figure 5.3 (b), the forgetting factor for the Poisson H-PMHT is set to $\eta = 3$. We can clearly



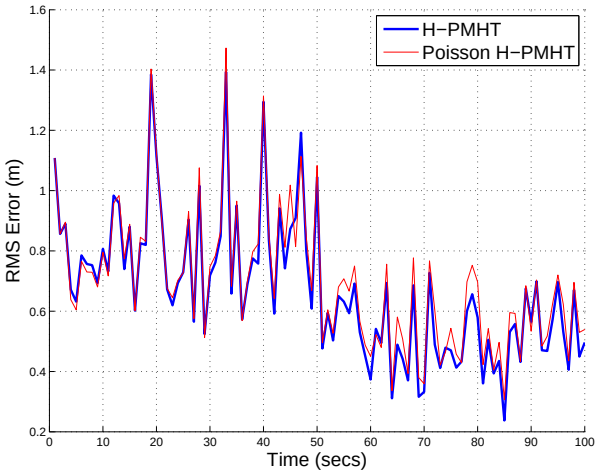
(a) Constant Amplitude Model



(b) Slowly Varying Amplitude Model

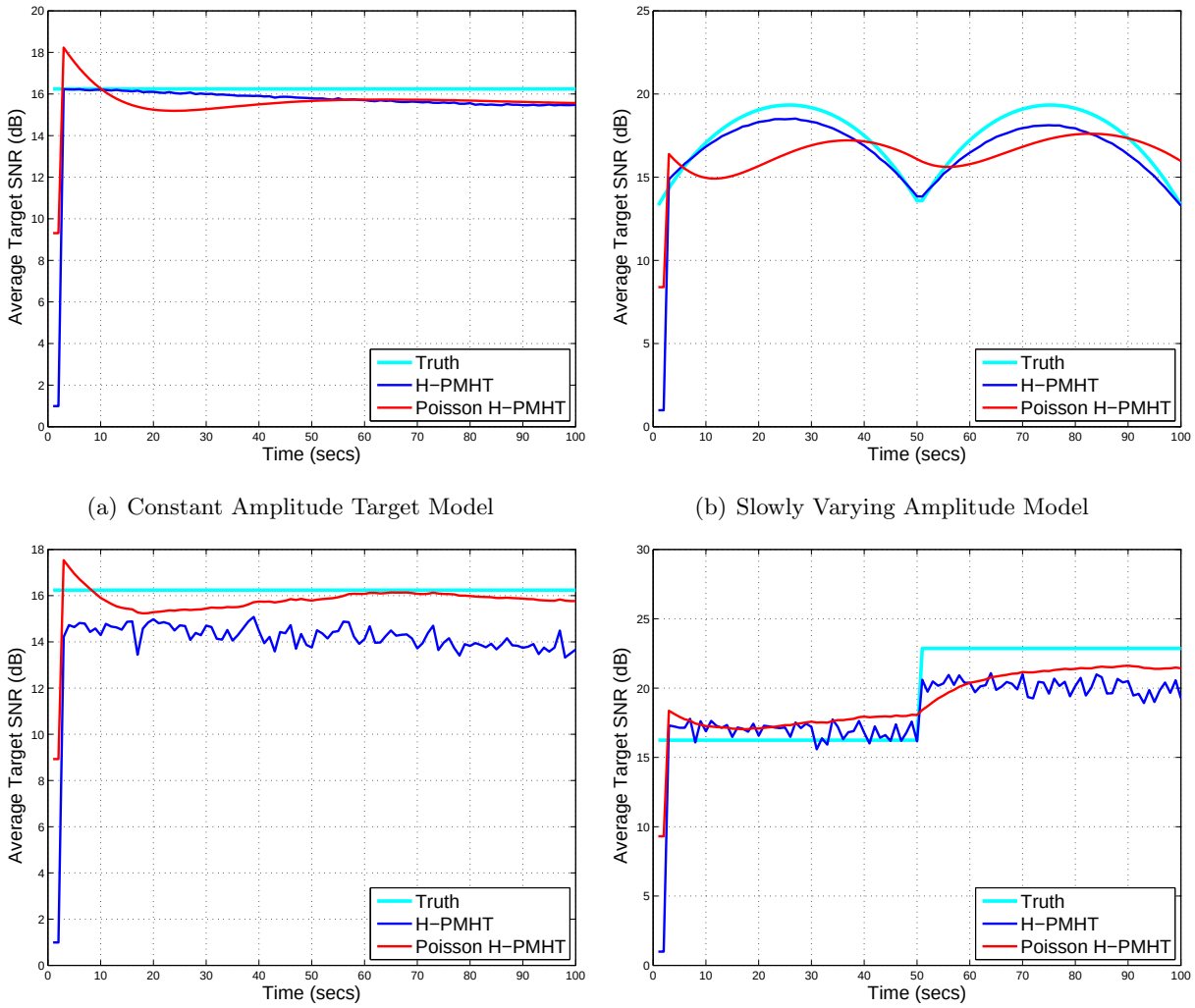


(c) Swerling I Model



(d) Step Function with Gaussian noise Model

Figure 5.1: RMS error averaged over 100 Monte Carlo runs comparing the standard H-PMHT with the Poisson H-PMHT under various target amplitude models for $\eta = 10$.



(a) Constant Amplitude Target Model

(b) Slowly Varying Amplitude Model

(c) Swerling I Model

(d) Step Function with Gaussian noise Scenario

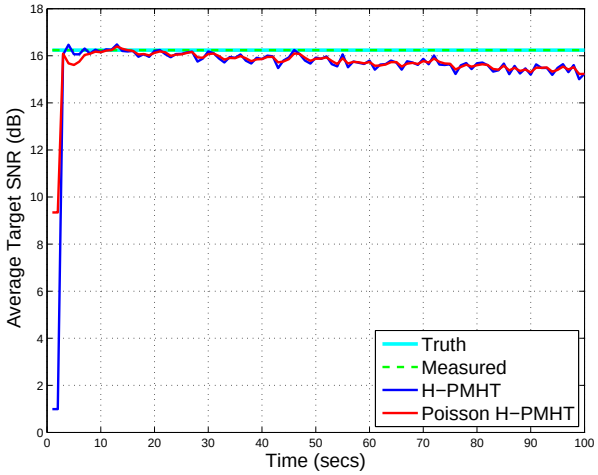
Figure 5.2: SNR averaged over 100 Monte Carlo runs comparing the standard H-PMHT with the Poisson H-PMHT under various target amplitude models for $\eta = 10$.

see that after initialisation, the Poisson H-PMHT reacts more slowly to the measurement data. The delay in the response time is even longer when we increase $\eta = 10$ in Figure 5.3 (c), as the Poisson H-PMHT mixing rates place increasingly more weight on past estimates. The same is observed for the slowly varying target amplitude scenario in Figure 5.4. As η increases, the Poisson H-PMHT takes progressively longer to respond to changes in amplitude. This is particularly obvious in Figure 5.4 (c) where the delay in response time results in a poor estimate for the target SNR in comparison to the standard H-PMHT. Clearly, in the case of the slowly varying amplitude model, the Poisson H-PMHT performs better under the assumption of time independent mixing terms.

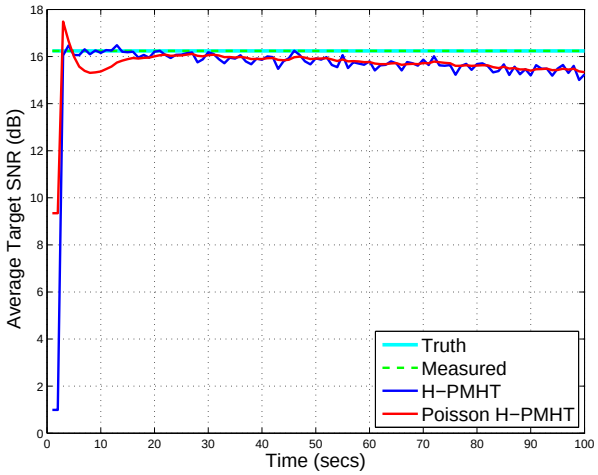
There are cases when the Poisson H-PMHT can benefit with assumption of time-correlated mixing terms. Consider Figure 5.5, which shows the estimated SNR for the Swerling I model amplitude scenario. As observed for the previous amplitude models, both algorithms again give similar performance when $\eta = 1$. However, as η increases, the SNR estimates from the Poisson H-PMHT becomes smoother, giving a better estimate of the true average target SNR. Similar results can also be observed for the step function amplitude case in Figure 5.6. In this scenario, the measurement noise level was selected so that the change in amplitude was not easily discernible. For $\eta = 3$, the Poisson H-PMHT gives a smoother estimate in comparison to the standard H-PMHT but nevertheless, both algorithms are unable to detect the step change in amplitude due to the high noise level in the measurement data. However, as the forgetting factor is increased to $\eta = 10$, the Poisson H-PMHT is able to distinguish a clear shift in the amplitude average after 50 scans, despite the high noise level.

We can see that when the target amplitude model features instantaneous fluctuations (e.g. Swerling I and step function amplitude scenarios), the Poisson H-PMHT clearly provides a more stable, less erratic prediction of the average target SNR for large η values. Due to the dynamics model imposed on the Poisson mixing rates λ_t^m, the SNR estimates are slower to respond to random fluctuations in the observed target SNR. This can be beneficial for track confirmation as it allows for a more stable test statistic for track quality. In contrast, the standard H-PMHT average SNR estimates are susceptible to variations in the measurement noise as it assumes there is no correlation with time.

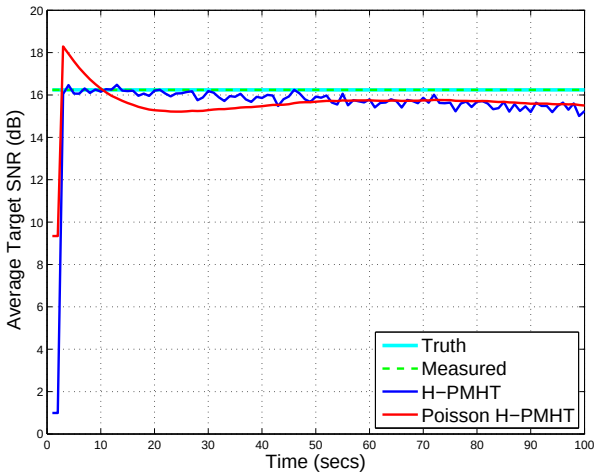
By allowing for a forgetting factor term, the Poisson H-PMHT is also able to reduce the track SNR variance estimates. This is evident in Figure 5.7, which shows the estimated track SNR variance versus forgetting factor η, averaged over 100 Monte Carlo runs and over all time scans. As mentioned earlier, the performance of the standard H-PMHT is independent of η and thus its variance remains constant with η. In contrast, the Poisson H-PMHT consistently gives smaller variance estimates than the standard H-PMHT. When the amplitude model is constant



(a) $\eta = 1$

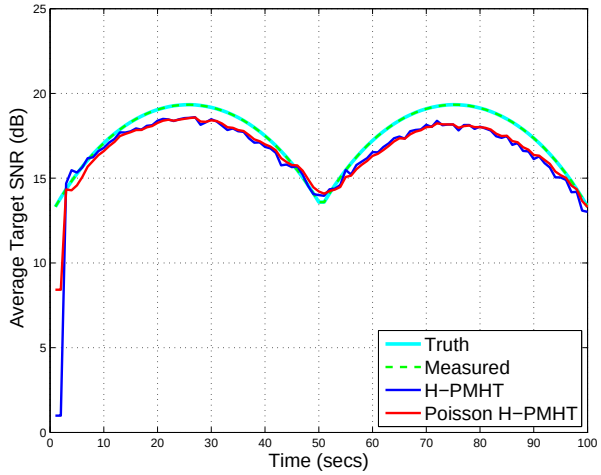


(b) $\eta = 3$

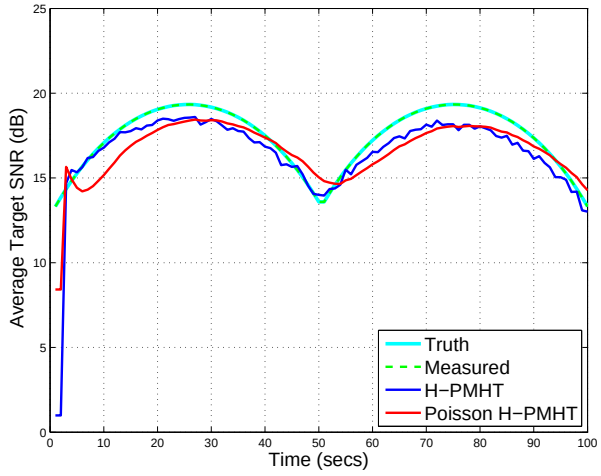


(c) $\eta = 10$

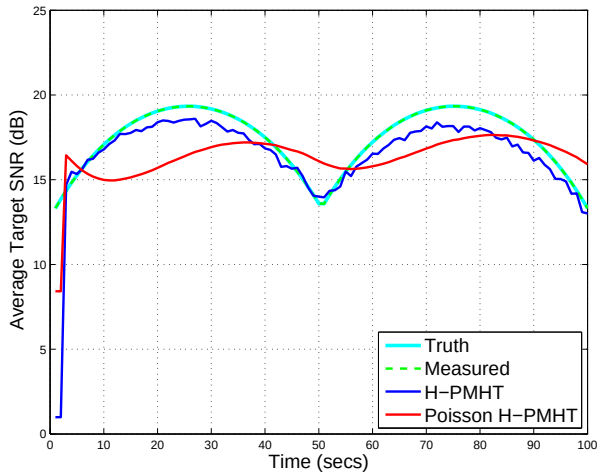
Figure 5.3: **Constant Amplitude scenario:** Comparison of the average target SNR for a single run for the standard H-PMHT and Poisson H-PMHT for varying forgetting factor η.



(a) $\eta = 1$

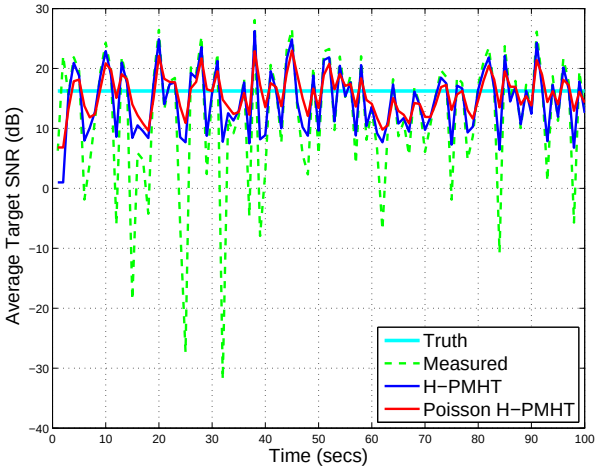


(b) $\eta = 3$

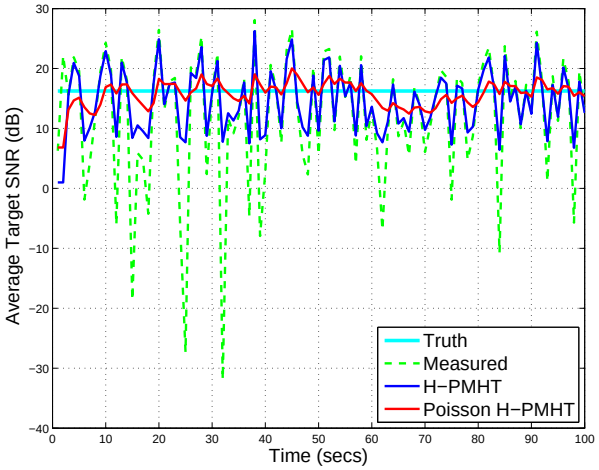


(c) $\eta = 10$

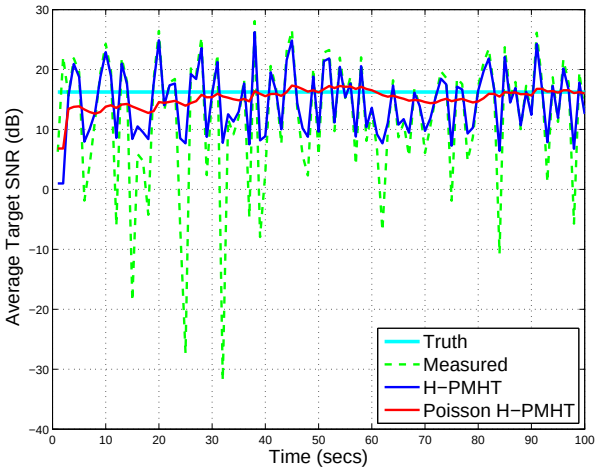
Figure 5.4: **Slowly Varying Amplitude Scenario:** Comparison of the average target SNR for a single run for the standard H-PMHT and Poisson H-PMHT for varying forgetting factor η.



(a) $\eta = 1$

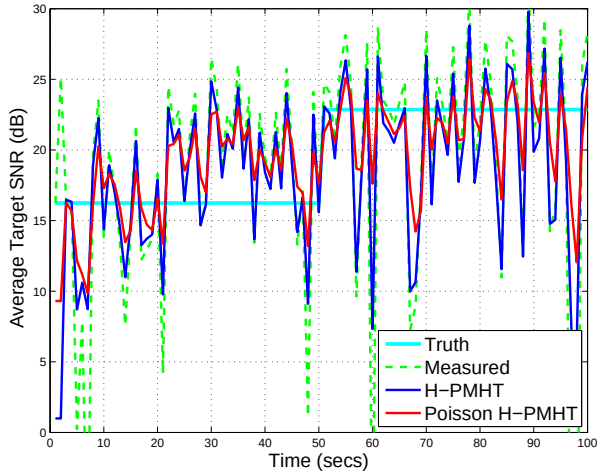


(b) $\eta = 3$

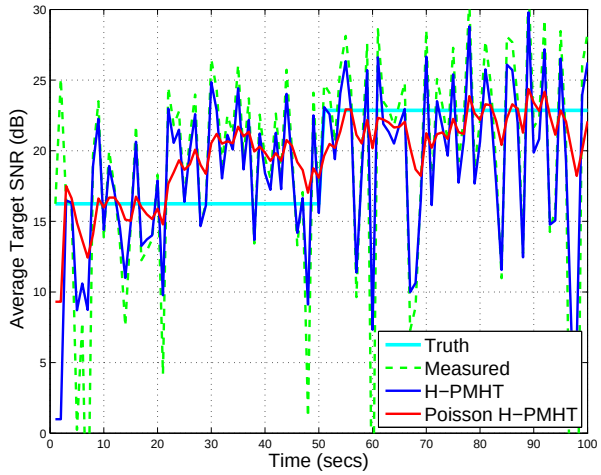


(c) $\eta = 10$

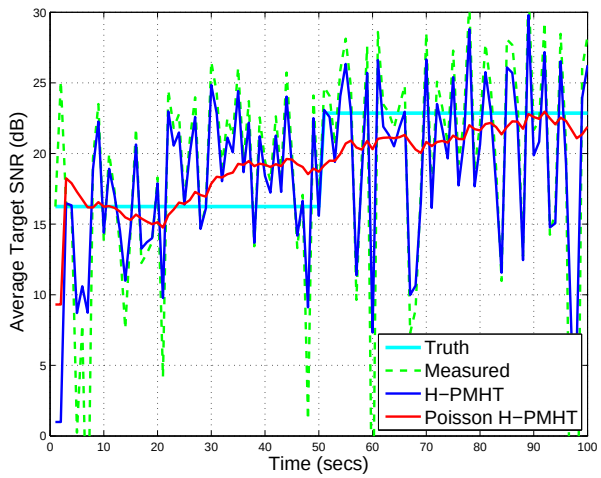
Figure 5.5: **Swerling I Scenario:** Comparison of the average target SNR for a single run for the standard H-PMHT and Poisson H-PMHT for varying forgetting factor η.



(a) $\eta = 1$



(b) $\eta = 3$



(c) $\eta = 10$

Figure 5.6: **Step Function with Gaussian noise Scenario:** Comparison of the average target SNR for a single run for the standard H-PMHT and Poisson H-PMHT for varying forgetting factor η.

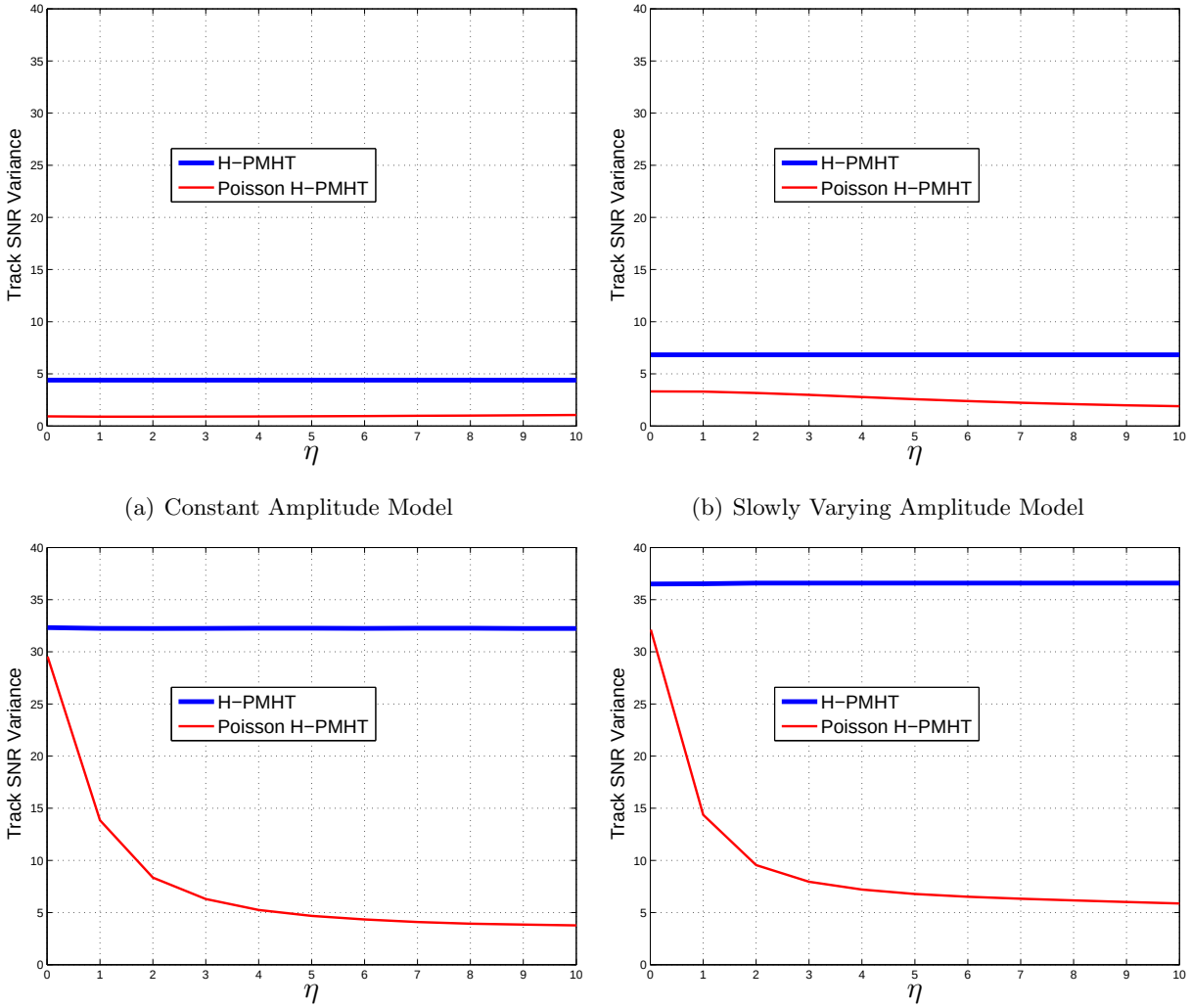
or smoothly varying as in Figure 5.7 (a) and (b), respectively, the Poisson H-PMHT's variance is also seemingly constant with η. However in Figure 5.7 (c) and (d), the Swerling I and step function target models feature highly fluctuating amplitudes and we see that the estimated variance of the Poisson H-PMHT decreases as η increases. The Poisson H-PMHT is able to reduce the target SNR variance by performing smoothing over a larger time window. Figure 5.7 also shows that as $\eta \to 0$, the performance of the Poisson H-PMHT starts to converge the performance of the standard H-PMHT, particularly in the Swerling I and step function cases. This further supports our claim that for small η values, the Poisson H-PMHT performance is equivalent to the standard H-PMHT.

In Figure 5.8, we further investigate the performance of the Poisson H-PMHT for the Swerling I amplitude model for different target SNR values. We observe that the variance estimates for the standard H-PMHT increases linearly with SNR. This is undesirable as it implies that even if a target has high SNR, the standard H-PMHT will struggle to give an accurate estimate of the target SNR as the estimated variance will also be high. On the other hand, we observe that the Poisson H-PMHT is able to reduce SNR variance through the smoothing factor η. Hence, in the case of highly fluctuating target models, we can see that the Poisson H-PMHT averages over a larger time frame to give significantly smaller variance estimates when compared with the standard H-PMHT.

5.6 Summary

The original H-PMHT assumes a multinomial distribution on the quantised image, which results in the derivation of component mixing proportion estimates that are uncorrelated with time. The H-PMHT also suffers from inconsistent tracking performance when the sensor surveillance region changes due to questionable independence assumptions.

In this chapter, we show that the multinomial assumption in the standard H-PMHT is consistent with a Poisson measurement model. The key contribution of this chapter is the first derivation and implementation of the H-PMHT assuming a Poisson measurement model. The new algorithm incorporates a time-correlated estimate of the component mixing terms by imposing a dynamics model on the Poisson measurement rate parameter. We also show that the Poisson H-PMHT is a generalisation of the H-PMHT under multinomial assumptions through a forgetting factor term. Another key contribution of this chapter is the proposal of an alternate state prior density to address some of the inconsistencies that arise due to quantisation issues in the original H-PMHT. Under linear Gaussian assumptions, the new state prior density results in a filtering performance that is independent of SNR. This ensures consistent tracking



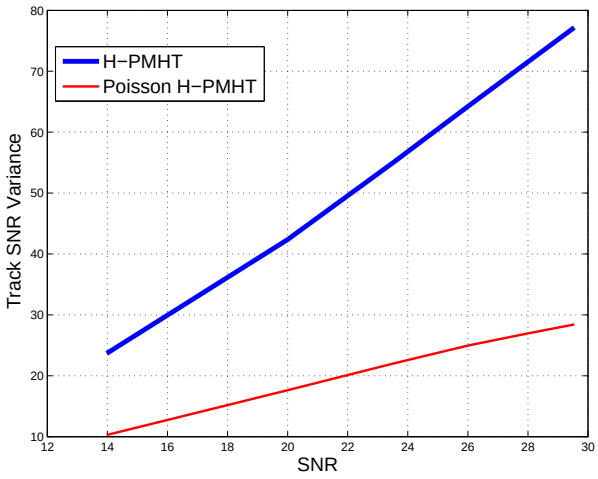
(a) Constant Amplitude Model

(b) Slowly Varying Amplitude Model

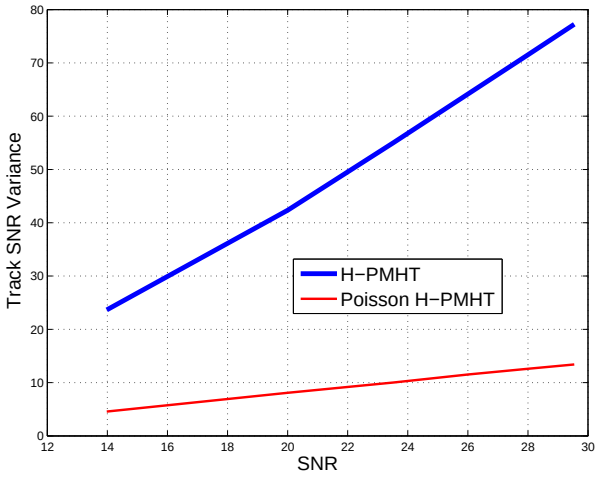
(c) Swerling I Model

(d) Step Function with Gaussian noise Model

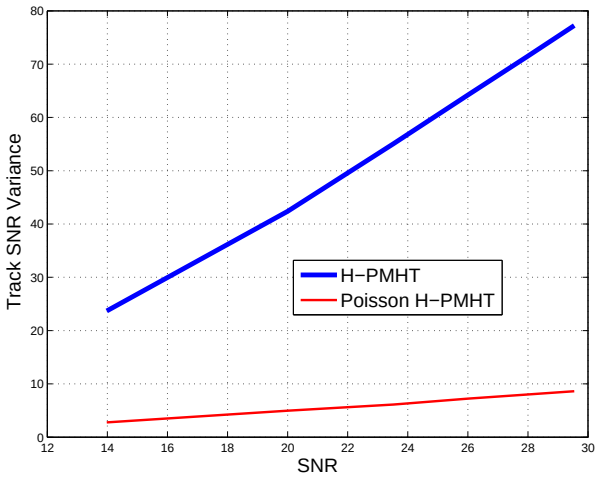
Figure 5.7: Track SNR variance versus forgetting factor η for the standard H-PMHT and Poisson H-PMHT under various target amplitude models.



(a) $\eta = 1$



(b) $\eta = 3$



(c) $\eta = 10$

Figure 5.8: Comparison of the average track SNR variance for the standard H-PMHT versus the Poisson H-PMHT for varying forgetting factor η for the Swerling I Scenario.

performance when the sensor surveillance region changes.

Through simulated scenarios, the Poisson H-PMHT is shown to be less sensitive to the observed fluctuations in a target's SNR. In scenarios featuring fluctuating target models, the Poisson H-PMHT is capable of providing a more consistent measure for track quality with smaller variance estimates through a smoothing factor term. This is an improvement on the standard H-PMHT, which struggles to estimate the underlying average target SNR for scenarios featuring fluctuating target models. The Poisson mixing rates also have the added advantage of being independent through the thinning properties of the Poisson process. This is in contrast to the multinomial mixing proportions in the standard H-PMHT, which are dependent through a normalising term. Although the Poisson H-PMHT is able to model the target SNR to a higher fidelity through the introduction of a dynamics model on the component mixing rates, the algorithm retains its linear complexity with the number of targets.

Chapter 6

An Interpolated Poisson Measurement Model for Track-Before-Detect

The original H-PMHT algorithm is based on an artificial quantisation procedure that is used to approximate the continuous intensity map measurement data with a set of discrete point measurements. In this chapter, we propose to eliminate this intermediate quantisation step by introducing an entirely new measurement model that is based on the novel application of the Probabilistic Multi-Hypothesis Tracker (PMHT) to continuous valued intensity maps. This is done by assuming the measurement image can be modelled by an interpolated version of the Poisson distribution. Like the Poisson measurement model discussed in Chapter 5, this new model also allows for a randomly evolving mean target amplitude estimate. Although the interpolated version of the Poisson distribution can only be shown to be a probability measure under certain conditions, we nevertheless make use of convenient properties to derive a TkBD algorithm for continuous valued intensity maps that is similar in principle to the standard H-PMHT. We refer to this algorithm as the Interpolated Poisson-PMHT (IP-PMHT).

The key contribution of this chapter is the first derivation of an alternative TkBD algorithm assuming an interpolated Poisson distribution on the energy generated by an individual target. Under this formulation, direct estimation of the measurement likelihood is possible, eliminating the need for an intermediate quantisation step. The main contributions of this chapter are summarised in conference article [134].

The chapter is arranged as follows; Section 6.1 discusses the motivation for modelling the measurement image directly rather than applying an approximation through quantisation; Section

6.2 introduces the interpolated version of the Poisson distribution and discusses the conditions under which it can be assumed to be a probability measure; Section 6.3 derives a new TkBD algorithm called the Interpolated Poisson-PMHT (IP-PMHT), that is based on the application of the interpolated Poisson measurement model and EM data association to continuous intensity maps; Section 6.5 verifies and compares the performance of the new algorithm with the standard H-PMHT and the Poisson H-PMHT introduced in Chapter 5; Section 6.6 summaries the key contributions for this chapter.

6.1 Introduction

The H-PMHT algorithm is a TkBD method that forms tracks by accessing the intensity map data directly. Generally, there exists no standard multi-target TkBD measurement model that can be used to describe the intensity map image, and it is difficult to perform filtering in the usual sense without applying an approximation to reduce the intensity map data to a set of discrete point measurements. The H-PMHT algorithm makes such an approximation by performing a quantisation [86] over the measurement image. This process converts the continuous valued data in the image into a collection of point measurements where the vector of counts $\mathbf{N}_t$ is assumed to follow a multinomial distribution. This results in the non-trivial assumption of independence between the shot measurements. Under the independence assumption, the quantisation level c^2 dictates the amount of information created in the synthetic data, yet in reality the true information content is independent of c^2. The H-PMHT's arbitrary choice of quantisation level combined with its assumption of independence between shots results in an infinite amount of synthetically generated data when the limit of the quantisation is taken to zero.

These assumptions have further ramifications when a Bayesian model is assumed as the infinite amount of synthetic data will also overwhelm any finite prior. In the standard H-PMHT, this problem is solved by modifying the Bayesian prior to ensure that it has sufficient influence on the target estimates. However this results in a data dependent prior term after the limit of quantisation is taken to zero. This in turn has an adverse effect on tracking performance whenever the sensor surveillance region changes. See subsection 4.5.2 for more details.

In Chapter 5, we proposed a solution to the data dependent prior issues by assuming a unique prior density for each component but the issues associated with the arbitrary choice of quantisation level and assumptions of independence remain unresolved. In this chapter, we propose an entirely different approach to address these limitations; rather than trying to retrospectively resolve the issues that stem from quantisation, we instead consider a novel measurement model that describes the continuous-valued measurement image directly using an interpolated version

of the Poisson distribution. We prove that interpolated version of the Poisson distribution approximates a probability measure and obeys an approximate superposition property for rate parameter $\lambda > 4$. Based on these assumptions, we derive an alternative TkBD algorithm that avoids the quantisation step and the subsequent use of a modified Bayesian prior inherent in the H-PMHT. The resulting TkBD algorithm is referred to as the Interpolated Poisson-PMHT (IP-PMHT).

The key contribution of this chapter is the *first derivation of an alternative TkBD algorithm assuming an interpolated Poisson distribution on the energy generated by an individual target.* This algorithm is similar in principle to the standard H-PMHT as it employs PMHT data associations but the application is to continuous valued data, rather than point measurements. Like the Poisson H-PMHT derived in Chapter 5, it naturally incorporates a stochastic model for the component mixing term.

6.2 Interpolated Poisson Distribution

Recall that a discrete random variable Z that follows a Poisson distribution with rate parameter $\lambda > 0$ has the following probability mass function:

$$p(Z = z; \lambda) = \exp(-\lambda)\frac{\lambda^z}{z!}, \tag{6.1}$$

for $z = 0, 1, 2 \ldots$. If however, Z is defined as a continuous random variable that can take on values $z \in \mathbb{R}_{\emptyset}^{+}$, where $\mathbb{R}_{\emptyset}^{+}$ is the set of all non-negative real numbers, a continuous version to (6.1) can be defined with the following form,

$$f(z; \lambda) = \frac{1}{K}\frac{\lambda^z}{\Gamma(z+1)}, \tag{6.2}$$

where K is some constant and $\Gamma(\cdot)$ is the well known Gamma function that interpolates $z!$ over continuous z. The function (6.2) can be thought of as an interpolated version of the Poisson distribution.

We now pose the following question: Is the interpolated version of the Poisson distribution a probability measure? If $f(z; \lambda)$ is in fact a probability density function (pdf), the parameter K corresponds to a normalising constant such that,

$$K = \int_0^{\infty} \frac{\lambda^z}{\Gamma(z+1)} dz. \tag{6.3}$$

Unfortunately an expression for K using (6.3) cannot be easily found as the integral in (6.3) cannot be computed analytically. Therefore, it is difficult to properly normalise the interpolated

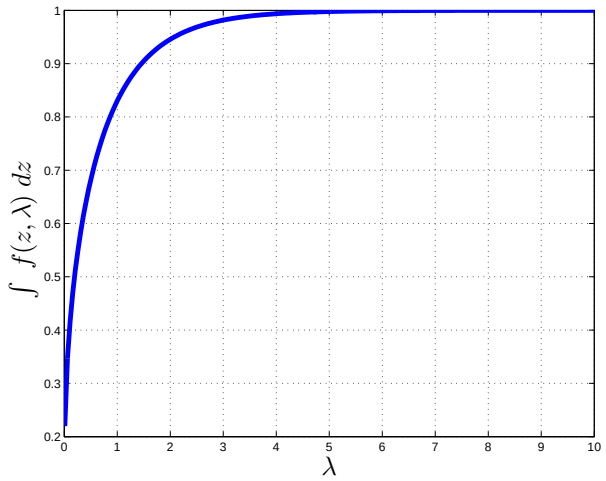


Figure 6.1: Integral of the Interpolated Poisson function for varying rate parameter λ.

Poisson distribution and mathematical manipulations are also restricted if K must be numerically evaluated. We can however consider approximations to solve the integral by employing Riemann integration, and assuming a step size $\triangle_n = 1$, the integral can be approximated by a power series:

$$\begin{aligned} K &= \int_0^\infty \frac{\lambda^z}{\Gamma(z+1)} dz \\ &\approx \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} \\ &= \exp(\lambda), \end{aligned} \tag{6.4}$$

where $\Gamma(n+1) = n!$ if n is a positive integer. Substituting (6.4) into (6.2) results in the following function,

$$f(z;\lambda) = \exp(-\lambda)\frac{\lambda^z}{\Gamma(z+1)}. \tag{6.5}$$

Observe that the function $f(z;\lambda)$ corresponds to an interpolated form of the Poisson distribution, however if the Riemann approximation holds, $f(z;\lambda)$ can also be considered a pdf itself in continuous random variable $z \in \mathbb{R}_{\mathbb{0}}^{+}$ with mean λ. Define $IPoiss(z;\lambda)$ to be the interpolated Poisson distribution with rate parameter λ, then we can write the following,

$$IPoiss(z;\lambda) \triangleq f(z;\lambda). \tag{6.6}$$

To gain an idea of how well the approximation (6.4) performs, consider Figure 6.1, which shows the integral of the $f(z;\lambda)$ over the variable z for varying values of λ. For small values of λ, i.e. $\lambda < 4$, the assumption that$f(z;\lambda)$ is a pdf is not valid. This is a result of overestimating K using the Riemann approximation, and thus λ, the assumed mean of the interpolated Poisson distribution is no longer correct.

In the wider literature, the notion of a "continuous Poisson distribution" is discussed in [79,127], although Ilienko [76] cautions that it should not be regarded as a continuous version of Poisson's discrete law. The successful application of this "distribution" is also discussed in [2, 72, 143].

In this chapter, we will treat the interpolated version of the Poisson distribution as a probability measure even though it is not strictly a pdf, and make use of its associated properties to derive a TkBD that is similar in principle to the H-PMHT. Like the Poisson distribution, we can show that the interpolated Poisson distribution obeys an approximate superposition principle for $\lambda > 4$. Superposition means that for $z = z^1 + z^2$ with $z^j \sim IPoiss(z^j; \lambda^j)$, the combined energy also follows an interpolated Poisson distribution with $z \sim IPoiss(z; \lambda^1 + \lambda^2)$. See Appendix A.1 for further details on the proof of approximate superposition for the interpolated Poisson distribution.

Note that derivation of the IP-PMHT depends on the validity of the superposition and pdf approximations. In later sections, we discuss the ramifications of making this assumption in the context of the TkBD problem.

6.3 Derivation

Assume a scenario in which a sensor observing M targets collects images $\mathbf{Z}_t$, defined by (4.1) as:

$$\mathbf{Z}_t = \{\boldsymbol{z}_t^1, \ldots, \boldsymbol{z}_t^I\},$$

at discrete times $t = 1 \ldots T$ where $\boldsymbol{z}_t^i$ denotes the energy in the i^{th} pixel of the sensor image at time t and I is the total number of observed pixels. The energy $\boldsymbol{z}_t^i$ in each of the measurement pixels is assumed to be continuous and as a result, it is difficult to formulate an appropriate measurement model.

Let $\boldsymbol{x}_t^m$ denote the state of component m at time t for $m = 0 \ldots M$. A component can be attributed to either a clutter or target object, therefore component $m = 0$ denotes the clutter contribution, which is assumed to be an empty set for all time scans. Assume that the remaining components $m = 1, \ldots, M$ are target objects that evolve according to a known process that may be non-linear and stochastic. Also define $h\left(\tau|\boldsymbol{x}_t^m\right)$ to be the the point spread function (psf) for the target components and $h^i(\boldsymbol{x}_t^m)$ to be the target per-pixel shot probabilities defined in (4.9).

The H-PMHT provides a convenient way to calculate the likelihood of observing the current measurement $\mathbf{Z}_t$ by employing a quantisation over the sensor data and assuming a multinomial distribution on the resulting integer-valued image. As shown in Chapter 5, this is consistent

with assuming that the number of discrete measurements from each component is Poisson distributed. Reducing the quantisation step size allows for finer precision but the resulting measurement count implies more information than is really present. We now propose an alternative measurement model that allows for a continuous measurement count.

Assume that the total energy in pixel i due to component m at time t, denoted by $\mathfrak{Z}_t^{im}$ follows an interpolated Poisson distribution,

$$p(\mathfrak{Z}_t^{im}; \boldsymbol{x}_t^m, \lambda_t^m) = \exp\left\{-\lambda_t^m h^i\left(\boldsymbol{x}_t^m\right)\right\} \frac{\left[\lambda_t^m h^i\left(\boldsymbol{x}_t^m\right)\right]^{\mathfrak{Z}_t^{im}}}{\Gamma(\mathfrak{Z}_t^{im}+1)}, \tag{6.7}$$

where as in Chapter 5, λ_t^m denotes the energy rate for component m, that is, the overall average amount of energy received by the sensor from component m per scan. The term $\lambda_t^m h^i\left(\boldsymbol{x}_t^m\right)$ denotes the rate of measurements from component m in pixel i at time t.

Following the discussion in subsection 6.2, we assume that the function (6.7) is a pdf even though this is strictly not correct. We have shown that like the Poisson distribution, it has the important property of superposition (see Appendix A.1). Denoting the energy due to component m at time t as $\mathfrak{Z}_t^m = \sum_{m=0}^{M} \mathfrak{Z}_t^{im}$, applying the superposition property over the image pixels results in

$$p(\mathfrak{Z}_t^m; \lambda_t^m) = \exp(-\lambda_t^m) \frac{\left[\lambda_t^m\right]^{\mathfrak{Z}_t^m}}{\Gamma(\mathfrak{Z}_t^m+1)}, \tag{6.8}$$

where $p(\mathfrak{Z}_t^m; \lambda_t^m)$ is also an interpolated Poisson distribution with rate parameter λ_t^m. Also define the i^{th} component of $\mathbf{Z}_t$ in (4.1) as,

$$\boldsymbol{z}_t^i = \sum_{m=0}^{M} \mathfrak{Z}_t^{im}, \tag{6.9}$$

as the total energy over all components m in pixel i at time t.

The parameters to be estimated under this new formulation are $\mathbf{X} = \boldsymbol{x}_{1:T}^{1:M}$, the collection of all component states m at all time scans t, and their associated measurement rates $\boldsymbol{\Lambda} = \lambda_{1:T}^{0:M}$. As in the standard H-PMHT, we again employ EM data association to estimate the target states and Poisson mixing rates with $\mathfrak{Z}_t^{im}$ considered as the missing data. Unlike in the Poisson H-PMHT algorithm where it is assumed that the shots in the quantised image are realisations of some mixture process, it is now assumed that the continuous valued *measurement image* itself can be modelled by some mixture process. The underlying mixture model intensity $\mathbb{f}$ and the per-pixel intensity $\mathbb{f}^i$ remain unchanged from the Poisson H-PMHT algorithm and are given by (5.8) and (5.9), respectively.

6.3.1 Expectation-Maximisation

The EM procedure for the IP-PMHT is similar to the standard H-PMHT and Poisson H-PMHT algorithms. As in the Poisson H-PMHT, the collection of Poisson rate parameters $\mathbf{\Lambda} = \lambda_{1:T}^{0:M}$ is part of the observer $\mathbf{O}$. As the measurements are now interpreted as energy and are no longer discrete variables, the assignment variable $\mathbf{K}$ no longer applies, instead we have the per-target energy $\mathfrak{Z}_t^{im}$. Define $\mathbf{Z} = \mathbf{Z}_{1:T}$ to be the collection of measurement images over time and $\mathfrak{Z} = \mathfrak{Z}_{1:T}^{1:I,1:M}$ to be the proportion of power from each mixture component for all time and pixels. The observer is given by $\mathbf{O} : \{\mathbf{X}, \mathbf{\Lambda}, \mathfrak{Z}, \mathbf{Z}\}$ and is assumed to be unknown.

The quantisation step is no longer required and the proportion of power from each mixture component is estimated directly. The variable $\mathfrak{Z}_t^{im}$ is considered as missing data under the EM algorithm. The assignments $\mathbf{K}$ of shots to targets and the precise locations $\mathbf{L}$ of each shot in each pixel that arises from measurement quantisation in the standard H-PMHT no longer apply under the interpolated Poisson measurement model.

6.3.2 E-Step

The new auxiliary function $Q^{(IP)}$ assuming an interpolated Poisson measurement model is given by,

$$Q^{(IP)}(\mathbf{X}, \mathbf{\Lambda} | \mathbf{X}', \mathbf{\Lambda}') = E_{\mathfrak{Z}} \left[\log \left\{ p(\mathbf{X}, \mathbf{\Lambda}, \mathfrak{Z}, \mathbf{Z}) \right\} | \mathbf{X}', \mathbf{\Lambda}', \mathbf{Z} \right], \tag{6.10}$$

where $\mathbf{X}'$ and $\mathbf{\Lambda}'$ denote the previous EM estimates for the target states and measurement rates, respectively. Simplifying the complete data density $p(\mathbf{X}, \mathbf{\Lambda}, \mathfrak{Z}, \mathbf{Z})$ results in

$$\begin{aligned} p(\mathbf{X}, \mathbf{\Lambda}, \mathfrak{Z}, \mathbf{Z}) &= p(\mathbf{Z}|\mathfrak{Z}, \mathbf{X}, \mathbf{\Lambda}) p(\mathfrak{Z}|\mathbf{X}, \mathbf{\Lambda}) p(\mathbf{X}) p(\mathbf{\Lambda}) \\ &= p(\mathbf{Z}|\mathfrak{Z}) p(\mathfrak{Z}|\mathbf{X}, \mathbf{\Lambda}) p(\mathbf{X}) p(\mathbf{\Lambda}), \end{aligned} \tag{6.11}$$

where $\mathbf{X}$ and $\mathbf{\Lambda}$ are assumed to be independent from each other. Note also that the measurement image $\mathbf{Z}$ is independent of the states $\mathbf{X}$ and rates $\mathbf{\Lambda}$. Substituting (6.11) into the auxiliary function (6.10) and only applying the expectation to terms that are dependent on $\mathfrak{Z}$, $Q^{(IP)}$ can be expanded as shown,

$$\begin{aligned} Q^{(IP)}(\mathbf{X}, \mathbf{\Lambda} | \mathbf{X}', \mathbf{\Lambda}') &= E_{\mathfrak{Z}} \left[\log \left\{ p(\mathbf{Z}|\mathfrak{Z}) p(\mathfrak{Z}|\mathbf{X}, \mathbf{\Lambda}) \; p(\mathbf{X}) \; p(\mathbf{\Lambda}) \right\} | \mathbf{X}', \mathbf{\Lambda}', \mathbf{Z} \right] \\ &= E_{\mathfrak{Z}} \left[\log \left\{ p(\mathfrak{Z}|\mathbf{X}, \mathbf{\Lambda}) \; p(\mathbf{X}) \; p(\mathbf{\Lambda}) \right\} | \mathbf{X}', \mathbf{\Lambda}', \mathbf{Z} \right]. \end{aligned} \tag{6.12}$$

Note that the $p(\mathbf{Z}|\mathfrak{Z})$ term can be removed from (6.12) as the variable $\mathbf{Z}$ is completely described by $\mathfrak{Z}$ such that,

$$p(\boldsymbol{z}_t^i|\mathfrak{Z}_t) = \begin{cases} 1, & \boldsymbol{z}_t^i = \sum_{m=0}^{M} \mathfrak{Z}_t^{im}, \\ 0, & \text{otherwise}, \end{cases} \tag{6.13}$$

where $\mathfrak{Z}_t$ denotes the energy from all components m at time t. Next, assume that the measurements $\mathfrak{Z}$ given $\mathbf{X}$ and $\mathbf{\Lambda}$ are independent across time, targets and pixels such that,

$$p(\mathfrak{Z}|\mathbf{X}, \mathbf{\Lambda}) = \prod_{t=1}^{T} \prod_{m=1}^{M} \prod_{i=1}^{I} p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^m, \lambda_t^m). \tag{6.14}$$

Substituting (6.14) into the auxiliary function (6.12) results in

$$\begin{aligned} Q^{(IP)}(\mathbf{X}, \mathbf{\Lambda}|\mathbf{X}', \mathbf{\Lambda}') &= \log\{p(\mathbf{X})\} + \log\{p(\mathbf{\Lambda})\} + E_{\mathfrak{Z}}\left[\log\left\{p(\mathfrak{Z}|\mathbf{X}, \mathbf{\Lambda})\right\}|\mathbf{X}', \mathbf{\Lambda}', \mathbf{Z})\right] \\ &= \log\{p(\mathbf{X})\} + \log\{p(\mathbf{\Lambda})\} + \int \log\left\{p(\mathfrak{Z}|\mathbf{X}, \mathbf{\Lambda}, \mathbf{Z})\right\} p(\mathfrak{Z}|\mathbf{X}', \mathbf{\Lambda}', \mathbf{Z}) d\mathfrak{Z} \\ &= \log\{p(\mathbf{X})\} + \log\{p(\mathbf{\Lambda})\} \\ &\qquad + \int \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{i=1}^{I} \log\left\{p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^m, \lambda_t^m)\right\} p(\mathfrak{Z}|\mathbf{X}', \mathbf{\Lambda}', \mathbf{Z}) d\mathfrak{Z} \\ &= \log\{p(\mathbf{X})\} + \log\{p(\mathbf{\Lambda})\} \\ &\qquad + \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{i=1}^{I} \int \log\left\{p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^m, \lambda_t^m)\right\} p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}, \boldsymbol{z}_t^i) d\mathfrak{Z}_t^{im}. \end{aligned} \tag{6.15}$$

In the second line of equation (6.15), the expectation over $\mathfrak{Z}$ converts to an integral as the measurements are continuous. Note also that in the last line of equation (6.15), the integral has been shifted to be inside the triple summation. The auxiliary function $Q^{(IP)}$ can be further simplified by substituting in (6.7) such that,

$$\log p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^m, \lambda_t^m) = -\lambda_t^m h^i(\boldsymbol{x}_t^m) + \mathfrak{Z}_t^{im} \log\{\lambda_t^m h^i(\boldsymbol{x}_t^m)\} - \log\Gamma(\mathfrak{Z}_t^{im} + 1). \tag{6.16}$$

Therefore, the integral in the last line of equation (6.15) can be expressed as,

$$\begin{aligned} &\int \log p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^m, \lambda_t^m) p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}, \boldsymbol{z}_t^i) d\mathfrak{Z}_t^{im} \\ &= -\lambda_t^m h^i(\boldsymbol{x}_t^m) + \log\left\{\lambda_t^m h^i(\boldsymbol{x}_t^m)\right\} \int_0^{\boldsymbol{z}_t^i} \mathfrak{Z}_t^{im} p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}, \boldsymbol{z}_t^i) d\mathfrak{Z}_t^{im} - C, \end{aligned} \tag{6.17}$$

where C is a constant with respect to $\mathbf{X}$ and $\mathbf{\Lambda}$.

We can see that the probability term on the rhs of (6.17) is the density of the missing data. This density can be further simplified by decomposing it into two terms, one consisting of the m^{th}

target and the other consisting of all remaining targets. First define $\widetilde{\mathfrak{Z}}_t^{im} = \boldsymbol{z}_t^i - \mathfrak{Z}_t^{im}$ to be the energy from all components excluding component m. This variable is also Gamma distributed with rate parameter,

$$\widetilde{\lambda h} = \sum_{s=0}^{M} \lambda_t^{s'} h^i(\boldsymbol{x}_t^{s'}) - \lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'}). \tag{6.18}$$

Then, the density on the rhs of (6.17) can be expressed as,

$$\begin{aligned} p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}, \boldsymbol{z}_t^i) &= \frac{p(\mathfrak{Z}_t^{im}, \widetilde{\mathfrak{Z}}_t^{im}, \boldsymbol{z}_t^i|\boldsymbol{x}_t^{m'}, \lambda_t^{m'})}{p(\boldsymbol{z}_t^i|\boldsymbol{x}_t^{m'}, \lambda_t^{m'})} \\ &= \frac{p(\boldsymbol{z}_t^i|\mathfrak{Z}_t^{im}, \widetilde{\mathfrak{Z}}_t^{im}) p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}) p(\widetilde{\mathfrak{Z}}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'})}{p(\boldsymbol{z}_t^i|\boldsymbol{x}_t^{m'}, \lambda_t^{m'})} \\ &= \frac{p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}) p(\widetilde{\mathfrak{Z}}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'})}{p(\boldsymbol{z}_t^i|\boldsymbol{x}_t^{m'}, \lambda_t^{m'})}. \end{aligned} \tag{6.19}$$

On the last line of (6.19), the $p(\boldsymbol{z}_t^i|\mathfrak{Z}_t^{im}, \widetilde{\mathfrak{Z}}_t^{im})$ term disappears due to (6.13).

As $\boldsymbol{z}_t^i$, $\mathfrak{Z}_t^{im}$ and $\widetilde{\mathfrak{Z}}_t^{im}$ all follow an interpolated Poisson distribution, substituting (6.7) for each of these variables results in

$$\begin{aligned} & p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}, \boldsymbol{z}_t^i) \\ = & \frac{\frac{1}{\Gamma\left(\mathfrak{Z}_t^{im}+1\right)} \exp\left\{-\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})\right\} \left[\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})\right]^{\mathfrak{Z}_t^{im}} \times \frac{1}{\Gamma\left(\widetilde{\mathfrak{Z}}_t^{im}+1\right)} \exp\left\{-\widetilde{\lambda h}\right\} \left[\widetilde{\lambda h}\right]^{\widetilde{\mathfrak{Z}}_t^{im}}}{\frac{1}{\Gamma\left(\boldsymbol{z}_t^i+1\right)} \exp\left\{-\sum_{s=0}^{M} \lambda_t^{s'} h^i(\boldsymbol{x}_t^{s'})\right\} \left[\sum_{s=0}^{M} \lambda_t^{s'} h^i(\boldsymbol{x}_t^{s'})\right]^{\boldsymbol{z}_t^i}} \\ = & \frac{\Gamma(\boldsymbol{z}_t^i+1)}{\Gamma(\mathfrak{Z}_t^{im}+1)\Gamma(\widetilde{\mathfrak{Z}}_t^{im}+1)} \times \frac{\left[\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})\right]^{\mathfrak{Z}_t^{im}} \left[\widetilde{\lambda h}\right]^{\widetilde{\mathfrak{Z}}_t^{im}}}{\left[\sum_{s=0}^{M} \lambda_t^{s'} h^i(\boldsymbol{x}_t^{s'})\right]^{\boldsymbol{z}_t^i}}. \end{aligned} \tag{6.20}$$

Substituting (6.20) into the integral on the rhs of (6.17), and replacing all $\widetilde{\mathfrak{Z}}_t^{im}$ with $\boldsymbol{z}_t^i - \mathfrak{Z}_t^{im}$, we can formulate the integral only in terms of the variable $\mathfrak{Z}_t^{im}$ such that,

$$\begin{aligned} & \int_0^{\boldsymbol{z}_t^i} \mathfrak{Z}_t^{im} p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'}, \lambda_t^{m'}, \boldsymbol{z}_t^i) d\mathfrak{Z}_t^{im} \\ = & \frac{1}{\left[\sum_{s=0}^{M} \lambda_t^{s'} h^i(\boldsymbol{x}_t^{s'})\right]^{\boldsymbol{z}_t^i}} \int_0^{\boldsymbol{z}_t^i} \frac{\mathfrak{Z}_t^{im} \Gamma(\boldsymbol{z}_t^i+1)}{\Gamma(\mathfrak{Z}_t^{im}+1)\Gamma(\boldsymbol{z}_t^i - \mathfrak{Z}_t^{im}+1)} \left[\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})\right]^{\mathfrak{Z}_t^{im}} \left[\widetilde{\lambda h}\right]^{\boldsymbol{z}_t^i - \mathfrak{Z}_t^{im}} d\mathfrak{Z}_t^{im}. \end{aligned} \tag{6.21}$$

In Appendix A.2, we show that the integral on the rhs of (6.21) can be decomposed as follows,

$$\begin{aligned}
&\int_0^{\boldsymbol{z}_t^i} \frac{\mathfrak{Z}_t^{im}\Gamma(\boldsymbol{z}_t^i+1)}{\Gamma(\mathfrak{Z}_t^{im}+1)\Gamma(\boldsymbol{z}_t^i-\mathfrak{Z}_t^{im}+1)} \left[\lambda_t^{m'}h^i(\boldsymbol{x}_t^{m'})\right]^{\mathfrak{Z}_t^{im}} \left[\widetilde{\lambda h}\right]^{\boldsymbol{z}_t^i-\mathfrak{Z}_t^{im}} d\mathfrak{Z}_t^{im} \\
\propto &\int_0^{\boldsymbol{z}_t^i} \mathfrak{Z}_t^{im} IPoiss\Big(\mathfrak{Z}_t^{im};\lambda_t^{m'}h^i(\boldsymbol{x}_t^{m'})\Big) \times IPoiss\Big(\boldsymbol{z}_t^i-\mathfrak{Z}_t^{im};\widetilde{\lambda h}\Big) d\mathfrak{Z}_t^{im}. \qquad (6.22)
\end{aligned}$$

Using characteristic functions, we then show that the left hand side of (6.22) simplifies to,

$$\begin{aligned}
&\int_0^{\boldsymbol{z}_t^i} \frac{\mathfrak{Z}_t^{im}\Gamma(\boldsymbol{z}_t^i+1)}{\Gamma(\mathfrak{Z}_t^{im}+1)\Gamma(\boldsymbol{z}_t^i-\mathfrak{Z}_t^{im}+1)} \left[\lambda_t^{m'}h^i(\boldsymbol{x}_t^{m'})\right]^{\mathfrak{Z}_t^{im}} \left[\widetilde{\lambda h}\right]^{\boldsymbol{z}_t^i-\mathfrak{Z}_t^{im}} d\mathfrak{Z}_t^{im} \\
= &\lambda_t^{m'}h^i(\boldsymbol{x}_t^{m'})\ \boldsymbol{z}_t^i \left[\sum_{s=0}^{M}\lambda_t^{s'}h^i(\boldsymbol{x}_t^{s'})\right]^{\boldsymbol{z}_t^i-1}, \qquad (6.23)
\end{aligned}$$

Substituting (6.23) into (6.21) results in

$$\int_0^{\boldsymbol{z}_t^i} \mathfrak{Z}_t^{im} p(\mathfrak{Z}|\boldsymbol{x}_t^{m'},\lambda_t^{m'},\boldsymbol{z}_t^i) d\mathfrak{Z}_t^{im} = \mu_t^{im}\boldsymbol{z}_t^i, \qquad (6.24)$$

where μ_t^{im} has been previously defined in (5.30):

$$\mu_t^{im} = \frac{\lambda_t^{m'}h^i\left(\boldsymbol{x}_t^{m'}\right)}{\sum_{s=0}^{m}\lambda_t^{s'}h^i\left(\boldsymbol{x}_t^{s'}\right)}, \qquad (6.25)$$

Note that $0 \leq \mu_t^{im} \leq 1$ represents the relative proportion of power from target m in pixel i. Also note that the $\mu_t^{im}\boldsymbol{z}_t^i$ term in (6.21) can be interpreted as the amount of energy in pixel i from component m and is the mean to the missing data density $p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'},\lambda_t^{m'},\boldsymbol{z}_t^i)$.

Substituting (6.24) into (6.17) results in

$$\int \log p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^m,\lambda_t^m) p(\mathfrak{Z}_t^{im}|\boldsymbol{x}_t^{m'},\lambda_t^{m'},\boldsymbol{z}_t^i) d\mathfrak{Z}_t^{im} = -\lambda_t^m h^i\left(\boldsymbol{x}_t^m\right) + \mu_t^{im}\boldsymbol{z}_t^i \log\lambda_t^m + \mu_t^{im}\boldsymbol{z}_t^i \log h^i\left(\boldsymbol{x}_t^m\right). \qquad (6.26)$$

A final expression for the auxiliary function can be found by substituting (6.26) into (6.12), and making use of $\sum_i h^i\left(\boldsymbol{x}_t^m\right) \approx 1$:

$$\begin{aligned}
Q^{(IP)}(\mathbf{X},\boldsymbol{\Lambda}|\mathbf{X}',\boldsymbol{\Lambda}') = \sum_{m=0}^{M}\Bigg[&\log\ p(\boldsymbol{x}_0^m) + \sum_{t=1}^{T}\log\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\} + \log\{p(\lambda_0^m)\} \\
&+ \sum_{t=1}^{T}\log\{p(\lambda_t^m|\lambda_{t-1}^m)\} - \sum_{t=1}^{T}\lambda_t^m + \sum_{t=1}^{T}\log\lambda_t^m \sum_{i=1}^{I}\mu_t^{im}\boldsymbol{z}_t^i + \sum_{t=1}^{T}\sum_{i=1}^{I}\mu_t^{im}\boldsymbol{z}_t^i \log h^i\left(\boldsymbol{x}_t^m\right)\Bigg]. \qquad (6.27)
\end{aligned}$$

Note that we have ignored all terms that are considered constant with respect to $\mathbf{X}$ and $\boldsymbol{\Lambda}$ in (6.17).

As in the standard and Poisson H-PMHT algorithms, the auxiliary function can be arranged into two parts, one for the estimation of $\mathbf{X}$, and the other for the estimation of the mixing terms $\boldsymbol{\Lambda}$,

$$Q^{(IP)} = \sum_{m=1}^{M} Q_X^m + \sum_{t=1}^{T} Q_{t\lambda}, \tag{6.28}$$

such that

$$Q_X^m = \log\ p(\boldsymbol{x}_0^m) + \sum_{t=1}^{T} \left[\log\{p(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m)\} + \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \log h^i\left(\boldsymbol{x}_t^m\right) \right], \tag{6.29}$$

$$Q_{t\lambda} = \sum_{m=0}^{M} \left[\log\left\{p(\lambda_0^m)\right\} + \log\{p(\lambda_t^m | \lambda_{t-1}^m)\} + \log \lambda_t^m \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \right] - \lambda_t, \tag{6.30}$$

where the last term in (6.30) is $\lambda_t = \sum_{t=1}^{T} \lambda_t^m$. Expressing the auxiliary function in this manner allows each target component and measurement rate to be optimised independently in the M-step of the EM algorithm. Comparing (6.30) with the Poisson H-PMHT auxiliary function (5.43), we see that $Q_{t\lambda}$ derived under the interpolated Poisson measurement model is identical to the measurement rate auxiliary function derived under a Poisson measurement model after the limit of the measurement quantisation is taken to zero.

Recall that the derivation of the auxiliary function (6.28) is based on (6.24) being true. However the evaluation of this integral is based on the assumption that the interpolated Poisson distribution is in fact a probability measure. By Figure 6.1, it is evident that this is not always the case. We now discuss the ramifications of making these modelling assumptions.

In (6.22), the target energy in each pixel, $\mathfrak{Z}_t^{im}$ is assumed to follow an interpolated Poisson distribution with rate parameter $\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})$. For small values of $\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})$, the Riemann approximation discussed in subsection 6.2 overestimates the normalising constant for the interpolated Poisson distribution. This implies that the mean $\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})$ is incorrect. As the μ_t^{im} term depends on $\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})$, this implies that the $\mu_t^{im} \boldsymbol{z}_t^i$ term, which approximates the mean to the missing density $p(\mathfrak{Z}_t^{im} | x_t^{m'}, \lambda_t^{m'}, \boldsymbol{z}_t^i)$, is also underestimated.

For practical applications, small values of $\lambda_t^{m'} h^i(\boldsymbol{x}_t^{m'})$ correspond to small pixel energy levels observed for component m. This generally occurs in the tails of the target psf and thus the IP-PMHT algorithm underestimates the tails in the target psf. Overall, this implies that the IP-PMHT algorithm incorrectly penalises the target states as the the mean energy of each target is underestimated. Nevertheless, we can assume that this error will be marginal as the contribution of the tails to the target energy is generally small. In the case when the target peak SNR is also small, the consequences of underestimating the contribution of the tails to the target psf can be more significant. However, note that most algorithms will generally struggle to form tracks on targets with such low levels of SNR.

In the evaluation the state auxiliary function Q_X^m, we are required to evaluate the integral,

$$\log h^i\left(\boldsymbol{x}_t^m\right) = \log \int_{B_i} h\left(\tau|\boldsymbol{x}_t^m\right) d\tau. \tag{6.31}$$

While the evaluation of the logarithm of a function is a non-trivial operation, we can make the following approximation by employing Jensen's Inequality:

$$\log \int_{B_i} h\left(\tau|\boldsymbol{x}_t^m\right) d\tau \geq \int_{B_i} \log h\left(\tau|\boldsymbol{x}_t^m\right) d\tau. \tag{6.32}$$

The rhs of (6.32) is of a form that is more easily evaluated, particularly in a linear Gaussian scenario when the $\log h\left(\tau|\boldsymbol{x}_t^m\right)$ equates to a quadratic. Using Jensen's Inequality (6.32), a lower bound on the auxiliary function Q_X^m can be formed such that,

$$Q_X^m \geq \widetilde{Q}_X^m, \tag{6.33}$$

where $\widetilde{Q}_X^m$ denotes the modified state auxiliary function using Jensen's Inequality:

$$\widetilde{Q}_X^m = \log\ p(\boldsymbol{x}_0^m) + \sum_{t=1}^{T} \left[\log\{p(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)\} + \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} \log h\left(\tau|\boldsymbol{x}_t^m\right) \right]. \tag{6.34}$$

Under this modification, the maximisation step is performed on a different type of lower bound than what is considered in the standard H-PMHT auxiliary function. As such, by maximising this new lower bound, we are no longer performing EM in the usual sense and the convergence property of the EM method is no longer valid.

We show later in subsection 6.5 that for a linear Gaussian scenario under various amplitude models, the local maximum of the modified auxiliary function $\widetilde{Q}_X^m$ still converges to a sensible estimate that aligns closely with the EM estimates from the standard H-PMHT and Poisson H-PMHT. However, it is important to realise that the local maximum of the modified auxiliary function may occur at a different $\boldsymbol{x}_t^m$ value from the local maximum of the original auxiliary function (6.28) formed under EM.

For the maximisation of the measurement rate component, note that the measurement rate auxiliary function (6.30) is identical to the Poisson H-PMHT auxiliary function (5.43). As a result, the maximisation procedure described in Chapter 5 can again be used to calculate smoothed estimates for the mixing proportion terms. See subsection 5.3.5 for further details.

6.4 Kalman Filter Implementation

In this subsection, a KF implementation for the IP-PMHT is presented. Like the original H-PMHT KF implementation introduced in subsection 4.3.1, an equivalent point measurement and covariance for the case when the measurement function is linear and Gaussian can

be derived. In order to do so, we can follow the same procedure described in 4.3.1 to show that the measurement component of the modified auxiliary function (6.34) is equivalent to a single quadratic such that,

$$\sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} \log h\left(\tau|\boldsymbol{x}_t^m\right) d\tau = C - \frac{1}{2}(H\boldsymbol{x}_t^m - \tilde{\boldsymbol{z}}_{t,m}^{(IP)})^T \tilde{\mathsf{R}}_{t,m}^{(IP)^{-1}} (H\boldsymbol{x}_t^m - \tilde{\boldsymbol{z}}_{t,m}^{(IP)}), \tag{6.35}$$

where C is some constant and $\tilde{\boldsymbol{z}}_{t,m}^{(IP)}$ and $\tilde{\mathsf{R}}_{t,m}^{(IP)}$ are the equivalent synthetic point measurement and covariance for the IP-PMHT, respectively.

Consider simplifying the left hand side of (6.35):

$$\begin{aligned}
& \sum_i \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} \log h\left(\tau|\boldsymbol{x}_t^m\right) d\tau \\
= \;& C - \frac{1}{2} \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} (H\boldsymbol{x}_t^m - \tau)^T \mathsf{R}^{-1} (H\boldsymbol{x}_t^m - \tau) d\tau \\
= \;& C - \frac{1}{2} \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} (H\boldsymbol{x}_t^m)^T \mathsf{R}^{-1} H\boldsymbol{x}_t^m - 2(H\boldsymbol{x}_t^m)^T R^{-1} \tau d\tau \\
= \;& C - \frac{1}{2} \left[\sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i (H\boldsymbol{x}_t^m)^T \mathsf{R}^{-1} H\boldsymbol{x}_t^m \int_{B_i} d\tau - 2 \sum_i \mu_t^{im} \boldsymbol{z}_t^i (H\boldsymbol{x}_t^m)^T R^{-1} \int_{B_i} \tau d\tau \right] \\
= \;& C - \frac{1}{2} \left[(H\boldsymbol{x}_t^m)^T \tilde{\mathsf{R}}_{t,m}^{(IP)^{-1}} H\boldsymbol{x}_t^m - 2 \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i (H\boldsymbol{x}_t^m)^T \mathsf{R}^{-1} \int_{B_i} \tau d\tau \right],
\end{aligned} \tag{6.36}$$

where the synthetic covariance $\tilde{\mathsf{R}}^{(IP)}$ is given by,

$$\begin{aligned}
\tilde{\mathsf{R}}_{t,m}^{(IP)} &= \frac{1}{\sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} d\tau} \mathsf{R} \\
&= \frac{1}{\sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i |B_i|} \mathsf{R}.
\end{aligned} \tag{6.37}$$

Note that all terms in (6.36) that are not dependent on $\boldsymbol{x}_t^m$ are absorbed into the constant C. The term $|B_i| = \int_{B_i} d\tau$ is defined as the size of pixel i and all terms that are not dependent on $\boldsymbol{x}_t^m$ are absorbed into the constant C. The second term on the last line of (6.36) can be further simplified to give,

$$\begin{aligned}
2 \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i (H\boldsymbol{x}_t^m)^T \mathsf{R}^{-1} \int_{B_i} \tau d\tau &= 2(H\boldsymbol{x}_t^m)^T \mathsf{R}^{-1} \frac{\sum_{j=1}^{I} \mu_t^{jm} \boldsymbol{z}_t^j |B_j|}{\sum_{j=1}^{I} \mu_t^{jm} \boldsymbol{z}_t^j |B_j|} \sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} \tau d\tau \\
&= 2(H\boldsymbol{x}_t^m)^T \tilde{\mathsf{R}}_{t,m}^{(IP)^{-1}} \frac{\sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} \tau d\tau}{\sum_{j=1}^{I} \mu_t^{jm} \boldsymbol{z}_t^j |B_j|}, \\
&= 2(H\boldsymbol{x}_t^m)^T \tilde{\mathsf{R}}_{t,m}^{(IP)^{-1}} \tilde{\boldsymbol{z}}_{t,m}^{(IP)},
\end{aligned} \tag{6.38}$$

where the synthetic measurement $\tilde{\boldsymbol{z}}_{t,m}^{(IP)}$ is given by,

$$\tilde{\boldsymbol{z}}_{t,m}^{(IP)} = \frac{\sum_{i=1}^{I} \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} \tau d\tau}{\sum_{j=1}^{I} \mu_t^{jm} \boldsymbol{z}_t^j |B_j|}. \tag{6.39}$$

Substituting (6.38) into the measurement component (6.36) results in

$$\sum_i \mu_t^{im} \boldsymbol{z}_t^i \int_{B_i} \log h\left(\tau | \boldsymbol{x}_t^m\right) d\tau = C - \frac{1}{2}\left[(H\boldsymbol{x}_t^m - \tilde{\boldsymbol{z}}_{t,m}^{(IP)})^T \tilde{\mathsf{R}}_{t,m}^{(IP)^{-1}} (H\boldsymbol{x}_t^m - \tilde{\boldsymbol{z}}_{t,m}^{(IP)}) \right], \tag{6.40}$$

The KF implementation for the IP-PMHT is summarised in Algorithm 9.

6.5 Simulations

This section demonstrates and verifies the performance of the proposed IP-PMHT for several simulated scenarios featuring different target fluctuation models. The four target amplitude models described in Subsection 5.5 are again used here, namely the constant amplitude model, slowly varying amplitude model, Swerling I model, and step function with Gaussian noise amplitude model. As in previous chapters, we only consider a linear Gaussian scenario as described in subsection 4.4.1. The initialisation procedure and parameters for estimating the mixing proportion λ_t^m such as the forgetting factor η are the same as used in Subsection 5.5. For the following simulations, the algorithm was run for 50 EM iterations and all results were also averaged over 100 Monte Carlo runs.

Recall that at the end of subsection 6.3.2, the state auxiliary function (6.34) was modified under Jensen's Inequality such that the maximisation step is now performed on a different type of lower bound that what is derived in the standard H-PMHT. As a result, we are no longer performing EM in the usual sense and the convergence property of the EM method no longer holds. In this section, we demonstrate that for a number of linear Gaussian scenarios, the new lower bound defined by (6.34) still converges.

In order to guarantee convergence for the IP-PMHT, the modified state auxiliary function is required to be non-decreasing at at each maximisation step:

$$\widetilde{Q}_X^{j+1}(\hat{\boldsymbol{x}}^{j+1}) \geq \widetilde{Q}_X^{j}(\hat{\boldsymbol{x}}^{j}), \tag{6.41}$$

where the superscript index j is the iteration index. Note that the target index m has been suppressed for notational convenience. In (6.41), the modified auxiliary function at the $(j+1)^{th}$ iteration evaluated at the maximising state $\hat{\boldsymbol{x}}^{j+1}$ is required to be greater than the modified auxiliary function at the j^{th} iteration evaluated at its corresponding maximum $\hat{\boldsymbol{x}}^{j}$. However,

Algorithm 9 Interpolated Poisson-PMHT using a KF implementation

1. Initialise the EM algorithm: For each target $m = 1 \ldots, M$, initialise the algorithm from a known $p(\boldsymbol{x}_0^m)$.

2. Initialise the measurement rate estimate λ_0^m using an initial estimate for α_0^m and β_0^m for each component m.

3. For time scans $t = 1, \ldots, T$,

   (a) Prediction Step: For each target component, compute the

   - predicted state estimates $\boldsymbol{x}_{t|t-1}^m$ by applying the dynamics model to the previous posterior state estimates,
   - mixing terms $\lambda_{t|t-1}^m$ by evaluating (5.52) and (5.54).

   (b) Initialise the probability that a shot due to a target falls in pixel i, $h^i(\cdot)$, using the predicted state estimate $\boldsymbol{x}_{t|t-1}^m$.

   (c) Select the number of iterations I^C so it is large enough to guarantee convergence.

   (d) For $j = 1, \ldots, I^C$,

   - Update Step: For each target component, compute
     - the updated per-pixel probability $h^i(\cdot)$ using the previous iterations state estimate $\boldsymbol{x}_{t-1|t-1}^{m,j-1}$ if it is available.
     - the KF updated state estimates $\boldsymbol{x}_{t|t}^{m,j}$ for the current iteration using the synthetic measurements $\breve{\boldsymbol{z}}_{t,m}^{(IP),j}$ via (6.39) and synthetic covariance $\tilde{\mathsf{R}}_{t,m}^{(IP),j}$ via (6.37).
     - mixing terms $\lambda_{t|t}^{m,j}$ by evaluating (5.52) and (5.55).

$\widetilde{Q}_X^{j+1}$ and $\widetilde{Q}_X^{j}$ are different functions as they depend on the measurement rates λ, which are updated at each iteration. Therefore there is no guarantee that (6.41) will be true.

Consider an alternative condition for convergence in which the modified auxiliary function evaluated at the current maximising state is now required to be greater than the value of the same auxiliary function evaluated at the previous iteration's maximising state:

$$\widetilde{Q}_X^{j+1}(\hat{\boldsymbol{x}}^{j+1}) \geq \widetilde{Q}_X^{j+1}(\hat{\boldsymbol{x}}^{j}). \tag{6.42}$$

This alternative criteria also implies that the modified auxiliary function is non-decreasing at each maximisation step. To demonstrate this through simulations, define ϵ_Q as follows:

$$\epsilon_Q = \widetilde{Q}_X^{j+1}(\hat{\boldsymbol{x}}^{j+1}) - \widetilde{Q}_X^{j+1}(\hat{\boldsymbol{x}}^{j}). \tag{6.43}$$

To evaluate ϵ_Q, let $\widetilde{Q}_{t,X}^{j+1}(\boldsymbol{x})$ denote the modified auxiliary function calculated at the $(j+1)^{th}$ iteration at time t evaluated at state $\boldsymbol{x}$. Also let $\tilde{\boldsymbol{z}}_t^{j+1}$ and $\tilde{R}_t^{j+1}$ denote the synthetic measurement and covariance evaluated at time t for the $(j+1)^{th}$ iteration. For a linear Gaussian scenario, it has the following form:

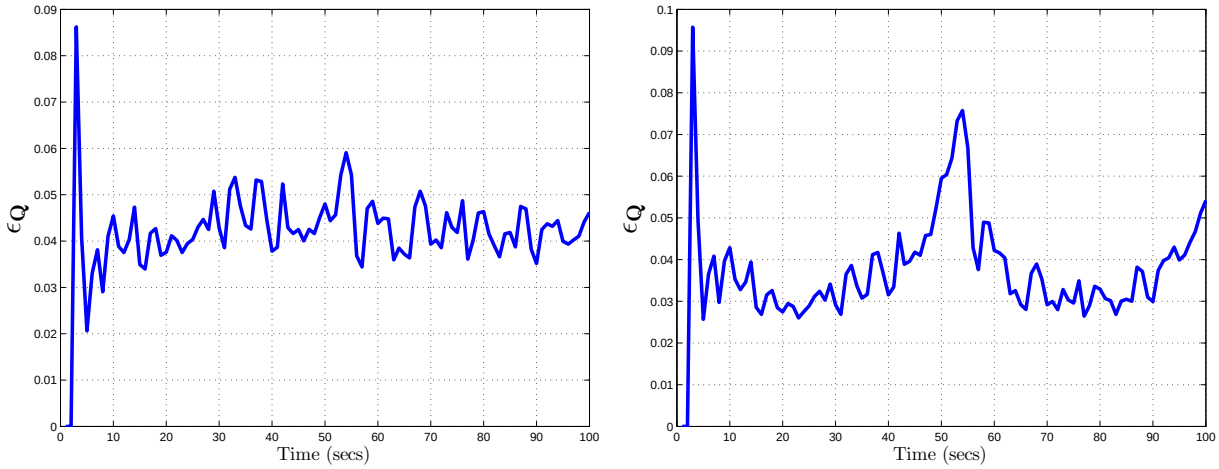
$$\widetilde{Q}_{t,X}^{j+1}(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - F\hat{\boldsymbol{x}}_{t-1})^T P_{t|t-1}^{-1}(\boldsymbol{x} - F\hat{\boldsymbol{x}}_{t-1}) - \frac{1}{2}(\tilde{\boldsymbol{z}}_t^{j+1} - H\boldsymbol{x})^T (\tilde{R}_t^{j+1})^{-1}(\tilde{\boldsymbol{z}}_t^{j+1} - H\boldsymbol{x}), \tag{6.44}$$

where $P_{t|t-1}^{-1}$ is the predicted covariance. Ignoring the negative scalar terms in front of each quadratic term, it can be seen that $\widetilde{Q}_{t,X}^{j+1}(\boldsymbol{x})$ simplifies to the sum of the two quadratic functions, which is positive. This implies that the $\widetilde{Q}_{t,X}^{j+1}(\boldsymbol{x})$ is always negative and is bounded above by zero.

Figure 6.2 shows ϵ_Q averaged over 100 Monte Carlo runs and 50 maximisation steps. For the estimation of the measurement rates, a forgetting factor of $\eta = 10$ was assumed. We verified during the simulation that ϵ_Q is always positive for all EM iterations and target amplitude models. As $\widetilde{Q}_X^{j+1}$ is bounded above by zero for linear Gaussian scenarios and simulations demonstrate that the lower bound (6.34) is always non-decreasing at each maximisation step and every time frame, this implies that for the given scenarios, the IP-PMHT lower bound always converges.

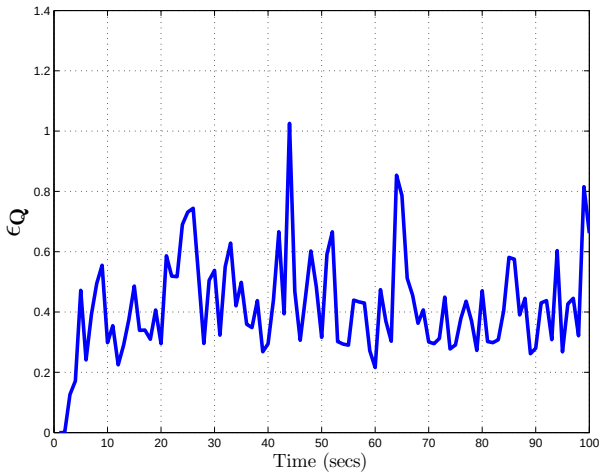
The performance of the IP-PMHT is now compared with the standard H-PMHT and the Poisson H-PMHT derived in Chapter 5. The root mean square (RMS) state estimation errors and target peak SNR estimates are computed for each algorithm for each target amplitude model.

Figure 6.3 show the RMS state estimation errors averaged over all Monte Carlo runs for the standard H-PMHT (solid blue line), Poisson H-PMHT (solid red line) and IP-PMHT (dashed black line) assuming a forgetting factor of $\eta = 10$ for the Poisson-based algorithms. Observe that the IP-PMHT gives slightly worse RMS performance for all target amplitude models and seems

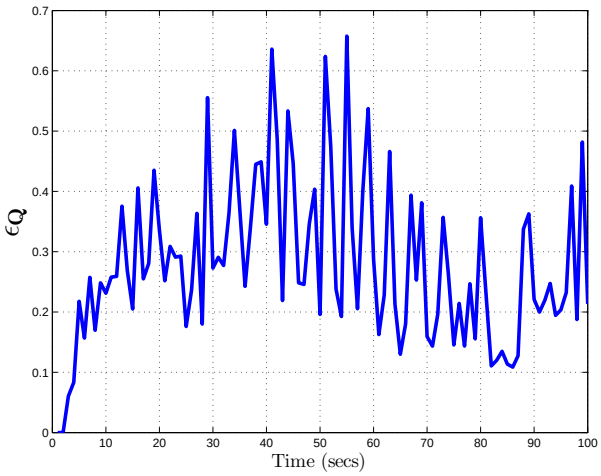


(a) Constant Amplitude Target Model

(b) Slowly Varying Amplitude Model



(c) Swerling I Model



(d) Step Function with Gaussian noise Scenario

Figure 6.2: ϵ_Q versus time averaged over 100 Monte Carlo and 50 iterations assuming a forgetting factor of $\eta = 10$.

to diverge at the end of the Swerling I model scenario. This is likely to be attributed to the IP-PMHT maximising a different lower bound than the standard H-PMHT and Poisson H-PMHT algorithms. Unsurprisingly, the IP-PMHT SNR estimates are identical to the Poisson H-PMHT estimates in Figure 6.4, as both algorithms employ the same dynamic estimation procedure for the mixing terms. This also verifies that the state and SNR estimates from the IP-PMHT converge to values that align well with the standard and Poisson H-PMHT algorithms.

If we consider the performance of the IP-PMHT SNR estimates for a single run, assuming a forgetting factor of $\eta = 1, 3$ and 10, we see that in Figures 6.5-6.8, the IP-PMHT SNR estimates again match the Poisson H-PMHT estimates. Like the Poisson H-PMHT, the IP-PMHT gives similar performance to the standard H-PMHT for scenarios featuring constant or relatively smooth changes in target amplitude. For scenarios in which the target amplitude has instantaneous fluctuations, the IP-PMHT outperforms the standard H-PMHT. The IP-PMHT is less sensitive to noisy fluctuations and is capable of giving more smoothed estimates of the true target SNR through the forgetting factor term η.

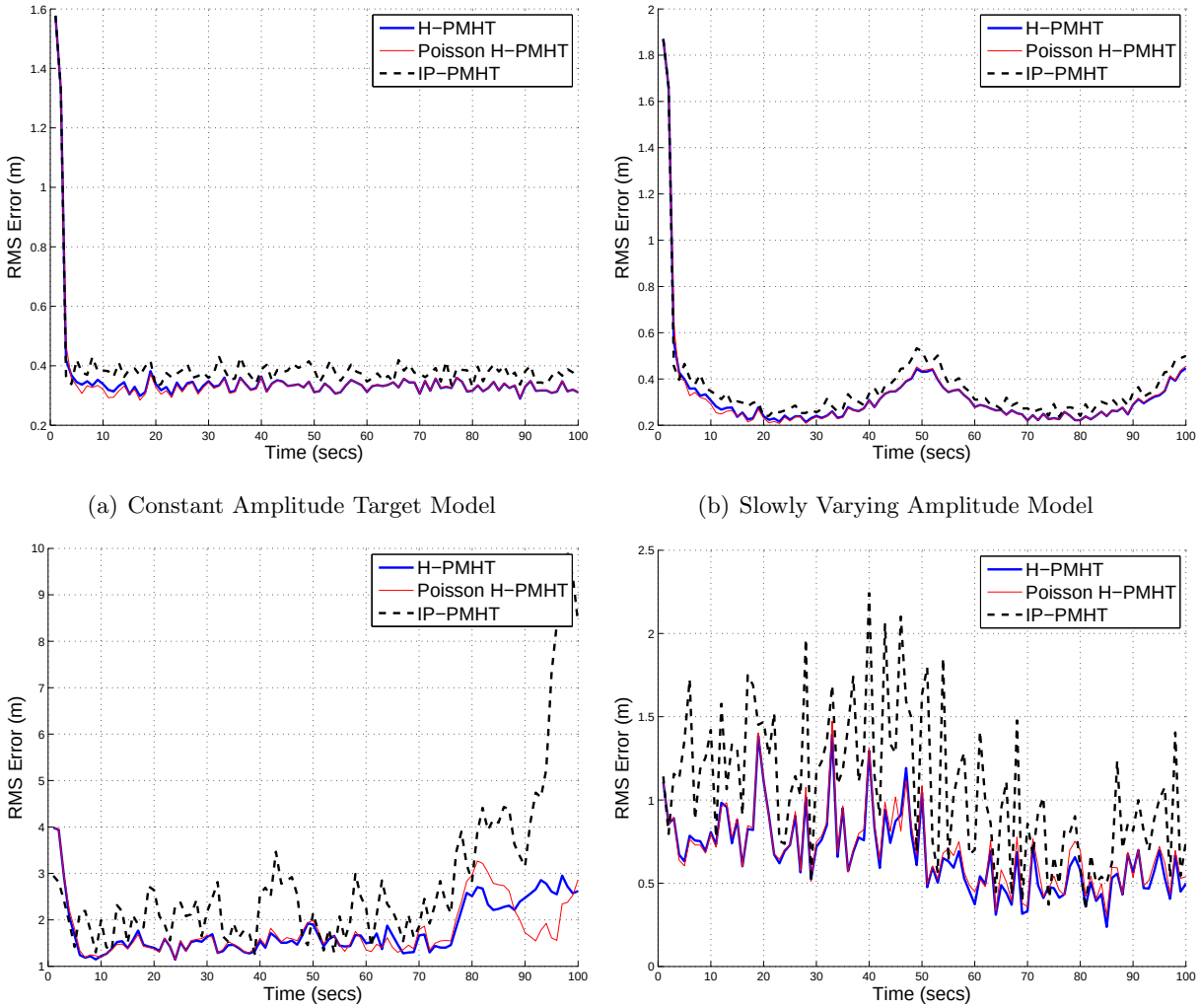
Figure 6.9 shows the track SNR variance versus forgetting factor η. We see that like the Poisson H-PMHT, the IP-PMHT outperforms the standard H-PMHT in all four amplitude models.

Figure 6.10 on page 163 shows the track SNR variance for different SNR values for the Swerling I amplitude model. Again, the IP-PMHT performance is identical to the Poisson H-PMHT and outperforms the standard H-PMHT. In both simulations, the IP-PMHT averages over a larger time frame to give significantly smaller variance estimates when compared with the standard H-PMHT.

6.6 Summary

This chapter has presented a new TkBD algorithm called the Interpolated Poisson PMHT (IP-PMHT) that is based on the application of PMHT data association to continuous valued data. This new algorithm models the measurement image directly using an interpolated form of the Poisson distribution. Like the Poisson measurement model discussed in Chapter 5, this new model also allows for a randomly evolving mean target amplitude estimate.

Although the interpolated version of the Poisson distribution can only be approximated by a continuous pdf on the non-negative real line for $\lambda > 4$, we nevertheless make use of convenient properties of probability measures to derive a TkBD algorithm that is similar in principle to the standard H-PMHT but removes the requirement for quantisation and thus the generation of an artificial histogram. The key contribution of this chapter is the first derivation of a TkBD algorithm assuming an interpolated Poisson distribution on the energy from an individual target



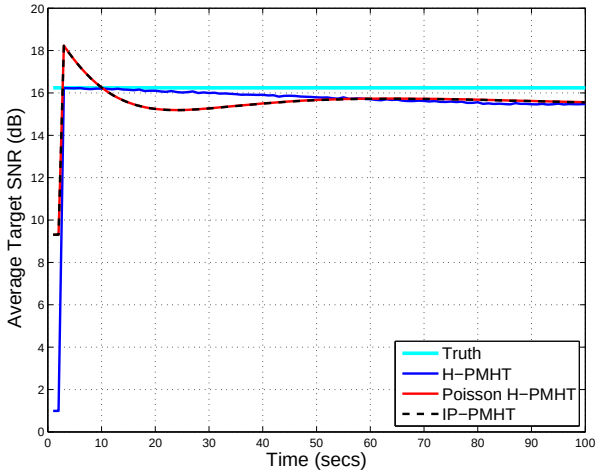
(a) Constant Amplitude Target Model

(b) Slowly Varying Amplitude Model

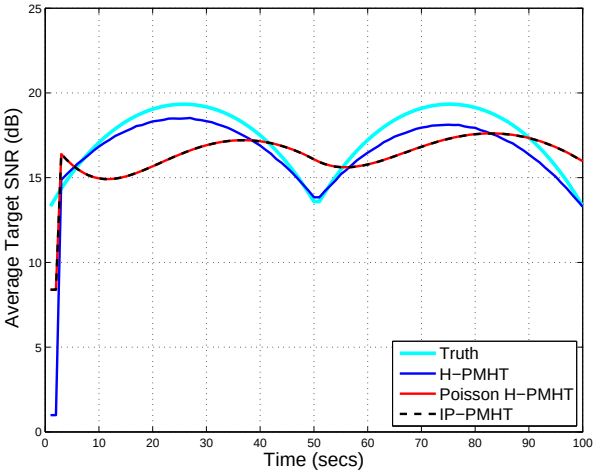
(c) Swerling I Model

(d) Step Function with Gaussian noise Scenario

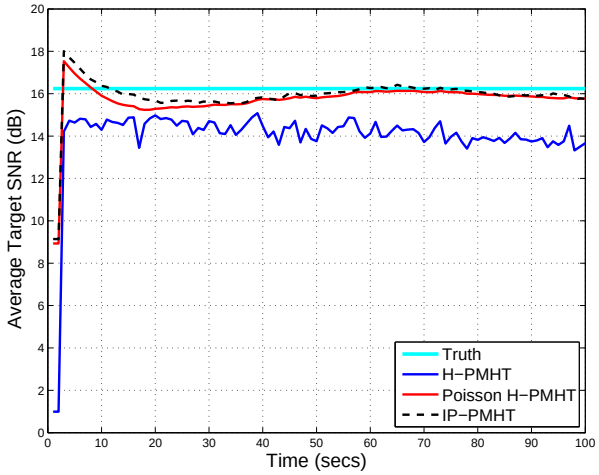
Figure 6.3: Comparison of RMS error versus time (averaged over 100 Monte Carlo runs) for the standard H-PMHT, Poisson H-PMHT and IP-PMHT for various target amplitude models assuming $\eta = 10$.



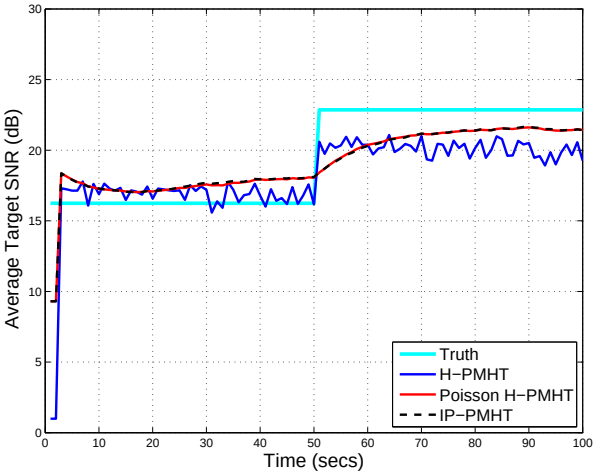
(a) Constant Amplitude Target Model



(b) Slowly Varying Amplitude Model

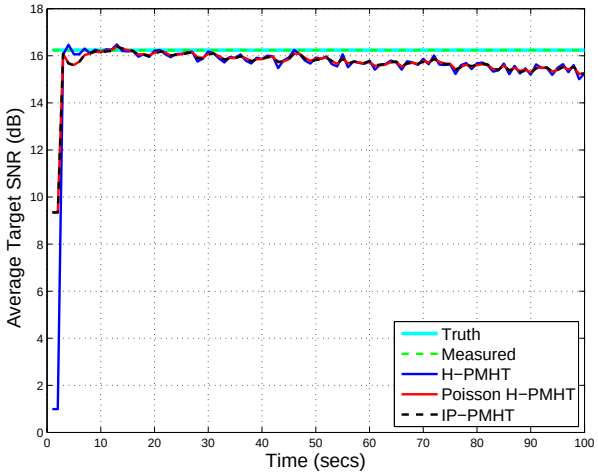


(c) Swerling I Model

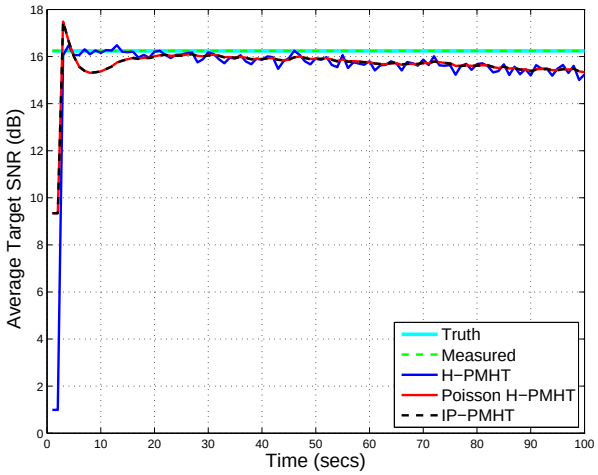


(d) Step Function with Gaussian noise Scenario

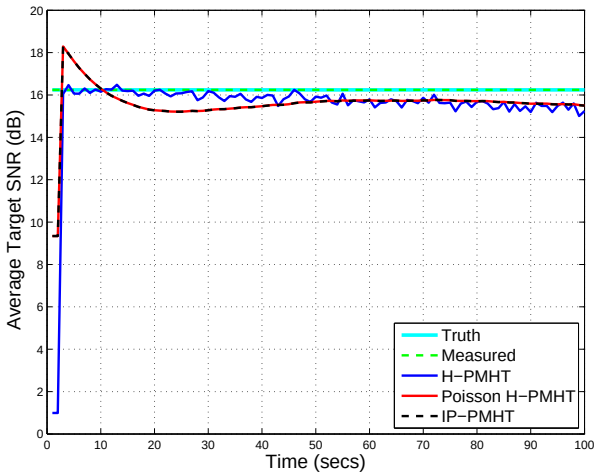
Figure 6.4: Comparison of track SNR versus time (averaged over 100 Monte Carlo runs) for the standard H-PMHT, Poisson H-PMHT and IP-PMHT for various target amplitude models assuming $\eta = 10$.



(a) $\eta = 1$

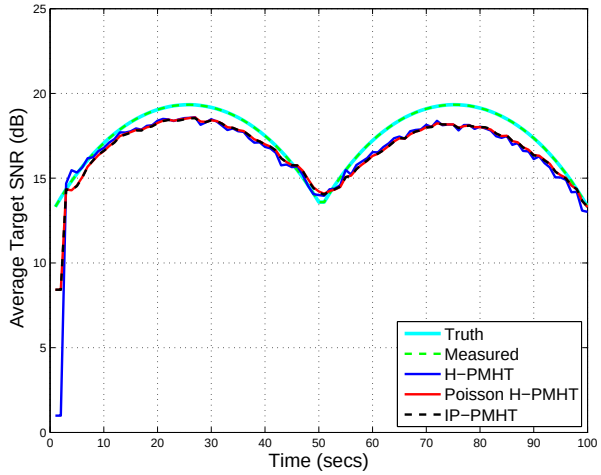


(b) $\eta = 3$

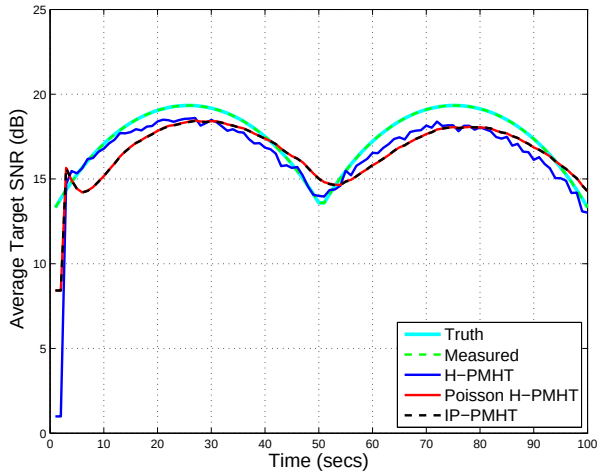


(c) $\eta = 10$

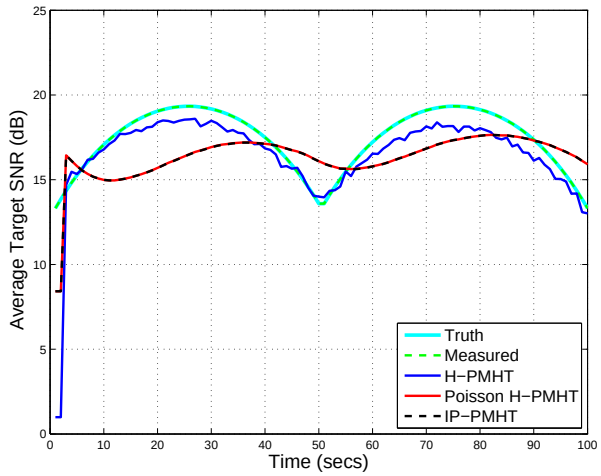
Figure 6.5: **Constant Amplitude scenario:** Comparison of the average target SNR for the standard H-PMHT, Poisson H-PMHT and IP-PMHT for varying forgetting factor η.



(a) $\eta = 1$

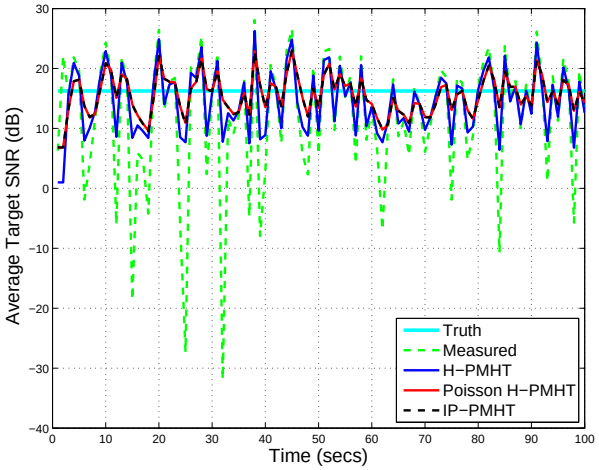


(b) $\eta = 3$

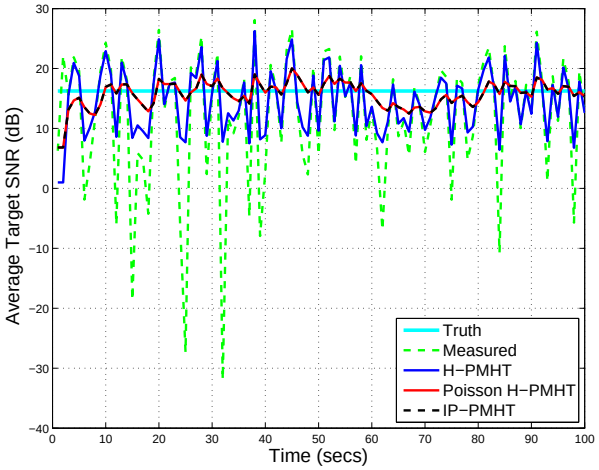


(c) $\eta = 10$

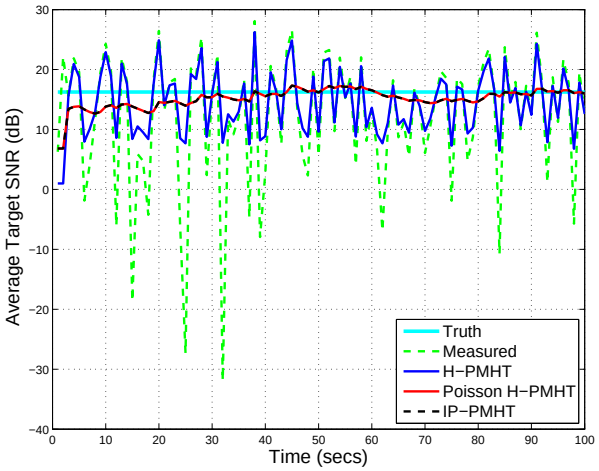
Figure 6.6: **Slowly Varying Amplitude Scenario:** Comparison of the average target SNR for the standard H-PMHT, Poisson H-PMHT and IP-PMHT for varying forgetting factor η.



(a) $\eta = 1$

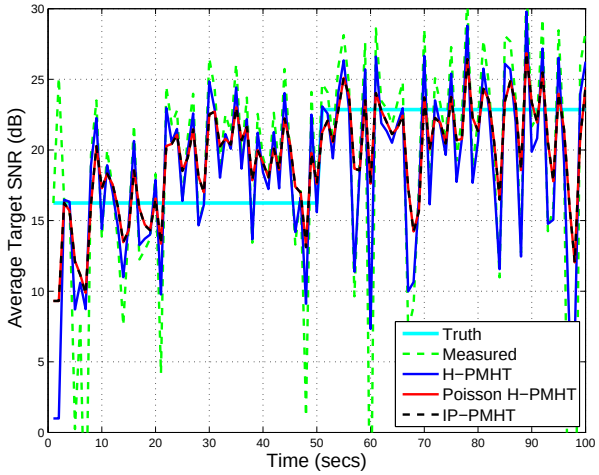


(b) $\eta = 3$

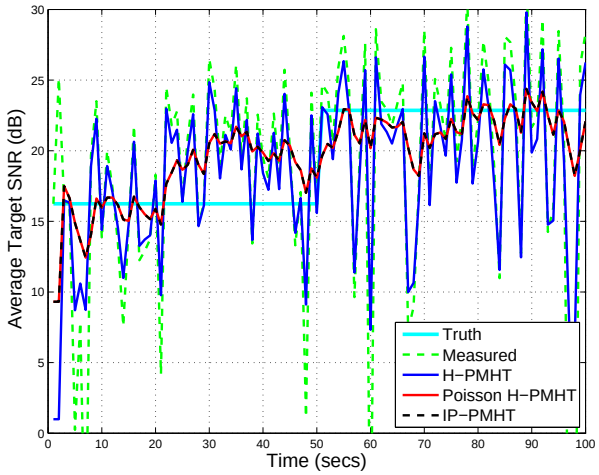


(c) $\eta = 10$

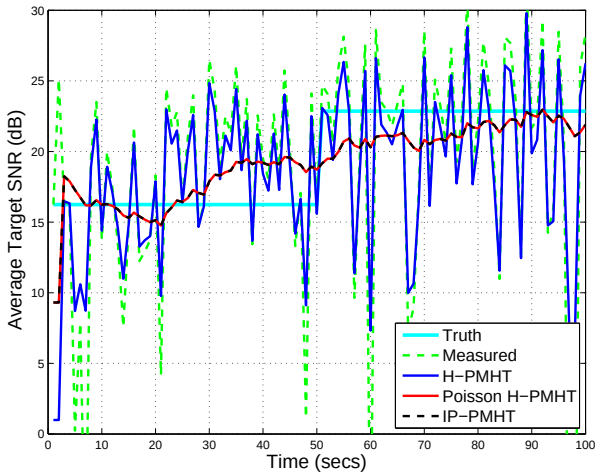
Figure 6.7: **Swerling Model I Scenario:** Comparison of the average target SNR for the standard H-PMHT, Poisson H-PMHT and IP-PMHT for varying forgetting factor η.



(a) $\eta = 1$

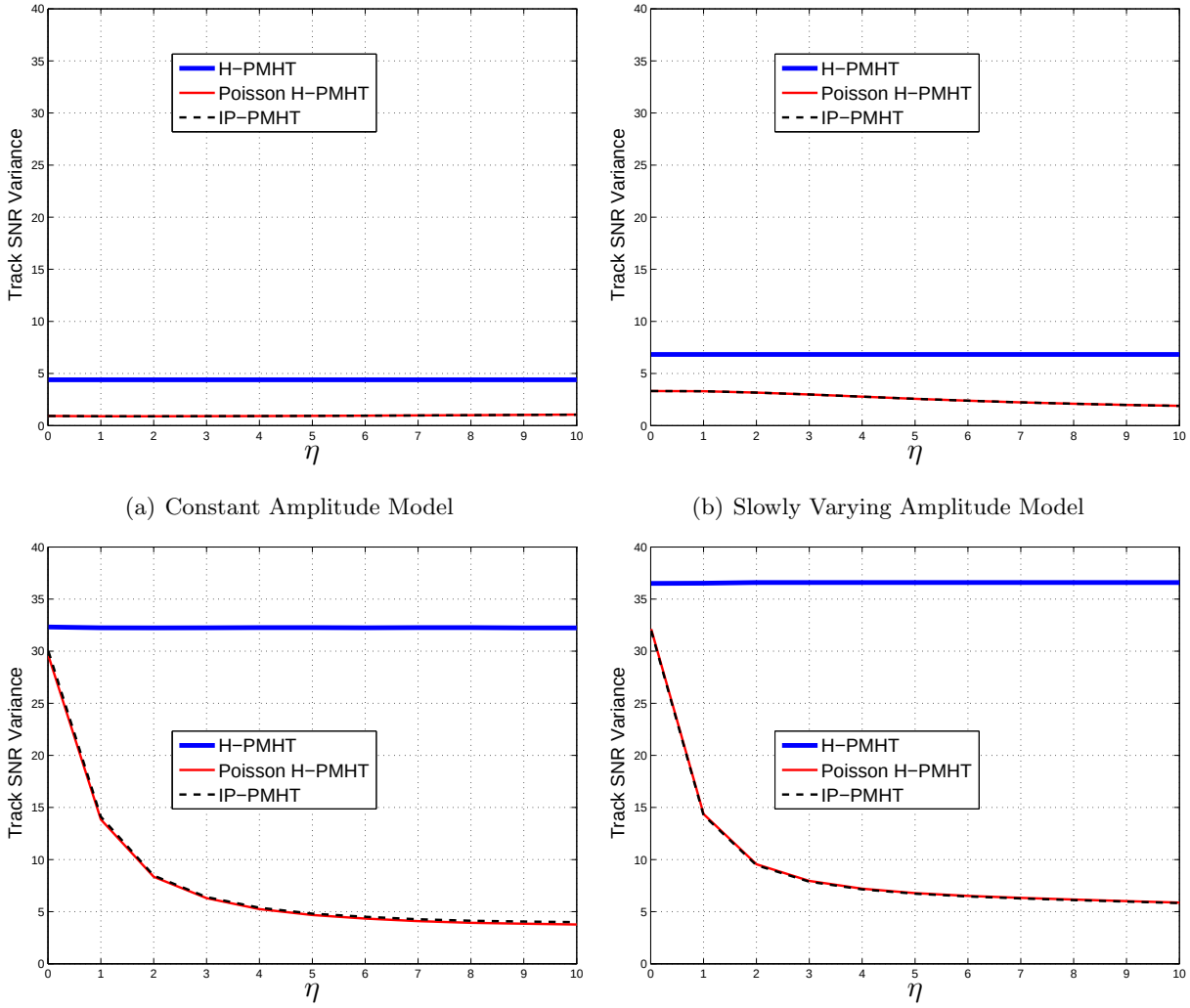


(b) $\eta = 3$



(c) $\eta = 10$

Figure 6.8: **Step Function with Gaussian noise Scenario:** Comparison of the average target SNR for the standard H-PMHT, Poisson H-PMHT and IP-PMHT for varying forgetting factor η.



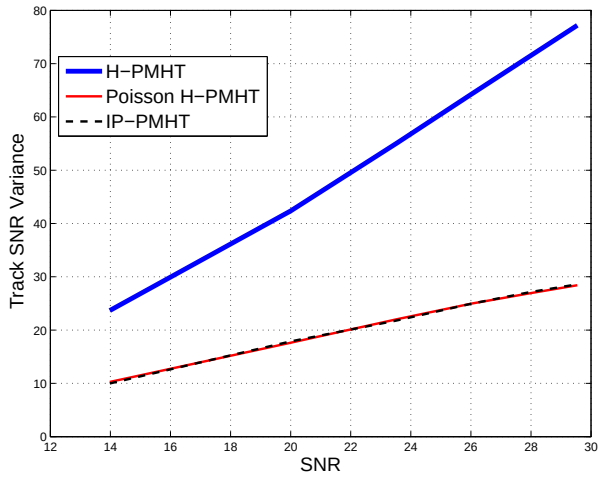
(a) Constant Amplitude Model

(b) Slowly Varying Amplitude Model

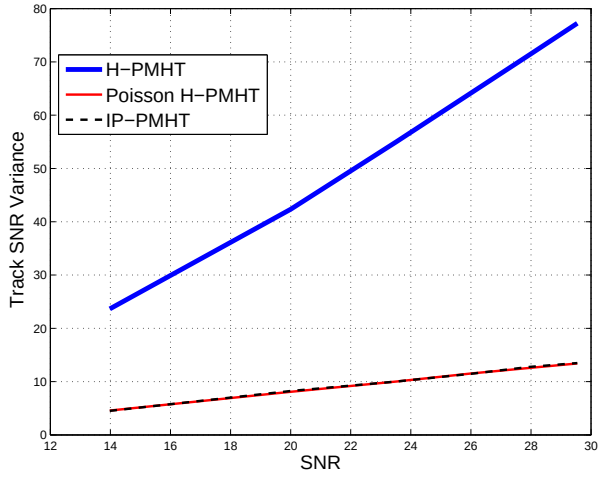
(c) Swerling I Model

(d) Step Function with Gaussian noise Scenario

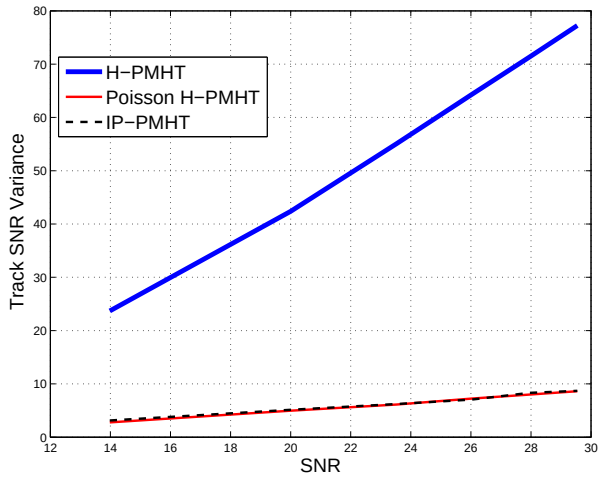
Figure 6.9: Track SNR variance versus forgetting factor η for the standard H-PMHT, Poisson H-PMHT and IP-PMHT under various target amplitude models.



(a) $\eta = 1$



(b) $\eta = 3$



(c) $\eta = 10$

Figure 6.10: Comparison of the average track SNR variance for the standard H-PMHT, Poisson H-PMHT and IP-PMHT for varying forgetting factor η for the Swerling Model I Scenario.

and the first KF implementation of this algorithm for linear Gaussian scenarios.

Through simulated linear Gaussian simulations, we showed that the state estimation performance of the IP-PMHT is similar in performance to the standard H-PMHT and Poisson H-PMHT. We also verified that the component mixing terms in the IP-PMHT are identical to the mixing terms derived under the Poisson H-PMHT after quantisation. Thus, the IP-PMHT algorithm has the benefits of a smoothed mixing term estimate without requiring an intermediate quantisation step.

By avoiding measurement quantisation, we are able to partially address some of the limitations of the H-PMHT, namely its questionable independence assumptions and quantisation issues that result in inconsistent track performance whenever the image size is changed. However, it can be argued that these assumptions have been replaced by questionable distributional assumptions as the derivation of the IP-PMHT depends on the validity of superposition and pdf approximations.

Chapter 7

Comparative Study using Trial Data from an Active Towed Array Sonar

The detection and tracking of targets using an active towed array sonar system is a non-trivial problem due to the complex nature of the underwater acoustic environment. Performance can be degraded by high levels of acoustic clutter, fluctuating target returns and a relatively low sonar data update rate. As a result, conventional point measurement trackers often struggle to form target tracks or can be overwhelmed by the high false alarm rate. This chapter considers the application of several TkBD algorithms based on PMHT data association to trial data from an active towed array sonar system. Through a comparative study, we investigate the benefits of using TkBD over conventional point measurement tracking for the the active sonar problem.

The key contribution of this chapter is a comparative study analysing the performance of a conventional point measurement tracker with the standard H-PMHT, the Poisson H-PMHT and IP-PMHT presented in Chapters 4, 5 and 6, respectively, using trial data from an active towed array sonar system. The TkBD algorithms are modified for the active sonar problem by allowing changes in target appearance with received array bearing to be included into the measurement model. The main contributions of this chapter are summarised in article [137].

The following study assumes that the target dynamics is modelled by a Markov process, however it is possible to assume an alternate target model such as the HRP discussed in Chapter 3. As the underwater targets considered in this study are generally assumed to travel without a predetermined path, the application of HRP target models is not considered here.

The chapter is arranged as follows: Section 7.1 introduces the active sonar tracking problem; Section 7.2 outlines the tracking problem for both the conventional and TkBD case; Section 7.3 gives a brief review of the Integrated Probabilistic Data Association (IPDA) approach to

conventional tracking and discusses how the TkBD measurement model can be modified to incorporate a bearing-dependent point spread function (psf); Section 7.4 outlines the details of the comparative study; we first verify that the inclusion of a bearing dependent psf in the standard H-PMHT results in improved tracking; next, the bearing dependent psf is integrated into the Poisson H-PMHT and IP-PMHT, and their performance is compared with the modified standard H-PMHT and a conventional point-measurement tracker based on IPDA; Section 7.5 summarises and concludes the chapter.

7.1 Introduction

Traditionally, a sonar system's detection and tracking capabilities have been considered separate functions. Conventional active sonar processing systems use beamforming and matched filter correlation with a replica of the transmitted pulse to generate a sensor image of the reflected acoustic intensity as a function of range and bearing. Typically, this sensor image is normalised to remove mean background variations and a fixed threshold is applied to produce detections that are then provided to the tracker. The role of the tracker is to associate point-measurements from a common target across time and return estimates of the target's trajectory. This approach is often sufficient for detecting and tracking high Signal-to-Noise Ratio (SNR) targets but becomes more challenging for low SNR targets, as the process of reducing the sensor image to thresholded detections discards valuable target information. TkBD has been proposed as a natural solution for tracking low SNR targets as the declaration of a target detection can be delayed until after a series of frames have been processed. This technique has the potential to provide significant gains in scenarios with low SNR targets and high clutter [39].

The first applications of TkBD to active sonar were based on dynamic programming techniques, which use a fixed grid to model the propagation of target states with time [16,46,125]. However, most of these techniques have been demonstrated with simulated data and the application of TkBD to at-sea data has been limited [97, 113]. The application of other alternative TkBD algorithms to active sonar data in the open literature has also been limited. Moreover, previous applications of TkBD have failed to address issues that are unique to active sonar such as the low update rate.

The key contribution of the chapter is a *study comparing the performance of several TkBD algorithms based on PMHT data association with a conventional point measurement tracker based on IPDA [27, 94] in two representative acoustic environments using trial data against an Echo-Repeater (ER) target from an active towed array sonar system.* An ER is an acoustic source that can simulate the returns from a simple point-like target in the ocean environment. The

TkBD algorithms considered are the standard H-PMHT, the Poisson H-PMHT and IP-PMHT presented in Chapter 4, 5 and 6, respectively. To model the variation in the target appearance with sonar receive bearing, the TkBD algorithms are extended to include a bearing-dependent psf.

7.2 Active Sonar Problem

In the active sonar tracking problem, the main objective is to identify the number of targets and estimate their trajectories over time using a sequence of noisy measurements. Define the state vector $\boldsymbol{x}_t^m$, which evolves with time $t \in \mathbb{N}$, where $\mathbb{N}$ is the set of all natural numbers, $m = 1, \ldots, M$ denotes the target index in the case of a multi-target scenario, and M is the total number of targets.

In active sonar tracking, the target state is generally modelled in a Cartesian frame. For conventional point measurement tracking, it is sufficient to describe the target state using position and velocity in two-dimensions. Define x_t^m and $\dot{x}_t^m$ to be the target position and velocity in the x-direction, respectively. Similarly, define y_t^m and $\dot{y}_t^m$ to be the respective target position and velocity in the y-direction. The target state for point measurement tracking is defined as:

$$\boldsymbol{x}_t^m = \begin{bmatrix} x_t^m & \dot{x}_t^m & y_t^m & \dot{y}_t^m \end{bmatrix}^T. \tag{7.1}$$

In the TkBD case, it is common to supplement the state vector with the target amplitude,

$$\boldsymbol{x}_t^m = \begin{bmatrix} x_t^m & \dot{x}_t^m & y_t^m & \dot{y}_t^m & A_t^m \end{bmatrix}^T, \tag{7.2}$$

where A_t^m denotes the amplitude for target m at time t. It is assumed that the state vector contains all relevant information about the system.

Generally, when analysing a dynamic system, two models are required: the target and measurement models. In practice, it is common to assume stochastic models for the system target dynamics and sensor data.

Target Model: The target model describes the target state evolution with time and can be expressed in terms of a linear discrete-time stochastic model

$$\boldsymbol{x}_t^m = \mathsf{F}_{t-1}\boldsymbol{x}_{t-1}^m + \boldsymbol{v}_{t-1}, \tag{7.3}$$

where F_{t-1} is a known matrix describing the linear state transition from $\boldsymbol{x}_{t-1}^m$ to $\boldsymbol{x}_t^m$, and $\boldsymbol{v}_{t-1}$ is an independent identically distributed (iid) system noise sequence representing uncertainties in the target motion. For the active sonar tracking problem, the target model needs to capture

the dynamics of an underwater target. It is assumed that a nearly-constant-velocity model is sufficient.

Measurement Model: The measurement model relates the noisy measurements to the state $\boldsymbol{x}_t^m$. In conventional point measurement sonar tracking, measurements are generally received in polar coordinates (range and bearing). Let $\mathbf{Z}_t = \{\boldsymbol{z}_t^j\}$ for $j \in \{1, \ldots, \mathbb{m}_t\}$ denote the set of point measurements received at time t. Note that since the detection probability of each target is, in general, less than unity, there is no guarantee that every target will produce a point measurement at each time. Furthermore, some of these detections may originate from clutter.

Suppose target m is associated with measurement j_m at time t. Then,

$$\boldsymbol{z}_t^{j_m} = \zeta_t(\boldsymbol{x}_t^m) + \boldsymbol{\psi}_t, \tag{7.4}$$

where $\zeta_t(\boldsymbol{x}_t^m)$ denotes the measurement function that maps the state into the measurement space and $\boldsymbol{\psi}_t$ is an iid measurement noise sequence. Clutter detections are assumed to be uniformly distributed across the surveillance region and the total number of clutter detections at each time is assumed to follow a Poisson distribution.

In the TkBD case, the measurement model now relates the images $\mathbf{Z}_t$ in range and bearing to the target state $\boldsymbol{x}_t$. Let $\boldsymbol{z}_t^i$ denote the i^{th} pixel in the measurement image at time t, and let $\mathbf{Z}_t = \{\boldsymbol{z}_t^i\}$ for $i = 1, \ldots, I$ represent a stacked vector of all the pixels in the image, and I is the total number of pixels in the measurement image. For ease of presentation, we have used a stacked vector to represent the image to allow single index referencing. A two dimensional representation could just as easily have been used. We assume a point-scatterer target, such that the target contribution to the measurement image can be described purely in terms of the psf, $h(\boldsymbol{x}_t^m)$. The TkBD measurement image is then described as follows,

$$\mathbf{Z}_t = \sum_{m=1}^{M} A_t^m h(\boldsymbol{x}_t^m) + \boldsymbol{w}_t, \tag{7.5}$$

where $\boldsymbol{w}_t$ is an IID noise sequence for the measurement image. Note that the psf is a property of the sensor and is the same for all targets, but can vary with different sensors.

The recursive Bayesian approach to dynamic state estimation requires the construction of a posterior probability density function (pdf) $p(\boldsymbol{x}_t^m|\mathsf{Z}_t)$ of the state based on all available information, including information gained from noisy measurements up until time t, $\mathsf{Z}_t = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_t\}$. The pdf is assumed to contain all available information of the state and is considered to be a complete solution to the estimation problem. The recursive Bayesian filter used to update the pdf given new information consists of two stages: the prediction and the update stage. In the prediction

stage, the state is propagated forward in time using the target dynamics model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. In the update state, the information contained in the new measurement is used to modify the pdf. For more details, refer to subsection 2.2.1.

The trial data uses Doppler insensitive waveforms, resulting in a measurement process that only observes the position component of the target. Thus the measurement model assumes that the likelihood is independent of the target velocity component.

In the TkBD case, the output of the sensor generally consists of a 2D image and the likelihood of seeing the sequence of images given the current state of the target. Assuming independent pixel noise, the likelihood for the image $\mathbf{Z}_t$ can be factorised as follows,

$$p(\mathbf{Z}_t|\boldsymbol{x}_t^m) = \prod_{i=1}^{I} p(\boldsymbol{z}_t^i|\boldsymbol{x}_t^m). \tag{7.6}$$

Under linear Gaussian assumptions, the optimal finite dimensional solution to the discrete-time recursive Bayesian state estimation problem is the Kalman Filter (KF). However, as the measurements in active sonar tracking are generally polar (range and bearing), and the target state is more naturally modelled in a Cartesian frame, both the conventional and TkBD measurement model defined by (7.4) and (7.5), respectively, are non-linear functions of the target state. In this case, approximations or suboptimal solutions must be considered to accommodate the non-linear measurement model. In this chapter, we consider a conventional tracking solution based on IPDA using converted measurements, and several TkBD solutions based on PMHT data association.

7.3 Tracking Algorithms

In this section we outline the tracking algorithms that will be implemented for the comparative study. In subsection 7.3.1, we present a conventional point measurement tracker based on IPDA. We then outline how the TkBD measurement model can be modified to include a bearing dependent psf.

7.3.1 Conventional Tracking using Integrated Probabilistic Data Association

One of the main difficulties in point-measurement tracking is determining which measurements arise due to a particular target and which measurements are the result of false alarms or due to objects that are not of interest. This problem is referred to as data association. Under a particular data association hypothesis, the target state can be estimated using a KF or non-linear counterpart. Probabilistic Data Association (PDA) expresses the target state pdf as a

weighted sum of target state pdfs over data association hypotheses formed for that hypothesis and provides expressions to determine the probabilities of these hypotheses [27]. The resulting pdf is a Gaussian mixture, which is then approximated by a single Gaussian. The Integrated PDA (IPDA) extends the target state-space by defining a binary existence variable E_t that indicates whether or not there is actually a target present and assumes that this variable evolves according to a Markov chain. The IPDA provides equations for recursively updating the target states and the probability of target existence based on the PDA approach for data association [94]. The probability of existence can then be used to automate track management.

The advantages of IPDA are that it is computationally inexpensive and can be implemented as a modified KF: the target description consists only of a mean vector, a covariance matrix and a scalar existence probability. However, the algorithm uses a single Gaussian component to approximate the Gaussian mixture arising from the weighted sum of Gaussian target state pdfs formed over data association hypotheses. This approximation can be poor, especially when the mixture has more than one dominant component. IPDA also assumes the existence of at most one target.

Let $E_t = 1$ and $E_t = 0$ denote the probability of target existence and non-existence, respectively. The IPDA optimal track update evaluates the joint probability of the target state $\boldsymbol{x}_t$ and the event of target existence E_t conditioned on the set of received measurements $\mathbf{Z}_t$ at time scan t:

$$p(\boldsymbol{x}_t, E_t|\mathbf{Z}_t) = p(\boldsymbol{x}_t|E_t, \mathbf{Z}_t)p(E_t|\mathbf{Z}_t). \tag{7.7}$$

Note that the component index on the target state has been suppressed for notational convenience. In this chapter, the IPDA algorithm is based on an implementation discussed in [94].

The non-linearity of the point measurement model is dealt with by converting the polar measurements to a Cartesian frame and using the converted measurements as inputs into the KF [13]. Tracks are initiated when the estimated probability of track existence rises above a certain threshold. Likewise, track termination occurs when the probability of track existence falls below another threshold. In this chapter, a single target IPDA tracker based on [94] is implemented. However, a multi-target track management logic is imposed based on the assumption that targets in the image do not overlap or interact with each other. In a multi-target scenario, this allows an independent IPDA filter to be run for each track. Joint Integrated Probabilistic Data Association (JIPDA) can be used for scenarios featuring interacting targets but it is more computational expensive than IPDA [93] and is not considered here.

7.3.2 Track-Before-Detect using Expectation-Maximisation Data Association

For the comparative study, we consider three different TkBD algorithms based on the application of EM data association:

- The standard H-PMHT algorithm introduced in Chapter 4
- The Poisson H-PMHT proposed in Chapter 5
- The IP-PMHT proposed in Chapter 6

The TkBD algorithms can be naturally extended to track in a multi-target scenario but still have linear complexity with the number of targets. In the study, the maximisation step of the EM procedure for all three algorithms is performed using an Extended Kalman Filter (EKF). However, the linearisation point for the EKF is modified with each Expectation-Maximisation (EM) iteration so the result is similar to the Iterated EKF, which is known to provide a more accurate estimate than the standard EKF [6].

Next, we will describe how the standard TkBD measurement model can be modified to include a bearing-dependent psf to model the variation in the beampattens for an active towed array sonar.

Due to left-right ambiguity issues characteristic of linear array systems, when a target is detected by a towed array, it will appear as two identical targets symmetrically placed on either side of the towed array in the sensor image. An own-ship manoeuvre is required to identify a real target from the ambiguous one. Figure 7.1 shows representative beam patterns and the effect of left-right ambiguity issues for an active towed array sonar system at the following receive directions:

- Broadside: defined as 90 degrees from front of the ship. The beam pattern consists of a narrow beam and the two peaks generated by left-right ambiguity are well-separated,
- Near aft of the ship: where the beams are wider and the left-right ambiguous peaks begin to overlap, and
- Aft of the ship: defined as close to 180 degrees from the front of the ship, where the left-right beam patterns merge into a single wide peak.

It can be seen that as the receive direction moves from broadside to aft of the ship, the spread in the beam pattern increases and the ambiguous target in the image merges with the real

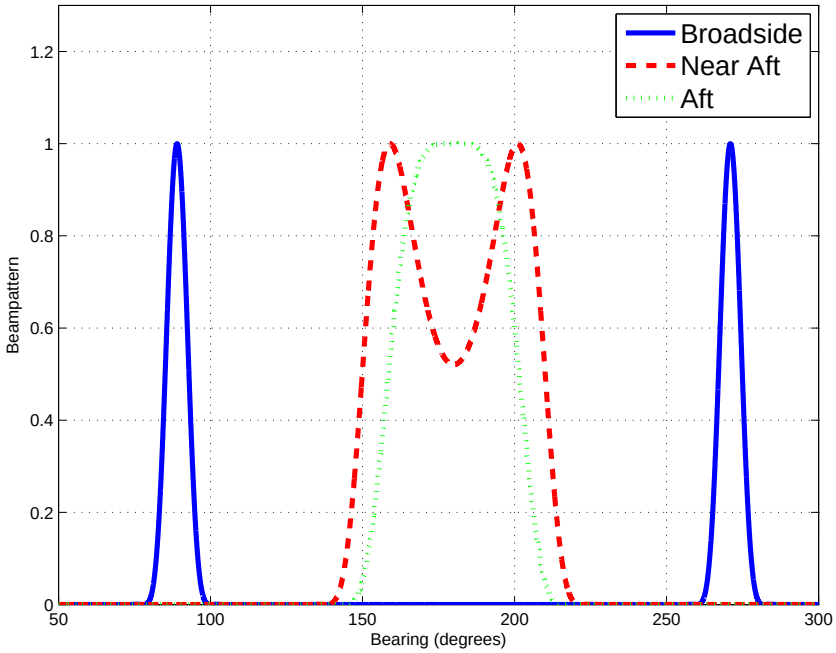


Figure 7.1: Beampattern vs. bearing for transmissions at broadside, near aft and aft of the ship.

target to form a single target smeared across multiple bearing bins in the sensor image. The half-height beamwidth for the beampattern is approximately 8 degrees at broadside, 50 degrees (across both peaks) in the near aft direction and 42 degrees at aft.

The variation in the beampattern with receive direction will be modelled by assuming a bearing-dependent psf. Recall that a sensor's psf can be used to describe the appearance of the target in the sensor image. For the sonar problem, the sensor outputs a measurement in range r and bearing θ space such that,

$$\begin{aligned} h(\boldsymbol{x}_t^m) &= h(r,\theta) \\ &= h_r(r)h_\theta(\theta), \end{aligned} \tag{7.8}$$

where $h_r(r)$ and $h_\theta(\theta)$ are defined as the psfs for range and bearing space respectively, and assumed to be independent of each other. Both psfs can be approximated using Gaussians. For an active towed array sonar system, the psf function $h_r(r)$ can be assumed to be consistent across all bearings, however the psf in the bearing space $h_\theta(\theta)$ will be dependent on the receive direction due to the variation of beam patterns with bearing.

In this study, a bearing-dependent psf using a Gaussian approximation will be assumed. For a given receive direction, the Gaussian psf can be calculated by setting the half-height width of the Gaussian pdf to be equal to the half-height beamwidth of the current receive direction. The corresponding variance for the Gaussian psf can then be easily calculated. Figure 7.2 shows the

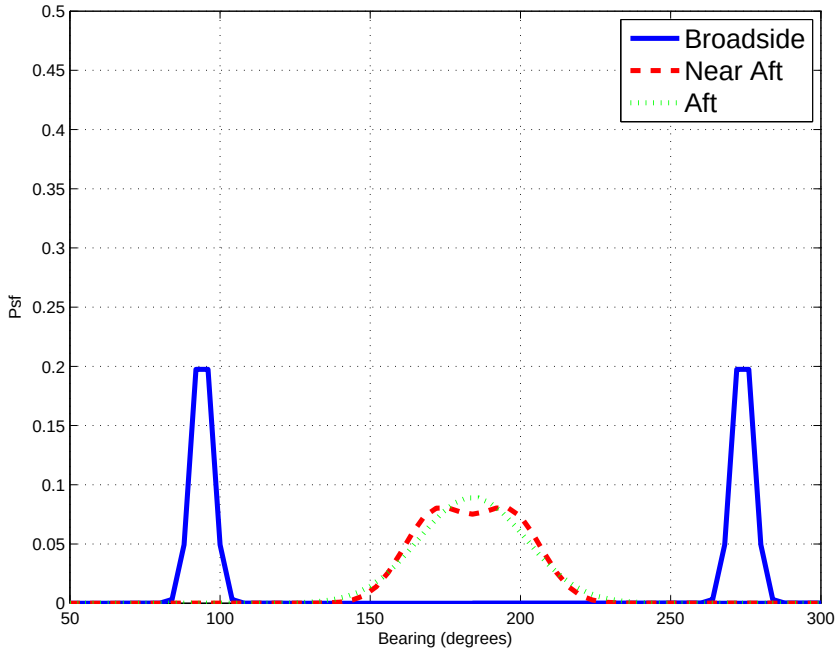


Figure 7.2: Variations in point spread function $h_\theta(\theta)$ in bearing space, for transmissions at broadside, near aft and aft of the ship.

resulting psfs corresponding to broadside, near aft and aft receive directions. Note that for the near aft direction, a sum of two Gaussians, one for each of the left-right beams is used to model the beam pattern.

Tracks are initiated based on a peak detection thresholding process at a given SNR level and the well known M/N logic is imposed such that if M out of N point detections are associated with a tentative track, the track is upgraded from tentative to confirmed status [21]. The transition matrix for the target dynamics is given by a constant velocity model which varies depending on the time between consecutive transmissions.

7.4 Comparative Study using Sonar Trial Data

This section outlines the details for the comparative study using archived sonar data from a towed array sonar system. This section is separated into two different studies; in Section 7.4.1 we verify that the inclusion of a bearing dependent psf in the standard H-PMHT results in improved tracking; then in Section 7.4.2 the bearing dependent psf is integrated into the Poisson H-PMHT and IP-PMHT, and their performance is compared with the modified standard H-PMHT and a conventional point-measurement tracker based on IPDA.

In the second study, both the TkBD and IPDA performance is quantified for a set of SNR

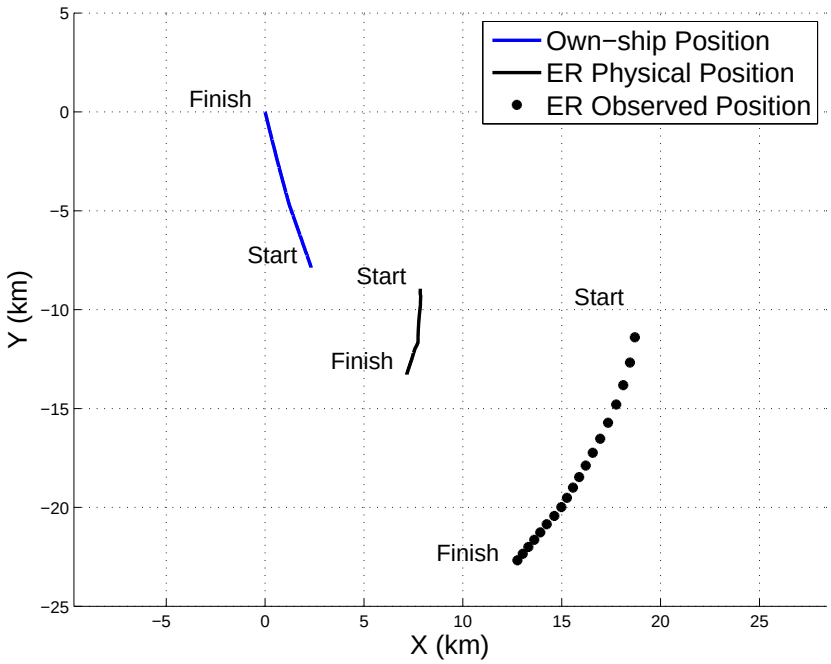
thresholds. For the IPDA, the SNR threshold will determine the number of detections that are passed to the tracker. In the case of the TkBD algorithms, the SNR threshold will be used in a peak detector for track initiation.

The inclusion of a bearing dependent psf into the TkBD measurement model and the comparison of various TkBD algorithms with the IPDA using trial sonar data are the key contributions of this chapter.

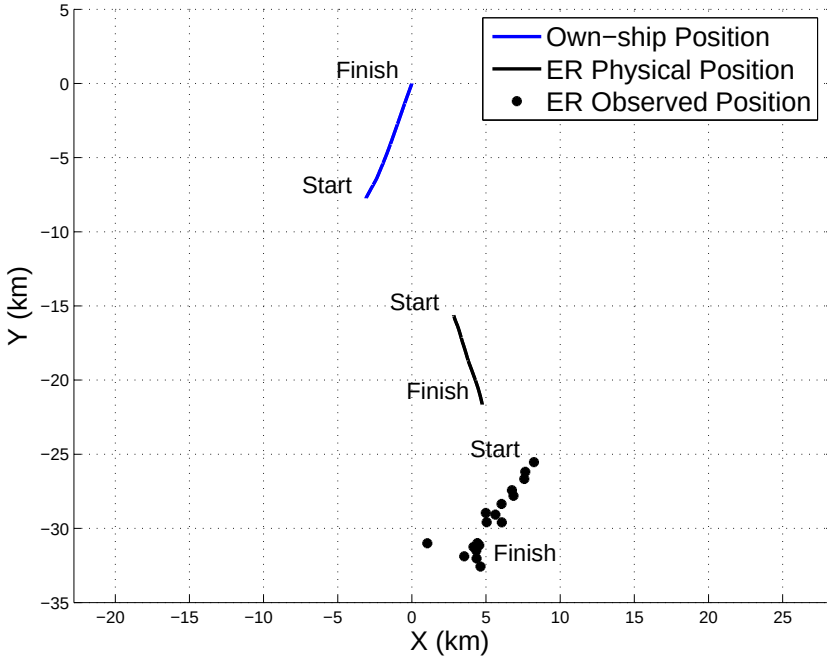
The data used in this study originates from a sonar trial conducted by the Defence Science and Technology Organisation (DSTO) from May to August 2003 using a containerised active towed array demonstration sonar system called CASSTASS. The sonar trial featured a line array towed behind a moving surface ship at two different locations in the Western Australian eXercise Area (WAXA). The two datasets feature characteristics that are unique to the sonar detection and tracking problem in a shallow environment with water depths from 150-250m and for an intermediate ocean environment with depths from 800-1400m. Transmissions were set to detect possible targets with a maximum range of 60 km with the majority of transmissions being in the aft direction. The datasets feature a fluctuating target and persistent clutter detections that are the result of reflections from bathymetric features along the continental shelf. Both datasets consist of approximately 20 transmissions with the time duration between transmissions being approximately 90 seconds.

The target is an ER made to simulate the returns from a simple point-like target in the ocean environment. During the trial, the ER platform was observed to have an average travelling speed of 0-4 knots with varying target strength of 9, 19 and 29 dB. The measured SNR for the two datasets is approximately 25 dB, which is relatively high, and it is expected that both the IPDA and TkBD will be able to form tracks on the ER. However, variations in performance are also expected as the 'target' SNR did fluctuate with time, with SNR levels as low as 13 dB being observed.

Both the IPDA and TkBD algorithms perform tracking using global Cartesian coordinates with an origin fixed at the last recorded own-ship position in each trial. Figure 7.3 shows the physical positions for the own-ship (blue line) and ER (black solid line) for each dataset. A characteristic of the ER is that it features a time delayed response. The returns from the ER have a constant time delayed response for each transmission sent out by the towed array source. As a result, the ER observed position will appear offset by some constant distance corresponding to the difference in time between when the ER receives the source's signal and the time the ER takes to respond with a simulated return. In Figure 7.3, the observed measurements from the ER (with offset) are plotted as black dots. The ER observed measurements will be used as input into the IPDA algorithm.



(a) Shallow



(b) Intermediate

Figure 7.3: Own-ship, ER and Observed ER Position for each dataset

The IPDA algorithm uses point-measurement detections which were generated from an automated detection scheme featuring clustering. To gain an idea of the sensitivity of conventional tracking to thresholding, the IPDA algorithm was implemented using three detection thresholding levels at $11, 13$ and 15 dB. The expected number of clutter measurements was modelled using a Poisson distribution with parameter $\lambda = \frac{\hat{N}_k}{V_k}$ where V_k denotes the area of surveillance and $\hat{N}_k$ is estimated using the average number of point detections which varied with the SNR thresholding level. The IPDA algorithm initiated tentative tracks using two-point differencing. When the probability of existence for a track rose above 0.5, tentative tacks were upgraded to confirmed status. Tracks were terminated when the probability of existence fell below a threshold of 0.3.

For the TkBD case, the magnitude of the sensor returns in terms of power was collected in bearing and range cells consisting of 181 beam bins at 2 degree intervals from 0 to 360 degrees and approximately a few thousand range bins, with range intervals of approximately 60m.

In order to reduce the number of potential false tracks, the measurements images that were fed to the TkBD algorithms were truncated so that when the left-right ambiguous beams were well separated, only the correct left or right beams were used. The measurement images were also truncated to only a subset of ranges that consisted of approximately $\pm$5km from the starting true target position. For a fair comparison, the IPDA point detections were also limited to only detections that fell within the limits of the truncated TkBD images.

Figure 7.4 and Figure 7.5 show examples of the truncated TkBD measurement images across range and bearing space for target SNR values of 24 dB and 13 dB respectively. The true target position is indicated by the red circle. Clearly, when the target SNR is high, the target return is easily distinguishable from the background clutter. However when the target SNR drops, it is more difficult to detect the target from the background clutter. It is also important to realise that as the resolution of the bearing bins is finer than the beamwidth of the sensor, it is expected that the target returns will be spread over multiple bearing bins. For track initiation, the TkBD algorithms applied a peak detection threshold to the sensor image according to a certain SNR threshold and then used two-point differencing to initiate tentative tracks. In the case when the current position estimates of several tentative tracks were within 250 m in position, the highest SNR track was retained and all other tracks were discarded. A measure for the track quality was derived using target SNR calculated from the mixing proportions estimates. Tracks were confirmed and terminated using a 3/5 logic rule requiring this track quality measure to be above a threshold for 3 out of 5 returns. Due to the high volume of data, time-recursive TkBD filters were used in the analysis rather than processing the sequence as a batch. It was also found that a maximum of ten EM iterations was sufficient to ensure convergence at each

time scan.

Multi-target implementations were used to compare TkBD to conventional tracking. In both algorithms, it was assumed that background noise was uniform and measurement noise followed a Gaussian distribution.

The tracking outputs of the TkBD and the IPDA algorithms for the two sets of trial data is now presented. The tracks were compared with truth data provided by GPS logs.

7.4.1 Bearing Dependent Point Spread Function

In this first study, we evaluate the benefits of modifying the standard H-PMHT to include a bearing dependent psf as discussed in subsection 7.3.2. The algorithms considered in this subsection are:

- **H-PMHT**: Standard H-PMHT as presented in Chapter 4,
- **H-PMHT-BD**: The standard H-PMHT as presented in Chapter 4 modified to include a bearing dependent psf.

Figure 7.6 shows the the track outputs for both environments assuming a threshold of 11 dB for track initiation. It is important to realise that although the TkBD algorithms initiate tracks on thresholded peaks, state estimates are updated using the sensor image data. The true target position is indicated by the green squares and the true track outputs from the H-PMHT and H-PMHT-BD are shown as a solid red and as a solid blue line with circle markers, respectively.

In the shallow environment, both the H-PMHT and H-PMHT-BD are able to successfully track the true target and return no false tracks. However, the H-PMHT-BD estimate cuts the track short near the end of the scenario. In the intermediate environment, the H-PMHT also starts a track on the target but its estimate diverges at the end of the scenario. In contrast, the H-PMHT-BD is able to maintain a good track on the target.

Figure 7.7 shows the corresponding SNR estimates for each track in Figure 7.6. The truth shown in cyan corresponds to the target strength as selected in the ER settings. Recall that the ER is able to output varying target strength levels of 9, 19 and 29 dB, however the signal can be perturbed as it travels through the environment. The perturbed signal as measured by the towed array sonar is shown in dashed green. We can see that the measured ER SNR can vary greatly from the original target strength setting. As before, the SNR estimates from the H-PMHT are shown in red and the H-PMHT-BD in blue.

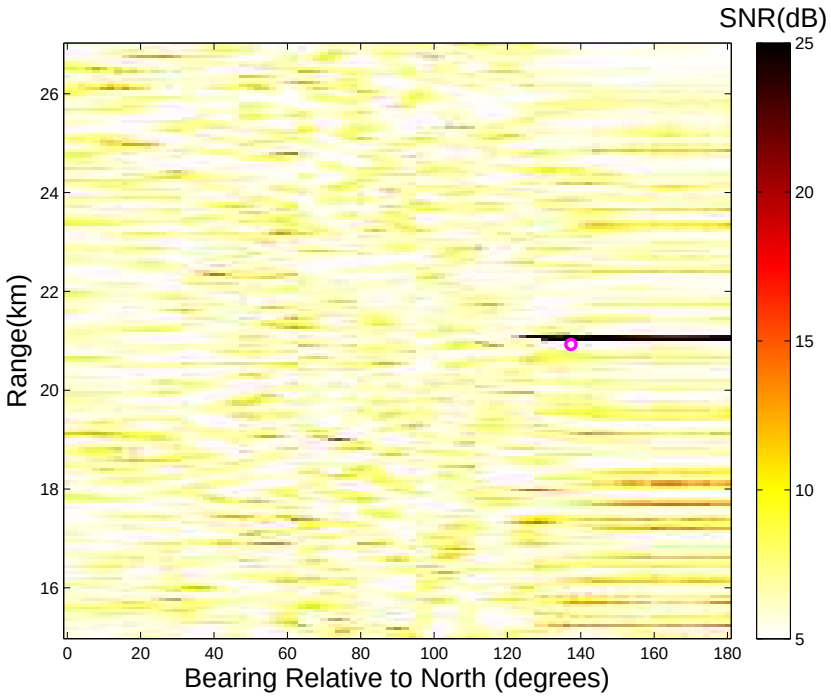


Figure 7.4: TkBD measurement image for a target SNR return value of 24 dB.

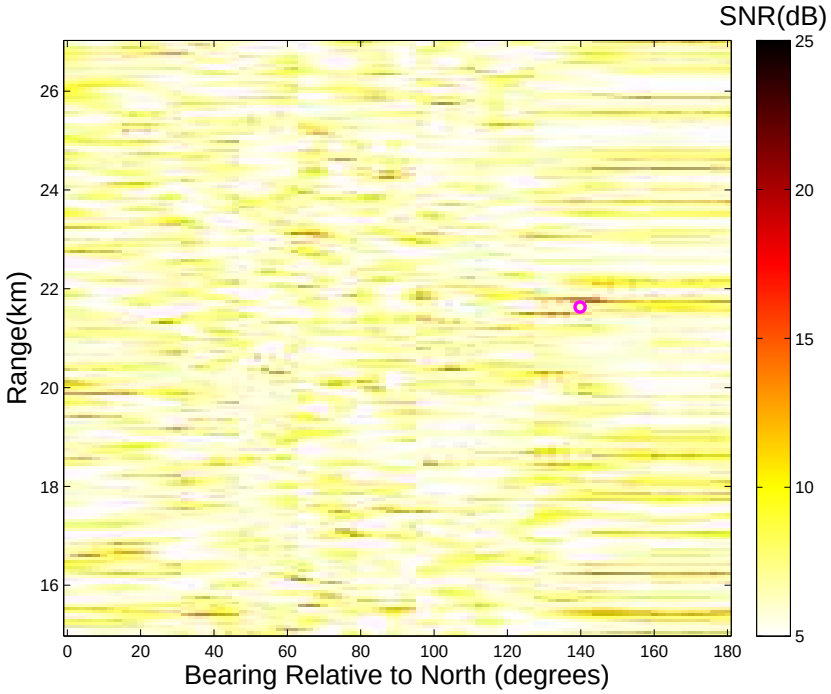
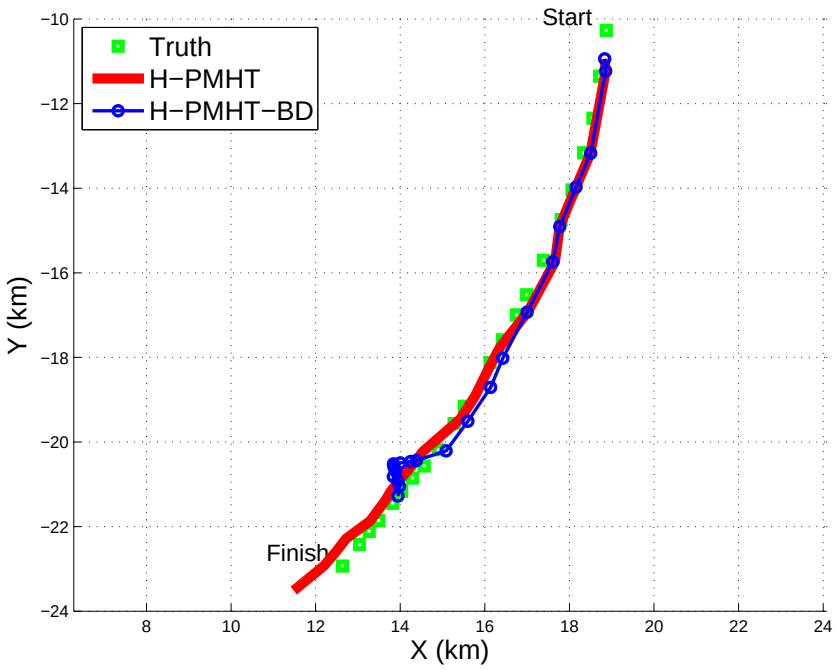
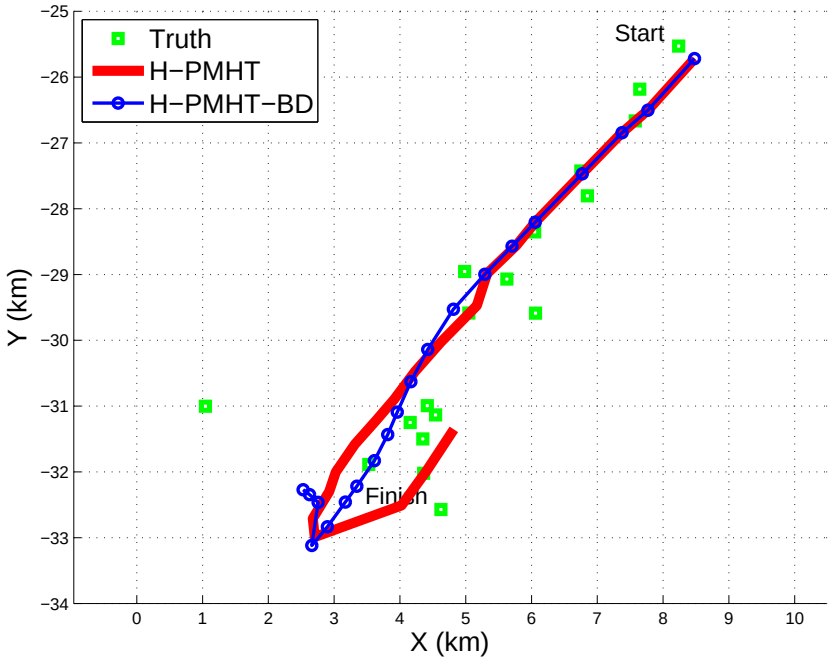


Figure 7.5: TkBD measurement image for a target SNR return value of 13 dB.

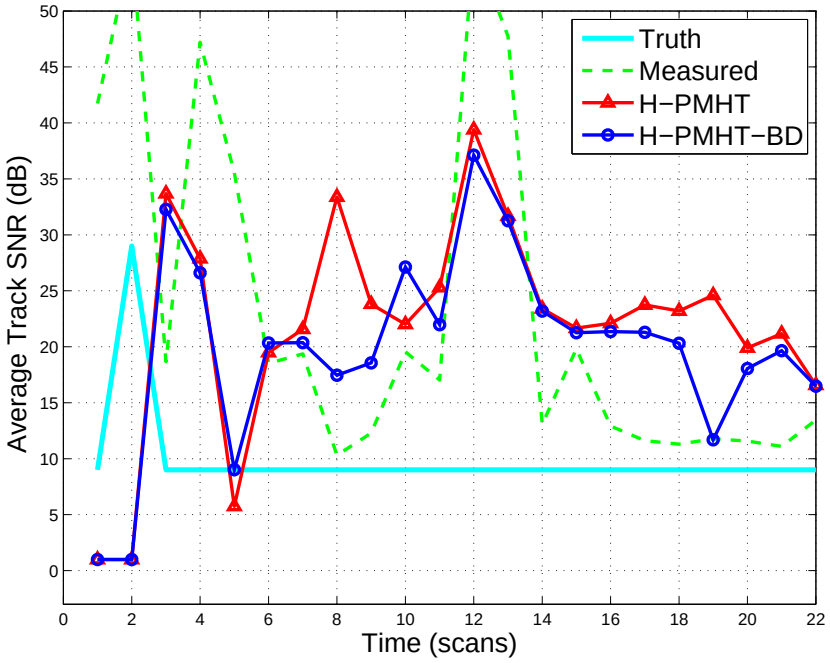


(a) Shallow

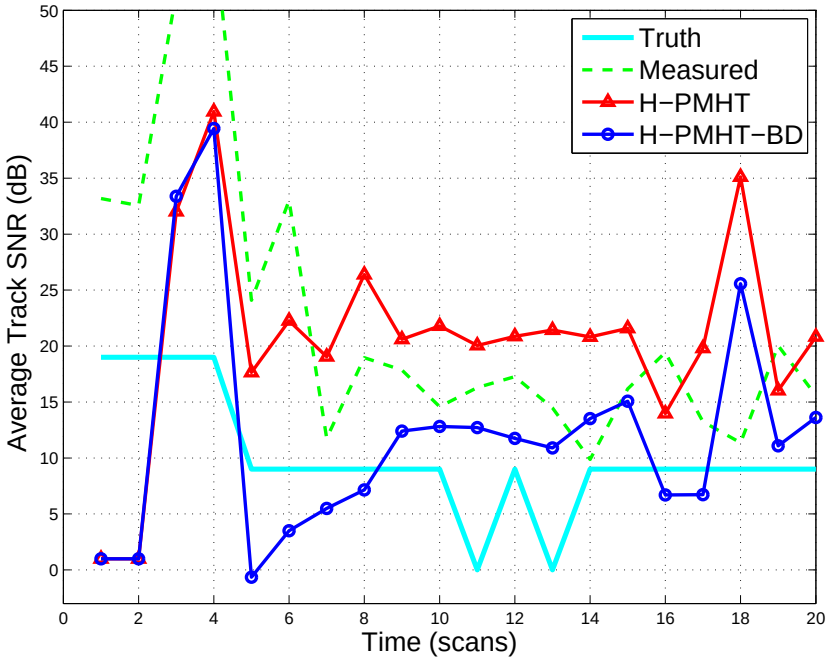


(b) Intermediate

Figure 7.6: Estimated target position: H-PMHT vs. the H-PMHT featuring a bearing dependent psf using SNR thresholding level of 11 dB.



(a) Shallow



(b) Intermediate

Figure 7.7: Average track SNR estimates: H-PMHT vs the H-PMHT featuring a bearing dependent psf using SNR thresholding level of 11 dB.

In the shallow environment, both algorithms gives similar estimates of the ER SNR, with the H-PMHT-BD giving slightly more smoothed estimates. In the intermediate environment, the H-PMHT-BD underestimates the measured SNR while the H-PMHT overestimates. However, the H-PMHT-BD seems to be more responsive to abrupt changes in SNR level and gives slightly better estimates in the second half of the scenario. Overall, the H-PMHT-BD seems to demonstrate slightly more robust performance than the H-PMHT.

7.4.2 Conventional Tracking versus Track-Before-Detect

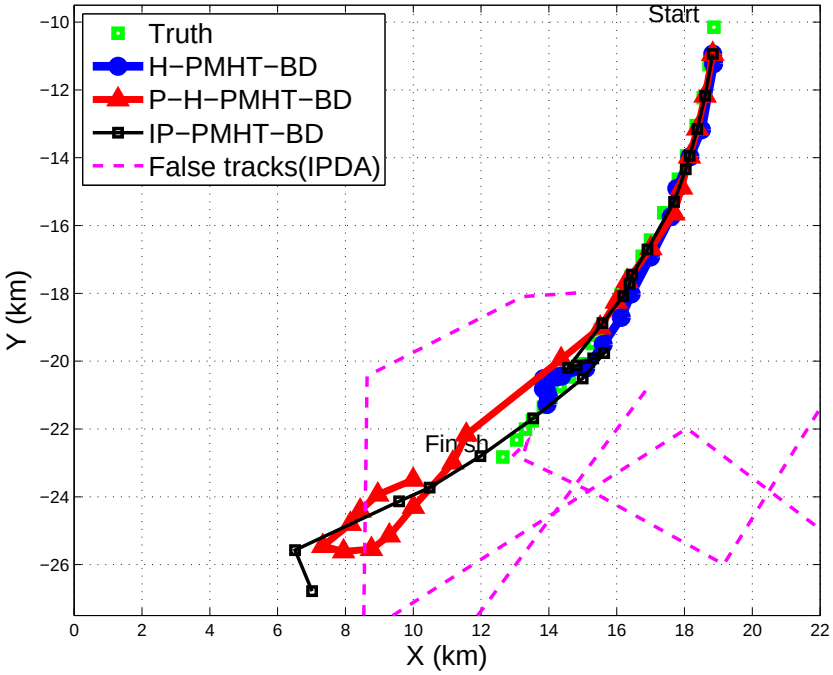
In the second study, we apply the TkBD algorithms discussed in previous chapters of the thesis to the trial sonar data. The performance of TkBD algorithms is compared with a conventional point measurement tracker based on IPDA. In the previous study, we observed that the H-PMHT-BD gave improved performance when compared with the standard H-PMHT. We now integrate the bearing dependent psf into all TkBD algorithms to ensure robust tracking performance. The algorithms considered in this study are:

- **H-PMHT-BD**:The standard H-PMHT as presented in Chapter 4 modified to include a bearing dependent psf.
- **P-H-PMHT-BD**: The Poisson H-PMHT as presented in Chapter 5 modified to include a bearing dependent psf.
- **IP-PMHT-BD**: The Interpolated Poisson H-PMHT as presented in Chapter 6 modified to include a bearing dependent psf.
- **IPDA**: The IPDA as discussed in subsection 7.3.1 [94].

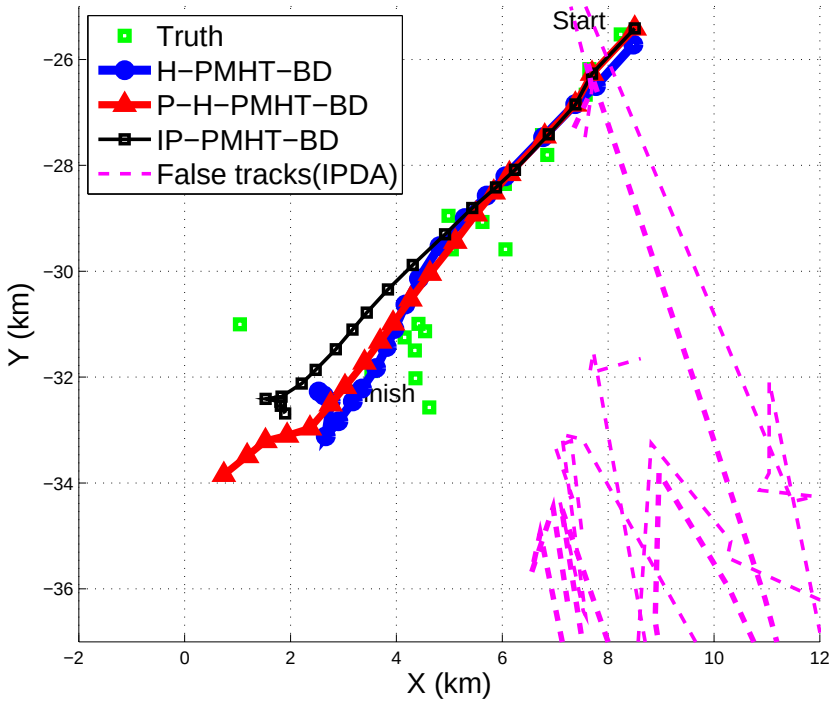
The P-H-PMHT-BD and IP-PMHT-BD algorithms assumed a forgetting factor $\eta = 200$.

Figures 7.8 - 7.10 show the tracking results for both environments at SNR thresholding levels of $11, 13$ and 15 dB, respectively. In these figures, the IPDA algorithm used detections thresholded at the respective SNR level while the TkBD algorithm used the same SNR level to initiate tracks using a peak detection thresholding scheme.

The TkBD algorithms were able to track the ER target successfully with zero false and divergent tracks in all environments and thresholding levels. The TkBD tracks also remained consistent across the different thresholding levels. For the shallow environment, we observed in the previous study that the H-PMHT-BD algorithms seems to terminate the track early. In contrast, both the P-H-PMHT-BD and IP-PMHT-BD tracks seem to overshoot the target endpoint. This is because the TkBD measurement covariance R can now potentially change at

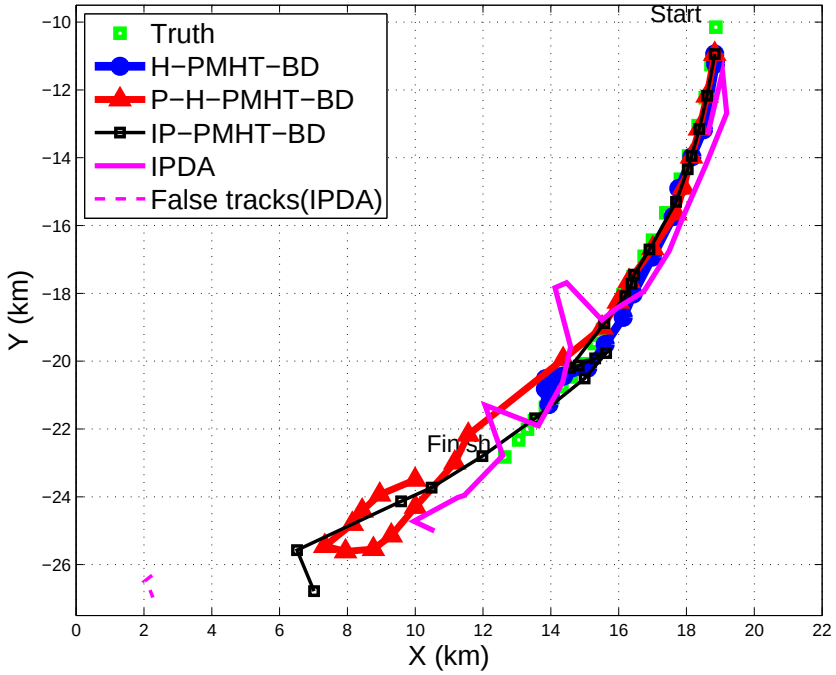


(a) Shallow

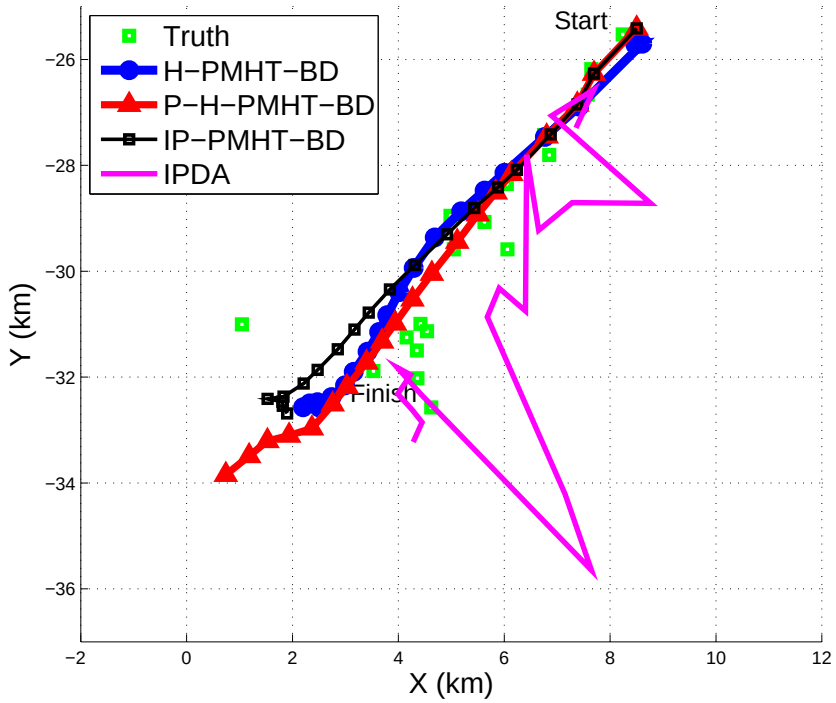


(b) Intermediate

Figure 7.8: Estimated target position: Various TkBD algorithms vs IPDA using a SNR thresholding level of 11 dB

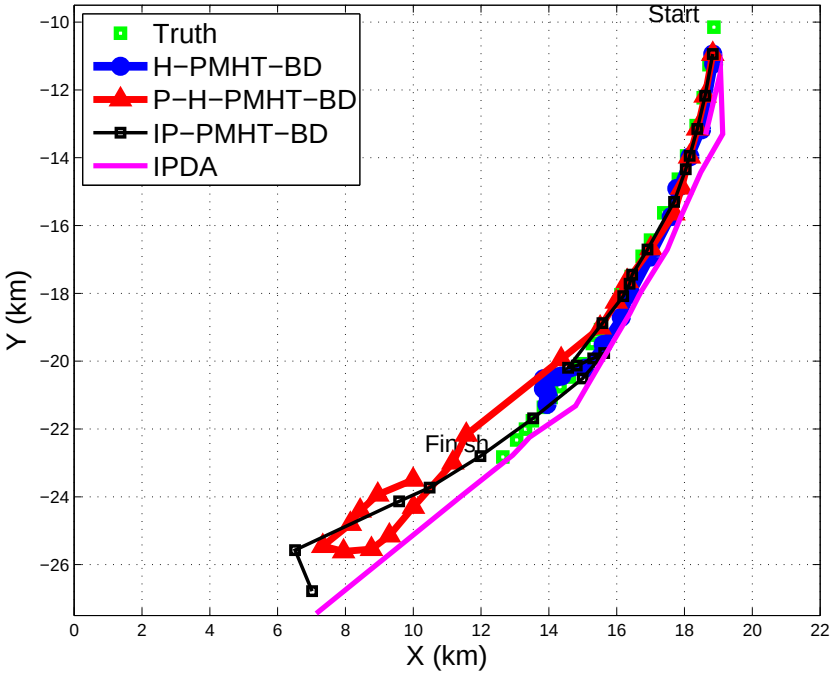


(a) Shallow

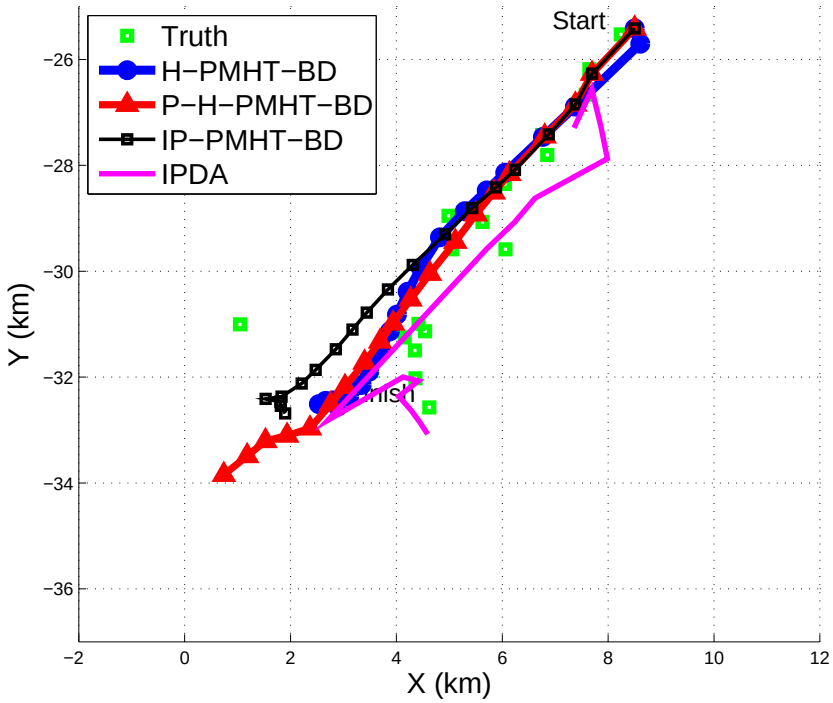


(b) Intermediate

Figure 7.9: Estimated target position: Various TkBD algorithms vs IPDA using a SNR thresholding level of 13 dB



(a) Shallow



(b) Intermediate

Figure 7.10: Estimated target position: Various TkBD algorithms vs IPDA using a SNR thresholding level of 15 dB

each time depending on the target estimate at that time, due to the inclusion of the bearing dependent psf. As each TkBD algorithm can produce a different target estimate at every time point, the measurement covariance can also potentially differ between each algorithm. This can either push the target estimates short or long as observed in Figure 7.8 (a).

The IPDA experienced more difficulty tracking at lower thresholding levels. When the SNR threshold level was set to 11 dB, the IPDA processed, on average, over 100 detections at each time frame. The IPDA algorithm was unable to form sensible tracks due to the large number of detections, which served to increase the size of the covariance and hence the validation gate at each iteration. This, in turn, increased the uncertainty even further as more and more measurements were associated with each track at each time. We can see that in Figure 7.8, the IPDA was unable to form a track on the ER target in the shallow dataset and appears to form one track on the target in the intermediate dataset but it diverges almost instantly after the track is initiated. The IPDA algorithm also produced a number of spurious false tracks in both environments that were not related to the ER target, most of which are not shown in the figures.

Figure 7.9 shows the tracking outputs for a threshold level of 13 dB. The IPDA algorithm formed a track on the ER target in the shallow dataset and was only able to form a divergent track on the ER target in the intermediate environment. However at this threshold level, the number of false tracks has significantly dropped off. The shallow environment featured only one false IPDA track whereas the intermediate environment produced none. When the SNR threshold level was raised to 15 dB, Figure 7.10 shows that the IPDA was able to form a track on the ER target in both datasets with zero false and divergent tracks. However, it should be noted that the target in both datasets had reasonably high SNR. The likelihood of the IPDA initiating and maintaining a track on a low SNR target at higher thresholds would be negligible as no detections would be obtained. Even if the SNR threshold was lowered, tracking low SNR targets would still be difficult due to the sheer number of point detections.

In the intermediate environment, the IPDA algorithm was not able to successfully detect or track the target at the lower thresholding levels. However, we observe that at the 13 dB thresholding level, the IPDA was able to start a track on the target but the large number of spurious detections at this thresholding level caused the track to diverge. Tables 7.1 and 7.2 show the average number of measurements at each time frame received by the IPDA at each thresholding level for the shallow and intermediate dataset, respectively. The tables also collate the number of false and divergent tracks formed by the IPDA in each case.

Figure 7.11 on page 187 shows the SNR track estimates for the TkBD algorithms assuming a thresholding level of 15 dB for both the shallow and intermediate environment. The performance

Threshold level (dB)	Average number of detections per frame	Number of divergent tracks	Number of false tracks
11	115.4	0	8
13	4.4	0	1
15	0.8	0	0

Table 7.1: Number of false and divergent IPDA tracks at different SNR thresholding levels for the shallow dataset.

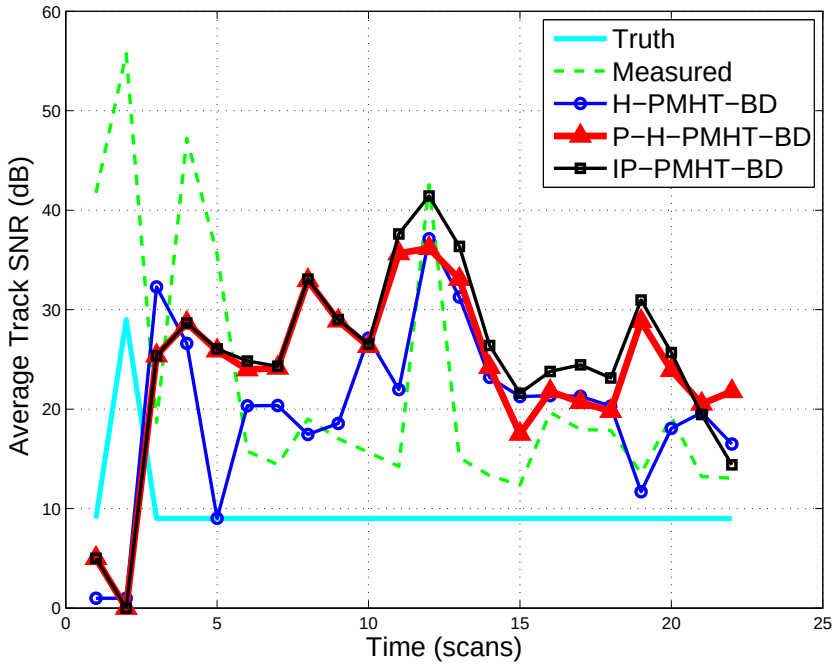
Threshold level (dB)	Average number of detections per frame	Number of divergent tracks	Number of false tracks
11	118.5	1	12
13	5.1	1	0
15	0.7	0	0

Table 7.2: Number of false and divergent IPDA tracks at different SNR thresholding levels for the intermediate dataset.

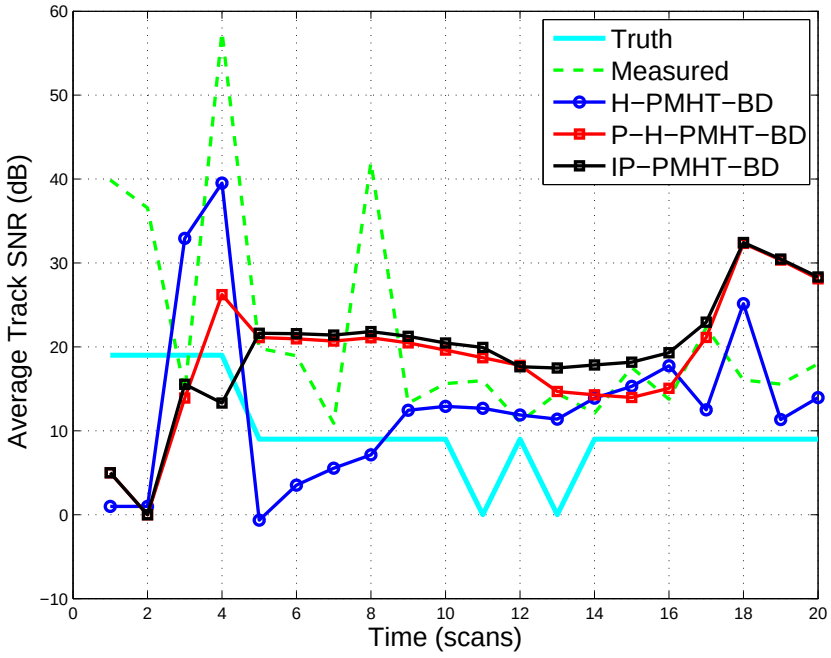
of the SNR track estimates for the lower thresholding levels were very similar and are not shown. As expected, the P-H-PMHT-BD and IP-PMHT-BD algorithms give very similar estimates. Again, both give more smoothed estimates of the track SNR as they assume a dynamics model on the amplitude with a high smoothing factor. This is particularly beneficial in the intermediate environment, in which the P-H-PMHT-BD and IP-PMHT-BD generally give better estimates for the average target SNR.

7.5 Summary

This chapter presents an application of TkBD to trial sonar data from an active towed array system. The key contribution of this chapter is a study comparing several TkBD algorithms discussed in previous chapters in this thesis with a conventional IPDA tracker with multi-target logic using point detections. This chapter also verified that the Poisson H-PMHT and IP-PMHT proposed in this thesis give similar state estimation performance to the standard H-PMHT, and



(a) Shallow



(b) Intermediate

Figure 7.11: Average track SNR estimates: Various TkBD algorithms using a SNR thresholding level of 15 dB

give more smoothed estimates of the target SNR.

Another key contribution is a modification to the TkBD algorithms to include bearing dependent psf to model the change in target appearance with receive array direction. This resulted in improved tracking and less likelihood of track divergence.

It was shown that TkBD can provide significant performance advantages over the conventional IPDA tracking. The IPDA algorithm was able to form a reliable track only when a high SNR threshold was applied during the point-measurement extraction stage. At lower thresholds, it was overwhelmed by the large number of detections and suffered from a high false track rate. In contrast the TkBD algorithms processed the intensity map data directly without an explicit SNR threshold (except artificially imposed for track initiation) and was able to detect the target without forming any false tracks. We can conclude that for active towed array sonar data, TkBD is a promising alternative to conventional point measurement tracking.

Chapter 8

Summary

The proliferation of submarines in the Asia-Pacific region has become a growing concern for the Australian government due to the potential impact on its maritime and naval interests. In response to this growing threat, the Australian government has focused on enhancing its undersea warfare capabilities; one such task involves enhancing its detection and tracking capabilities using active sonar. This task is challenging due to the highly complex nature of the underwater environment, which can be characterised by fluctuating low Signal-to-Noise Ratio (SNR) target returns, high false alarm rate, slow update rates and multi-path effects.

This thesis has considered the problem of detecting and tracking targets using active sonar in the underwater environment. This chapter summarises the main contributions of the thesis and discusses areas for future research. We have proposed a number of enhancements to the target dynamics and measurement models within the Bayesian tracking context for the active sonar problem: a target model based on Hidden Reciprocal Processes (HRPs) was proposed to allow for destination aware tracking; multiple enhancements to the measurement model were introduced for a particular Track-Before-Detect (TkBD) algorithm called the Histogram-Probabilistic Multi-Hypothesis Tracker (H-PMHT) to allow for a randomly evolving target amplitude prior with instantaneous fluctuations.

The chapter is arranged as follows: Section 8.1 presents a summary of the thesis contributions; Section 8.2 discusses the limitations of the thesis and directions for future research.

8.1 Conclusions

8.1.1 Maximum Likelihood Sequence Estimation for Track-Before-Detect

In Bayesian tracking, the target dynamics model is generally assumed to follow a Markov process. Some targets however, travel with a pre-known destination and their dynamics can be described using an acausal model, which includes both past and future information. This thesis has proposed an alternative target model for tracking based on HRPs, which is able to include target destination information. A Maximum Likelihood Sequence Estimator (MLSE) for HRPs was derived using the Markov bridge approach. Through simulations, the resultant estimator was shown to give improved state estimation performance compared to a Markov process for a reciprocal target.

8.1.2 Viterbi Implementation for H-PMHT

In conventional target tracking, the measurement model generally approximates the sensor output, a continuous valued intensity map, with a set of point measurements. However, reducing the sensor data to a set of point detections results in the loss of information. An alternative to point measurement tracking is the concept of TkBD, which is based on providing the continuous valued intensity map directly to the tracker. This thesis has focused on a particular TkBD algorithm called the H-PMHT algorithm, which is based on Expectation-Maximisation (EM) data association and has performance that is close to the optimal Bayesian estimator but still retains linear complexity with the number of targets. The state estimation component of the EM procedure has been demonstrated in the past with Kalman Filter (KF), Particle Filter (PF) and Extended Kalman Filter (EKF) implementations. This thesis presented a Viterbi implementation for the H-PMHT and showed through simulations, that it outperforms the KF in the linear non-Gaussian case. It also can potentially provide better state estimation performance for applications with discrete states.

8.1.3 Poisson Measurement Model for H-PMHT

One of the problems with the H-PMHT algorithm is that it assumes that the component mixing proportions are constant or time-independent. To address this, we proposed an alternative derivation of the H-PMHT based a Poisson measurement model. The resulting algorithm is referred to as the Poisson H-PMHT. We showed that the Poisson assumption is consistent with the multinomial assumption in the standard H-PMHT. Using this approach, we can impose a dynamics model on the component mixing terms to allow for a randomly evolving prior for the

target amplitude that allows for instantaneous fluctuations. Through simulations, we show that the new Poisson mixing terms are less sensitive to noise fluctuations, and are able to give better target estimates of the true mean SNR for fluctuating target models.

8.1.4 Interpolated Poisson Measurement Model for Track-Before-Detect

The H-PMHT uses quantisation of the measurement image to approximate the sensor intensity image by a set of synthetically generated point measurements. This approximation can result in inconsistent performance due to questionable independence assumptions that stem from using a multinomial measurement model. To address this limitation, this thesis proposed a new measurement model based on the application of EM data association to continuous valued data. This approach describes the measurement image directly using an interpolated form of the Poisson distribution. Although this function is not strictly a probability measure, we can make use of convenient properties in measure theory to derive a TkBD algorithm that is similar in principle to the H-PMHT but avoids the issues that arise from quantisation and assumptions of independence. This new algorithm was named the Interpolated Poisson-PMHT (IP-PMHT). Like the Poisson measurement model, this new algorithm also allows for a dynamic model to be imposed on the component mixing terms.

8.1.5 Comparative Study of Track-Before-Detect and Conventional Point Measurement Tracking using Sonar Trial Data

To quantify the performance of the TkBD algorithms proposed in this thesis in the context of the active sonar tracking problem, a comparative study analysing their performance with a conventional point measurement tracker using trial data from a towed array sonar system was considered. The TkBD algorithms considered in the study included the standard H-PMHT, the proposed Poisson H-PMHT and the proposed IP-PMHT algorithm. The point measurement tracker considered was based on an Integrated Probabilistic Data Association (IPDA) tracker. The TkBD algorithms were modified for the active sonar problem to include a bearing dependent point spread function to model the changes of target appearance with receive array bearing. The modified TkBD algorithms returned good tracks for targets with fluctuating amplitudes. Also, the Poisson H-PMHT and the IP-PMHT algorithms returned smoother target SNR estimates for scenarios with highly fluctuating target models. In contrast, the performance of the IPDA was highly dependent on detection thresholding and suffered from false alarms at lower thresholding levels. We can conclude that TkBD is a promising alternative to point measurement tracking that can give consistent tracking performance for the active sonar problem.

8.2 Future Work

8.2.1 Application of Track-Before-Detect to the Active Sonar Problem

Although the thesis demonstrated the performance of the MLSE for a HRP using simulated data, it did not consider the application of this alternative target model to the active sonar problem. In future, it would be beneficial to explore the application of HRP-based estimators to trial sonar data. Generally, maritime platforms such as merchant ships travel with a specific destination in mind while submarine targets travel in paths with unknown intent. HRPs can potentially assist in the classification of benign targets and targets of interest. Using a detection scheme, we can return tracks that diverge from the path assumed by the HRP model to help identify targets behaving in anomalous manner [49].

The research on state estimation using HRPs in this thesis was also only limited to fixed grid approximation techniques for a one-dimensional problem with only a single target. Other potential areas of future research could involve deriving other filters for HRPs, such as the PF or Kalman based filters, and consider extensions to the multi-target and multi-dimensional cases.

8.2.2 Extensions to Sonar Trial Data Application

To allow for a simplified implementation, the trackers considered in the comparative study were only implemented using truncated images from the towed array sonar sensor, centred around the known target position. Further studies of TkBD performance would benefit from considering the entire sonar sensor measurement image. With an enlarged image, it is expected that the number of false tracks will increase as the area of surveillance is expanded. Assuming also that the resolution of the image remains the same, the TkBD algorithms would also increase in complexity as a result of iterating over a larger number of image pixels.

Although the CASSTASS data set considered in this thesis featured a fluctuating target, the observed levels of SNR were relatively high. In future, a comparison of the TkBD performance with conventional point measurement trackers using a dataset featuring a low SNR target may bring to light other issues unique to the active sonar problem.

In the underwater tracking problem, target returns can travel in different paths along the environment. Depending on the environmental parameters, these returns can arrive back at the sensor at different times. This phenomenon is referred to as multi-path propagation and can have adverse effects on tracking, particularly in deep water environments. In deep water, signals can travel very long distances and multi-path returns from a target can appear in multiple

range bins that are a considerable distance apart. This issue can be potentially resolved by modifying the TkBD algorithms to include a range-dependent psf. In future, it would be will beneficial to include this enhancement when tracking in deep water scenarios. However as the range difference in multi-path propagation is highly dependent on environmental parameters, modelling multi-path effects is a challenging issue as environmental information is not always readily available.

8.2.3 Extension of Track-Before-Detect to the Multi-target Active Sonar Tracking Problem

Finally, we note that the simulations and trial data studies conducted in this thesis were limited to scenarios featuring only a single target of interest. As the H-PMHT TkBD paradigm is naturally suited to tracking in multi-target scenarios, it would be useful to assess its performance, and its Poisson variants, for multi-target datasets in the active sonar tracking framework to gain further insight into the potential benefits or disadvantages of the proposed algorithms.

Appendix A

Interpolated Poisson Distribution

A.1 Superposition

The function $f(z;\lambda)$ defined in (6.5) can be be thought of as an interpolated form of the Poisson distribution for non negative continuous variable $Z = z$ with rate parameter λ:

$$f(Z = z;\lambda) = \exp(-\lambda)\frac{\lambda^z}{\Gamma(z+1)}. \tag{A.1}$$

As shown in Figure 6.1, the function $f(z;\lambda)$ is approximately a probability density function for $\lambda > 4$ such that,

$$\int_0^\infty f(z;\lambda)dz = 1, \tag{A.2}$$

Like the Poisson distribution, it can be shown that the interpolated form of the Poisson distribution also has the important properties of superposition. Define $IPoiss(\cdot\ ;\ \lambda)$ to be the interpolated Poisson distribution with rate parameter λ, then superposition means that for $z = z^1 + z^2$ with $z^j \sim IPoiss(z^j;\lambda^j)$, the combined energy also follows an interpolated Poisson distribution with $z \sim IPoiss(z;\lambda^1+\lambda^2)$.

We now prove that the function $f(z;\lambda)$ approximately satisfies the superposition property through characteristic functions. The characteristic function is an alternative way of describing the behaviour of a random variable. A characteristic function $F(t;\lambda)$, can be defined for the random variable Z as following,

$$\begin{aligned} F(t;\lambda) &= E\big[\exp\{jtz\}\big] \\ &\equiv \int_{-\infty}^{\infty}\exp\{jtz\}f(z;\lambda)dz. \end{aligned} \tag{A.3}$$

Substituting (A.1) into (A.3) results in

$$\begin{aligned} F(t;\lambda) &= \exp\{-\lambda\}\int_0^\infty \big[\exp\{jt\}\big]^z \frac{\lambda^z}{\Gamma(z+1)}dz \\ &= \exp\{-\lambda\}\exp\Big\{\lambda\exp\{jt\}\Big\}\int_0^\infty \exp\Big\{-\lambda\exp\{jt\}\Big\}\frac{\Big[\lambda\exp\{jt\}\Big]^z}{\Gamma(z+1)}dz \\ &= \exp\Big\{\lambda\exp\{jt\}-\lambda\Big\}\int_0^\infty f\Big(z;\lambda\exp\{jt\}\Big)dz \\ &= \exp\Big\{\lambda\Big(\exp\{jt\}-1\Big)\Big\}. \end{aligned} \tag{A.4}$$

Note that on the third line of (A.4), it is necessary to assume that $\lambda > 4$ in order to make use of equation (A.2).

Now consider two random variable z^1 and z^2 that are assumed to follow an interpolated Poisson distribution (A.1) with rate parameters λ^1 and λ^2 respectively, where $\lambda^1, \lambda^2 > 4$. For the superposition property to be satisfied, we require that the sum $z = z^1 + z^2$ to also follow an interpolated Poisson distribution with rate parameter $\lambda = \lambda^1 + \lambda^2$. This can easily shown using characteristic functions. Using (A.4), characteristic functions for random variables z^1 and z^2 can defined as following,

$$F(t;\lambda^1) = \exp\Big\{\lambda^1\Big(\exp\{jt\}-1\Big)\Big\}, \tag{A.5}$$

$$F(t;\lambda^2) = \exp\Big\{\lambda^2\Big(\exp\{jt\}-1\Big)\Big\}. \tag{A.6}$$

The characteristic function corresponding to the superposition property is given as following,

$$\begin{aligned} F(t;\lambda^1)\times F(t;\lambda^2) &= \exp\Big\{\lambda^1\Big(\exp\{jt\}-1\Big)\Big\}\times\exp\Big\{\lambda^2\Big(\exp\{jt\}-1\Big)\Big\} \\ &= \exp\Big\{(\lambda^1+\lambda^2)\Big(\exp\{jt\}-1\Big)\Big\} \\ &= F(t,\lambda), \end{aligned} \tag{A.7}$$

where $F(t,\lambda) = \exp\Big\{\lambda\Big(\exp\{jt\}-1\Big)\Big\}$. This implies that the function $f(z,\lambda)$ also has an interpolated Poisson distribution. This proves that for $\lambda > 4$, an approximate superposition property holds for the interpolated Poisson distribution.

A.2 Proof of Integral (6.23) in the Derivation of the Interpolated Poisson-PMHT

In chapter 6, the integral (6.24) has the following form:

$$m(N) = \int_0^N x\frac{\Gamma(N+1)}{\Gamma(x+1)\Gamma(N-x+1)}\lambda^x(A-\lambda)^{N-x}dx. \tag{A.8}$$

To solve this integral, define

$$\tilde{f}(x;\lambda) = xf(x;\lambda). \tag{A.9}$$

where $f(x;\lambda)$ is the interpolated Poisson distribution defined in (A.1). We can then separate the integral (A.8) into two terms such that,

$$\begin{aligned} x\frac{\lambda^x}{\Gamma(x+1)} &= x\exp\{\lambda\}\exp\{-\lambda\}\frac{\lambda^x}{\Gamma(x+1)} \\ &= \exp\{\lambda\}\tilde{f}(x;\lambda), \end{aligned} \tag{A.10}$$

$$\begin{aligned} \frac{(A-\lambda)^{N-x}}{\Gamma(N-x+1)} &= \exp\{A-\lambda\}\exp\{-(A-\lambda)\}\frac{(A-\lambda)^{N-x}}{\Gamma(N-x+1)} \\ &= \exp\{A-\lambda\}f(N-x;A-\lambda). \end{aligned} \tag{A.11}$$

Substituting (A.10) and (A.11) into (A.8), the integral simplifies to:

$$\begin{aligned} m(N) &= \Gamma(N+1)\exp\{\lambda\}\exp\{A-\lambda\}\int_0^N \tilde{f}(x;\lambda)f(N-x;A-\lambda)dx \\ &= \Gamma(N+1)\exp\{A\}\ \tilde{f}(x;\lambda)\otimes f(x;A-\lambda), \end{aligned} \tag{A.12}$$

where $\otimes$ denotes the convolution operator. We now consider solving the convolution term in (A.12). Define $h(N)$ as the convolution term,

$$h(N) = \tilde{f}(x;\lambda)\otimes f(x;A-\lambda). \tag{A.13}$$

The convolution can be solved by forming a corresponding function $H(t)$ using characteristic functions such that

$$.H(t) = \tilde{F}(t;\lambda)\times F(t;A-\lambda). \tag{A.14}$$

The characteristic function $F(t,\lambda)$ is given by (A.4). In a similar way, we can calculate the characteristic function for $\tilde{F}(x,\lambda)$ as following,

$$\begin{aligned} .\tilde{F}(t;\lambda) &\equiv \int_{-\infty}^{\infty}\exp\{jtx\}\tilde{f}(x;\lambda)dx, \\ &= \exp\{-\lambda\}\int_0^{\infty} x\big[\exp\{jt\}\big]^x\frac{\lambda^x}{\Gamma(x+1)}dx, \\ &= \exp\{-\lambda\}\exp\Big\{\lambda\exp\{jt\}\Big\}\int_0^{\infty} x\exp\Big\{-\lambda\exp\{jt\}\Big\}\frac{\Big[\lambda\exp\{jt\}\Big]^x}{\Gamma(x+1)}dx, \\ &= \exp\Big\{\lambda\exp\{jt\}-\lambda\Big\}\int_0^{\infty} xf\Big(x;\lambda\exp\{jt\}\Big)dx. \end{aligned} \tag{A.15}$$

The integral on the last line of (A.15) is the expectation of the interpolated Poisson distribution such that,

$$\int_0^{\infty} xf\Big(x;\lambda\exp\{jt\}\Big)dx = \lambda\exp\{jt\}. \tag{A.16}$$

The characteristic function for $\tilde{F}(x,\lambda)$ simplifies to:

$$\tilde{F}(t;\lambda) = \lambda \exp\{jt\} \exp\Big\{\lambda\big(\exp\{jt\} - 1\big)\Big\}. \tag{A.17}$$

Substituting (A.4) and (A.17) into (A.14) results in

$$\begin{aligned} H(t) &= \tilde{F}(t;\lambda) \times F(t; A-\lambda) \\ &= \lambda \exp\{jt\} \exp\Big\{\lambda\big(\exp\{jt\} - 1\big)\Big\}. \exp\Big\{(A-\lambda)\big(\exp\{jt\} - 1\big)\Big\} \\ &= \lambda \exp\{jt\} \exp\Big\{A(\exp\{jt\} - 1)\Big\} \\ &= \frac{\lambda}{A}\tilde{F}(t;A). \end{aligned} \tag{A.18}$$

This implies that the convolution term in (A.13) simplifies to,

$$\begin{aligned} h(N) &= \frac{\lambda}{A}\tilde{f}(N;A) \\ &= \lambda N \exp\{-A\}\frac{A^{N-1}}{\Gamma(N+1)} \end{aligned} \tag{A.19}$$

Finally, substituting (A.19) into (A.12) gives the following solution to the integral,

$$\begin{aligned} m(N) &= \Gamma(N+1)\exp\{A\}\ h(N) \\ &= \Gamma(N+1)\exp\{A\}\frac{\lambda N \exp\{-A\} A^{N-1}}{\Gamma(N+1)} \\ &= \lambda N A^{N-1}. \end{aligned} \tag{A.20}$$

References

[1] P. L. Ainsleigh. A tutorial on EM-based density estimation with histogram intensity data. Technical report, 11807, NUWC, Newport, RI, USA, June 2009.

[2] C. E. Alissandrakis, D. Dialetis, and G. Tsiropoula. Determination of the mean lifetime of solar features from photographic observations. *Astronomy and Astrophysics*, 174(1–2):275–280, 1987.

[3] M. S. Arulampalam, S. Maskell, N. J. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[4] J. M. Aughenbaugh and B. R. La Cour. Use of prior information in active sonar tracking. In *Proceedings of the 12th International Conference on Information Fusion*, pages 1584–1591, Seattle, USA, July 2009.

[5] V. S. Babu, M. R. Shankar, S. Majumdar, and S. K. Rao. IMM unscented Kalman filter based tracking of maneuvering targets using active sonar measurements. In *2011 International Conference on Communications and Signal Processing (ICCSP)*, pages 126–130, February 2011.

[6] Y. Bar-Shalom, X. R. Li, and T. E. Kirubarajan. *Estimation with Applications to Tracking and Navigation Theory, Algorithm and Software*. Wiley, New York, 2001.

[7] Y. Barniv. Dynamic programming solution for detecting dim moving targets Part II: Analysis. *IEEE Transactions on Aerospace and Electronic Systems*, 23(6):776–788, 1987.

[8] Y. Barniv. Dynamic programming algorithm for detecting dim moving targets. *Multitarget-Multisensor Tracking: Advanced Applications (Yaakov Bar-Shalom, ed.)*, 1990.

[9] Y. Barniv and O. Kella. Dynamic programming solution for detecting dim moving targets. *IEEE Transactions on Aerospace and Electronic Systems*, 21(1):144–156, 1985.

[10] L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[11] I. Bekkerman and J. Tabrikian. Target detection and localization using MIMO radars and sonars. *IEEE Transactions on Signal Processing*, 54(10):3873–3883, October 2006.

[12] E. Biglieri, J. Proakis, and S. Shamai. Fading channels: information-theoretic and communications aspects. *IEEE Transactions on Information Theory*, 44(6):2619–2692, October 1998.

[13] S. Blackman. *Multiple Target Tracking with Radar Applications*. Artech House, Dedham, MA, 1986.

[14] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.

[15] W. Blanding, P. Willett, and Y. Bar-Shalom. ML-PDA: Advances and a new multitarget approach. *EURASIP Journal on Advances in Signal Processing*, 2008:1–13, 2008.

[16] W. R. Blanding, P. K. Willett, Y. Bar-Shalom, and S. Coraluppi. Multisensor track management for targets with fluctuating SNR. *IEEE Transactions on Aerospace and Electronic Systems*, 45(4):1275–1292, October 2009.

[17] Y. Boers and J. N. Driessen. Particle filter based detection for tracking. In *Proceedings of the American Control Conference*, pages 4393–4397, Arlington, VA, USA, 2001.

[18] M. G. S. Bruno, R. V. Araujo, and A. G. Pavlov. Sequential monte carlo methods for joint detection and tracking of multiaspect targets in infrared radar images. *EURASIP Journal on Advances in Signal Processing*, 2008, January 2008.

[19] F. Carravetta and L. B. White. Modelling and estimation for finite-state reciprocal processes. *IEEE Transactions on Automatic Control*, 57(9):2190–2202, September 2012.

[20] D. A. Castanon, B. C. Levy, and A. S. Willsky. Algorithms for the incorporation of predictive information in surveillance theory. *International Journal of Systems Science*, 16:367–382, 1985.

[21] F. R Castella. Sliding window detection probabilities. *IEEE Transactions on Aerospace and Electronic Systems*, 12:815–819, 1976.

[22] S. Ceylan and M. Efe. Performance of Histogram PMHT algorithm for underwater target tracking. In *2010 IEEE 18th Signal Processing and Communications Applications Conference*, pages 871–873, April 2010.

[23] S. Challa and G. W. Pulford. Joint target tracking and classification using radar and ESM sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 37(3):1039–1055, July 2001.

[24] W.C. Chow. Brownian bridge. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1:325–332, 2009.

[25] D. Clark, B. Ristic, B.-N. Vo, and B. T. Vo. Bayesian multi-object filtering with amplitude feature likelihood for unknown object SNR. *IEEE Transactions on Signal Processing*, 58(1):26–37, January 2010.

[26] D. Clark, I.T. Ruiz, Y. Petillot, and J. Bell. Particle phd filter multiple target tracking in sonar image. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):409–416, January 2007.

[27] S. B. Colegrove, A. W. Davis, and J. K. Ayliffe. Track initiation and nearest neighbours incorporated into probabilistic data association. *Journal of Electrical and Electronics Engineering, Australia*, 6(3):191–198, September 1986.

[28] Commonwealth of Australia. 2013 Defence White Paper. `http://www.defence.gov.au/whitepaper2013/docs/WP_2013_web.pdf`. [Accessed:15-August-2014].

[29] S. Coraluppi. Multistatic sonar localization. *IEEE Journal of Oceanic Engineering*, 31(4), October 2006.

[30] S. Coraluppi and C. Carthel. Distributed tracking in multistatic sonar. *IEEE Transactions on Aerospace and Electric System*, 41(3), July 2005.

[31] S. Coraluppi and D. Grimmett. Multistatic sonar tracking. In *Proceedings of SPIE 5096, Signal Processing, Sensor Fusion and Target Recognition XII,399*, 2003.

[32] I.J. Cox and S.L Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2), February 1996.

[33] M. Daun and F. Ehlers. Tracking algorithms for multistatic sonar systems. *EURASIP Journal on Advances in Signal Processing 2010*, 461538, 2010.

[34] S. J. Davey. *Extensions to the Probabilistic Multi-Hypothesis Tracker for Improved Data Association*. PhD thesis, University of Adelaide, 2003.

[35] S. J. Davey. Detecting a small boat using histogram PMHT. *Journal of Advances in Information Fusion*, 6(2):167–186, December 2011.

[36] S. J. Davey. Histogram PMHT with particles. In *Proceedings of the 14th International Conference on Information Fusion*, Chicago, USA, July 2011.

[37] S. J. Davey. Probabilistic multi-hypothesis tracker with an evolving Poisson prior. *IEEE Transactions on Aerospace and Electronic Systems*, 51(1), January 2015.

[38] S. J. Davey and D. A. Gray. Integrated track maintenance for the PMHT via the Hysteresis model. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):93–111, January 2007.

[39] S. J. Davey, M. G. Rutten, and B. Cheung. A comparison of detection performance for several track-before-detect algorithms. In *Proceedings of the 11th International Conference on Information Fusion 2008*, pages 1–8, June-July 2008.

[40] S. J. Davey, H. X. Vu, S. Arulampalam, F. Fletcher, and C.C. Lim. Clutter mapping for Histogram PMHT. In *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, pages 153–156, Gold Coast, Australia, June-July 2014.

[41] S. J. Davey and M. Wieneke. H-PMHT with an unknown arbitrary target. In *Proceedings of the 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 443–448, December 2011.

[42] S. J. Davey, M. Wieneke, and H. Vu. Histogram-PMHT unfettered. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):435–447, June 2013.

[43] A. P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistics Society*, 39(1):1–38, 1977.

[44] A. Doucet, J. F. G. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, NY, USA, 2010.

[45] F. Ehlers, M. Daun, and M. Ulmke. System design and fusion techniques for multistatic active sonar. In *OCEANS 2009*, pages 1–10, Seattle, USA, May 2009.

[46] M. El-Jaber, A. Osman, G. Mellena, and A. Nourledin. Target tracking in multi-static active sonar systems using dynamic programming and Hough transform. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, July 2009.

[47] A. Erdelyi. *Higher Transcendental Functions, Volume II.* McGraw-Hill Book Company Inc, 1953.

[48] L. Fan, X. Zhang, and J. Shi. Track-before-detect procedures for detection of extended object. *EURASIP Journal on Advances in Signal Processing*, 2011(35), 2011.

[49] M. Fanaswala and V. Krishnamurthy. Detection of anomalous trajectory patterns in target tracking via stochastic context-free grammars and reciprocal process models. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):76–90, 2013.

[50] M. Fanaswala, V. Krishnamurthy, and L. B. White. Destination-aware trajectory filtering via syntactic signal processing. In *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3692–3695, Prague, May 2011.

[51] M. Farshchian and R. G. Raj. A multi-scale and adaptive track-before-detect technique for maritime environments. In *Proceedings of the 2011 IEEE Radar Conference (RADAR)*, pages 818–823, May 2011.

[52] W. Feller. *An Introduction to Probability Theory and Its Applications.* John Wiley and Sons, New York, Third edition, 1968.

[53] R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

[54] E. Fishler, A. Haimovich, R. Blum, D. Chizhik, L. Cimini, and R. Valenzuela. MIMO radar: an idea whose time has come. In *Proceedings of the IEEE Radar Conference, 2004*, pages 71–78, April 2004.

[55] P. Fitzsimmons, J. Pitman, and M. Yor. Markovian bridges: construction, palm interpretation, and splicing. In *Seminar on Stochastic Processes*, pages 101–134, 1992.

[56] F. Fletcher and S. Arulampalam. A comparison of existence-based multitarget trackers for multistatic sonar. In *2012 15th International Conference on Information Fusion (FUSION)*, pages 2362–2369, July 2012.

[57] B. Fortin, R. Lherbier, and J.C. Noyer. A track-before-detect approach for extended target tracking in multi-lidar systems using low-level centralized fusion framework. In *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, 2014.

[58] J. E. Gentle. *Random Number Generation and Monte Carlo Methods*. Springer, New York, Second edition, 2003.

[59] N. Gordon, S. Maskell, and T. Kirubarajan. Efficient particle filters for joint tracking and classification. *Proceedings of SPIE: Signal and Data Processing of Small Targets*, 2002.

[60] K. Granström and U. Orguner. Estimation and maintenance of measurement rates for multiple extended target tracking. In *Proceedings of the 15th International Conference on Information Fusion*, pages 2170–2176, Singapore, July 2012.

[61] D. Grimmett. Multistatic target tracking using specular cue initiation and data retrieval. In *Proceedings of the 11th International Conference on Information Fusion*, pages 1–8, Singapore, June/July 2008.

[62] D. Grimmett. SPECSweb multistatic tracking on a truth-blind simulated scenario of the MSTWG. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, July 2009.

[63] D. Grimmett. Specular-cued multistatic sonar tracking on the SEABAR'07 dataset. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, July 2009.

[64] D. J. Grimmett. Automatic identification of specular detections in multistatic sonar system. In *OCEANS 2009*, pages 1–10, Seattle, USA, October 2009.

[65] X. Guyon. *Random Fields on a Network: Modelling, Statistics, and Applications*. Springer Verlag, New York, 1991.

[66] B.K. Habtemariam, R. Tharmarasa, and T. Kirubarajan. Multitarget track before detect with MIMO radars. In *2010 IEEE Aerospace Conference*, pages 1–9, March 2010.

[67] B.K. Habtemariam, R. Tharmarasa, and T. Kirubarajan. PHD filter based track-before-detect for MIMO radars. *Signal Processing*, 92(3):667–678, January 2013.

[68] B.K. Habtemariam, R. Tharmarasa, T. Kirubarajan, D.Grimmett, and C. Wakayama. Multiple detection probabilistic data association filter for multistatic target tracking. In *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pages 1–6, July 2011.

[69] B. K. Habtermariam. *Probability Hypothesis Density Filter Algorithm for Track Before Detect Applications*. PhD thesis, McMaster University, Available from: Open Access Dissertations and Theses. Paper 4500, 2010.

[70] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[71] C. G. Hempel and S. L. Doran. A PMHT algorithm for active sonar. In *Proceedings of SPIE 5430, Acquisition, Tracking, and Pointing XVIII, 132*, July 2004.

[72] A. G. Herzog, T. Binder, F. Friedl, B. Jähne, and E. Kostina. Estimating water-sided vertical gas concentration profiles by inverse modelling. In *Proceedings of 2nd International Conference on Engineering Optimization*, Lisbon, Portugal, September 2010.

[73] R. Hoseinnezhad, B.-N. Vo, and B.-T. Vo. Visual tracking in background subtracted image sequences via Multi-Bernoulli filtering. *IEEE Transactions on Signal Processing*, 61(2):392–397, January 2013.

[74] D. Huang, A. Xue, and Y. Guo. A particle filter track-before-detect algorithm for multi-radar system. *Elektronika ir Elektrotechnika, ISSN 1392–1215*, 2013.

[75] J. Huang, L. Zhang, Q. Zhang, Y. Jin, and M. Jiang. Performance analysis of DOA estimation for MIMO sonar based on experiments. In *IEEE/SP 15th Workshop on Statistical Signal Processing, 2009. SSP '09*, pages 269–272, Aug 2009.

[76] A. Ilienko. Continuous counterparts of Poisson and binomial distributions and their properties. *Annales Univ. Sci. Budapest, Sect. Comp.*, 39:137–147, 2013.

[77] B. Jamison. Reciprocal processes. *Z. Warscheinlichkeitstheorie*, 30:65–86, 1974.

[78] S. J. Julier. A skewed approach to filtering. In *Proceedings of SPIE 3373, Signal and Data Processing of Small Targets*, pages 271–282, September 1998.

[79] T. Kim, A. V. Nefian, and M. J. Broxton. Photometric recovery of apollo metric imagery with the lunarlambertian reflectance. *IEEE Letters*, 46(9):631–633, 2010.

[80] J. F. C. Kingman. *Poisson Processes*. Oxford University Press, 1993.

[81] C. M. Kreucher and B. Shapo. Multitarget detection and tracking using multisensor passive acoustic data. *IEEE Journal of Oceanic Engineering*, 36(2):205–218, 2011.

[82] D. Lerro and Y. Bar-Shalom. Automated tracking with target amplitude information. In *American Control Conference, 1990*, pages 2875–2880, May 1990.

[83] B. C. Levy, R. Frezza, and A. J. Krener. Modelling and estimation of discrete-time gaussian reciprocal processes. *IEEE Transactions on Automatic Control*, 35(9):1013–1023, September 1990.

[84] T. C. Liu and H. Schmidt. UV-based seabed target detection and tracking. In *OCEANS 2002*, volume 1, pages 474–478, October 2002.

[85] T. E. Luginbuhl. Estimation of general, discrete-time FM processes, *Doctoral Dissertaions*, Paper AAI9926264. `http://digitalcommons.uconn.edu/dissertations/AAI9926264`, 1999.

[86] T. E. Luginbuhl and P. Willett. Estimating the parameters of general frequency modulated signals. *IEEE Transactions on Signal Processing*, 52(1):117–131, January 2004.

[87] J. M. F. Moura M. G. Bruno. Multiframe detector/tracker: optimal performances. *IEEE Transactions on Aerospace and Electronic Systems*, 37(3):925–945, July 2001.

[88] R. Mahler. Multi-target Bayes filtering via first-order multi-target moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.

[89] R. Mahler. *Statistical multisource-multitarget information fusion.* Artech House, 2007.

[90] F. Martinerie. Data fusion and tracking using HMMs in a distributed sensor network. *IEEE Transactions on Aerospace and Electronic Systems*, 3(1):11–28, January 1997.

[91] P. S. Maybeck and D. E. Mercier. A target tracker using spatially distributed infrared measurements. *IEEE Transactions on Automatic Control*, 25(2):222–225, 1980.

[92] M. Morelande and B. Ristic. Signal-to-noise ratio threshold effect in track before detect. *Radar, Sonar and Navigation, IET*, 3(6):601–608, December 2009.

[93] D. Musicki and R. Evans. Joint Integrated Probabilistic Data Association: JIPDA. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1093–1099, July 2004.

[94] D. Musicki, R. Evans, and S. Stankovi. Integrated Probabilistic Data Association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, June 1994.

[95] D. Musicki, X. Wang, R. Ellem, and F. Fletcher. Efficient active sonar multitarget tracking. In *OCEANS 2006*, pages 1–8, May 2007.

[96] C. Musso, N. Oudjane, and F. Le Gland. Improving regularised particle filters. In A. Doucet, N. de Freitas, and N. J. Gordon, editors, *Sequential Methods in Practice.* Springer, New York, 2001.

[97] D. Orlando, F. Ehlers, and G. Ricci. Track-before-detect algorithms for bistatic sonar. In *International Workshop on Cognitive Information Processing*, pages 180–185, June 2010.

[98] A. G. Pakfiliz and M. Efe. Multi-target tracking in clutter with histogram probabilistic multi-hypothesis tracker. In *Proceedings of the 18th International Conference on Systems Engineering*, pages 137–142, August 2005.

[99] F. Papi, V. Kyovtorov, R. Giuliani, F. Oliveri, and D. Tarchi. Bernoulli filter for track-before-detect using MIMO radar. *IEEE Signal Processing Letters*, 21(9):1145–1149, September 2014.

[100] F. Papi, B.-T. Vo, M. Bocquel, and B.-N. Vo. Multi-target track-before-detect using labelled random finite set. In *2013 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 116–121, Nov 2013.

[101] A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, Fourth edition, 2002.

[102] N. H. Parsons. A track-before-detect algorithm for active sonar based on a hidden Markov Model. *Journal of the Acoustical Society of America*, 123(5):3947, May 2008.

[103] C. M. Payne. *Principles of Naval Weapons Systems*. Naval Institute Press, Second edition, 2010.

[104] J. W. Pitton and W. L. J. Fox. Incorporating target strength into environmentally-adaptive sonar tracking. In *OCEANS 2007*, pages 1–5, June 2007.

[105] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[106] R. Raheli, A. Polydoros, and T. Ching-Kae. Per-Survivor Processing: a general approach to MLSE in uncertain environments. *IEEE Transactions on Communications*, 43(1):354–364, April 1995.

[107] S. K. Rao. Maneuvering target tracking using pseudo linear estimator with active sonar measurements. In *International Conference on Signal Processing, Communications and Networking 2007, ICSCN '07*, pages 208–213, Feb 2007.

[108] B. Ristic, S. Arulampalam, and N. J. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[109] S. Roelly and M. Thieullen. Duality formula for the bridges of a Brownian diffusion: Application to gradient drifts. *Stochastic Processes and their Applications*, 115(10):1677–1700, 2005.

[110] M. G. Rutten, N. J. Gordon, and S. Maskell. Recursive track-before-detect with target amplitude fluctuation. *IEE Proceedings Radar, Sonar and Navigation*, 152(5):345–352, October 2005.

[111] D. J. Salmond and H. Birch. A particle filter for track-before-detect. In *Proceedings of the American Control Conference*, volume 5, pages 3755–3760, Arlington, VA, USA, June 2001.

[112] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[113] A.-A. Saucan, C. Sintes, T. Chonavel, and J.-M. Le Caillec. Robust, track before detect particle filter for bathymetric sonar application. In *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, 2014.

[114] S. Schoenecker, P. Willett, and Y. Bar-Shalom. A comparison of the ML-PDA and the ML-PMHT algorithms. In *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pages 1–8, July 2011.

[115] L. Stone. Likelihood ratio detection and tracking. In *2009 Proceedings of the 12th International Conference on Information Fusion (FUSION)*, page 10, July 2009.

[116] L. D. Stone, T. L. Corwin, and C. A. Barlow. *Bayesian multiple target tracking*. Artech House, 1999.

[117] R. L. Streit. Tracking on intensity-modulated data streams. Technical report, 11221, NUWC, Newport, RI, USA, May 2000.

[118] R. L. Streit. *Poisson Point Processes: Imaging, Tracking, and Sensing*. Springer, 2010.

[119] R. L. Streit, M.L. Graham, and M.J. Walsh. Multiple target tracking of distributed targets using Histogram-PMHT. In *Proceedings of the 4th International Conference on Information Fusion*, Montreal, QC, Canada, August 2001.

[120] R. L. Streit, M.L. Graham, and M.J. Walsh. Multitarget tracking of distributed targets using Histogram-PMHT. *Digital Signal Processing*, 12(2):394–404, July 2002.

[121] R. L. Streit and T. E. Luginbuhl. Probabilistic multi-hypothesis tracking. Technical report, 10428, NUWC, Newport, RI, USA, February 1995.

[122] M. Swift, J. L. Riley, S. Lourey, and L. Booth. An overview of the multistatic sonar program in Australia. In *Proceedings of the Fifth International Symposium on Signal Processing and Its Applications*, volume 1, pages 321–324, 1999.

[123] M. Taj and A. Cavallaro. Multi-camera track-before-detect. In *Third ACM/IEEE International Conference on Distributed Smart Cameras, 2009. ICDSC 2009*, pages 1–6, Aug 2009.

[124] S. M Tonissen. *Target tracking in low SNR environments.* PhD thesis, University of Melbourne, 1996.

[125] S. M. Tonissen and R. J. Evans. Performance of dynamic programming techniques for track-before-detect. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1440–145, October 1996.

[126] S. M. Tonissen and R. J. Evans. Techniques for low SNR target detection and tracking. In *Proceedings of the Fourth International Symposium on Signal Processing and its Application*, volume 1, pages 21–24, Gold Coast, Australia, August 1996.

[127] J. M. Turowski. Probability distributions of bed load transport rates: A new derivation and comparison with field data. *Water Resources Research*, 46, 2010.

[128] T. Vercauteren, D. Guo, and X. Wang. Joint multiple target tracking and classification in collaborative sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):714–723, April 2005.

[129] B.-N. Vo and W.-K. Ma. The Gaussian mixture Probability Hypothesis Density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, 2006.

[130] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter. Joint detection and estimation of multiple objects from image observations. *IEEE Transactions on Signal Processing*, 58(10):5129–5141, October 2010.

[131] B.-T. Vo, B.-N. Vo, and A. Cantoni. Bayesian filtering with random finite set observations. *IEEE Transactions on Signal Processing*, 56(4):1313–1326, 2008.

[132] H. X. Vu. *Track-Before-Detect for Active Sonar.* PhD thesis, University of Adelaide, 2014.

[133] H. X. Vu and S. J. Davey. Track-before-detect using Histogram PMHT and dynamic programming. In *Proceedings of the 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–8, December 2012.

[134] H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.-C. Lim. H-PMHT with a Poisson measurement model. In *Proceedings of the 2013 International Conference on Radar (Radar)*, pages 446–451, September 2013.

[135] H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.C. Lim. A new state prior for the Histogram-PMHT. *IEEE Signal Processing Letters, in preparation.*

[136] H. X. Vu, S. J. Davey, S. Arulampalam, F. K. Fletcher, and C.C. Lim. Histogram-PMHT with an evolving Poisson prior. In *2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2015.

[137] H. X. Vu, S. J. Davey, F. K. Fletcher, S. Arulampalam, R. Ellem, and C.-C. Lim. Track-before-detect for an active towed array sonar. In *Proceedings of Acoustics 2013*, November 2013.

[138] A. D. Waite. *SONAR for Practising Engineers.* Wiley, Third edition, 2002.

[139] A. Wald. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[140] M. J. Walsh, M. L Graham, R. L. Streit, T. E Luginbuhl, and L. A. Andrews. Tracking on intensity-modulated sensor data streams. *IEEE Proceedings of the Aerospace Conference, 2001*, 4:1901–1909, 2001.

[141] E. A. Wan and R. Van der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000*, pages 153–158, October 2000.

[142] J. Wang, A. von Trojan, and S. Lourey. ctive sonar target tracking for anti-submarine warfare applications. In *IEEE OCEANS 2010*, pages 1–7, Sydney, Australia, May 2010.

[143] G. I. Webb. Multiboosting: a technique for combining boosting and wagging. *Machine Learning*, 40:159–196, 2000.

[144] T. A. Wettergren. Performance of search via track-before-detect for distributed sensor networks. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1):314 –325, January 2008.

[145] T. A. Wettergren and M. J. Walsh. Localization accuracy of track-before-detect search strategies for distributed sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2008, January 2008.

[146] L. B. White. Optimal smoothing for hidden reciprocal models. In *Proceedings of Defence Application of Signal Processing*, Hawaii, 2009.

[147] L. B. White and F. Carravetta. Optimal smoothing for finite-state hidden reciprocal processes. *IEEE Transactions on Automatic Control*, 56(9):2156–2161, 2011.

[148] L. B. White and H. Vu. Track-before-detect using maximum likelihood sequence estimation for hidden reciprocal processes. In *Proceedings of Defense Applications of Signal Processing*, July 2011.

[149] L. B. White and H. X. Vu. Maximum likelihood sequence estimation for hidden reciprocal processes. *IEEE Transactions on Automatic Control*, 58(10):2670–2674, October 2013.

[150] M. Wieneke and S. J. Davey. Histogram PMHT with target extent estimates based on random matrices. In *Proceedings of the 14th International Conference on Information Fusion*, pages 322–329, July 2011.

[151] J. Wong, B.-T. Vo, B.-N. Vo, and R. Hoseinnezhad. Multi-Bernoulli based track-before-detect with road constraints. In *Proceedings of the 15th International Conference on Information Fusion*, pages 840–846, July 2012.

[152] T. M. Wood. *Random finite sets for multitarget tracking.* PhD thesis, University of Oxford, 2011.

[153] S. Z. Xia and H.W Liu. Research for underwater target tracking by using multi-sonar. *Radar, Sonar and Navigation, IET*, 6(8):719–728, October 2012.

[154] P. Xie, F. Kang, and S. Wang. Research for underwater target tracking by using multi-sonar. In *Proceedings of the 2010 3rd International Congress on Image and Signal Processing*, volume 9, pages 4249–4253, October 2010.

[155] W. Yi, M. R. Morelande, L. Kong, and J. Yang. An efficient multi-frame track-before-detect algorithm for multi-target tracking. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):421–434, June 2013.

[156] H. Yong and G.Jian. A track-before-detect algorithm for statistical MIMO radar multi-target detection. In *2010 IEEE Radar Conference*, pages 12–16, May 2010.

[157] R. Zhan, D. Lu, and J. Zhang. Maneuvering targets track-before-detect using multiple-model multi-bernoulli filtering. In *2013 International Conference on Information Technology and Applications*, pages 348–353, November 2013.