Extensions to the Probabilistic Multi-Hypothesis Tracker for Improved Data Association

Samuel J Davey

Thesis submitted for the degree of

Doctor of Philosophy

School of Electrical and Electronic Engineering
Faculty of Engineering
The University of Adelaide
Adelaide, South Australia

September 2003

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Overview of Multi-target Tracking . . . 3
  - 1.2.1 PMHT Measurement Model . . . 3
- 1.3 Thesis Overview . . . 4

2 Background **9**
- 2.1 State Estimation . . . 9
- 2.2 Data Association . . . 10
  - 2.2.1 Nearest Neighbour . . . 11
  - 2.2.2 Track Split . . . 11
  - 2.2.3 Multi-Hypothesis Tracker . . . 12
  - 2.2.4 Viterbi Algorithm . . . 12
  - 2.2.5 Assignment Techniques . . . 13
  - 2.2.6 Bayesian Data Association . . . 15
  - 2.2.7 Probabilistic Multi-Hypothesis Tracker . . . 17
  - 2.2.8 Histogram PMHT . . . 20
  - 2.2.9 Markov-chain Monte Carlo . . . 21
  - 2.2.10 Probabilistic Least Squares Tracker . . . 21
- 2.3 Tracking with Augmented Measurements . . . 21
  - 2.3.1 Augmented Measurement Vector . . . 22
  - 2.3.2 Augmented State Vector . . . 23
  - 2.3.3 Joint Classification and Tracking . . . 23
- 2.4 Track Initiation . . . 23
  - 2.4.1 M of N initiation . . . 25
  - 2.4.2 Accumulated Log-Likelihood . . . 25
  - 2.4.3 Model Order Estimation Techniques . . . 25
  - 2.4.4 Hidden Markov Model . . . 26
  - 2.4.5 Radon and Hough Transforms . . . 29

3 Multi-Target Problem Formulation **31**
- 3.1 Problem Definition . . . 31
- 3.2 The Observer . . . 32
  - 3.2.1 The Assignment Model . . . 33
  - 3.2.2 The Measurement Process . . . 34
- 3.3 Target and Measurement Models . . . 34
  - 3.3.1 State Evolution Model . . . 35
  - 3.3.2 Measurement Model . . . 37
- 3.4 The Kalman Filter . . . 38

3.4.1 The Kalman Smoother . . . 39
3.5 The Probabilistic Multi-Hypothesis Tracker . . . 40
3.5.1 Homothetic Measurement Models . . . 42
3.5.2 Manoeuvring Target Models . . . 44
3.5.3 PMHT Permutations . . . 45
3.6 Problem Areas in PMHT . . . 45
3.6.1 Additional Sensor Information . . . 45
3.6.2 Dynamic Assignment Prior . . . 46
3.6.3 Track Initiation . . . 46
3.6.4 Application of PMHT to Operational Sensor Systems . . . 46

4 The PMHT with Classification Measurements 47
4.1 Derivation . . . 48
4.1.1 Classification measurements . . . 48
4.1.2 Complete data likelihood . . . 50
4.1.3 Conditional probability of the assignments . . . 50
4.1.4 Expectation Step . . . 52
4.1.5 Maximisation Step . . . 53
4.2 Summary of the PMHT-c algorithm . . . 57
4.3 Special Cases . . . 59
4.3.1 Uninformative classification measurements . . . 59
4.3.2 Perfect classification measurements . . . 59
4.4 Performance Analysis . . . 60
4.4.1 Simulation Details . . . 61
4.4.2 Performance Metrics . . . 62
4.4.3 Improvement with known confusion matrix . . . 64
4.4.4 Improvement with unknown confusion matrix . . . 68
4.5 Sensitivity of the PMHT-c . . . 73
4.5.1 Improved Performance with Mismatch . . . 75
4.6 Value of Estimating the Confusion Matrix . . . 81
4.7 Summary . . . 83

5 The PMHT with Hysteresis 85
5.1 The Standard PMHT Measurement Assignment Model . . . 85
5.2 Assignment State Model . . . 87
5.2.1 A Note on Terminology . . . 89
5.2.2 PMHT with Hysteresis . . . 90
5.3 Hysteresis as Missing Data, PMHT-ym . . . 90
5.3.1 Assignment Weights . . . 93
5.3.2 Statement of PMHT-ym Algorithm . . . 95
5.4 Estimated Assignment State, PMHT-ye . . . 96
5.4.1 Modified Auxiliary Function . . . 96
5.4.2 Assignment State Sequence Estimate . . . 98
5.4.3 Statement of PMHT-ye Algorithm . . . 100
5.4.4 Approximate PMHT-ye with Reduced Complexity . . . 100
5.4.5 Special Cases . . . 101
5.5 Comparison of PMHT-ym and PMHT-ye . . . 102
5.6 Unknown Assignment State Parameters . . . 103
5.7 Simulated Example . . . 104

5.7.1 Assumed Assignment State Model . . . 105
5.7.2 State Estimation Performance . . . 105
5.7.3 Prior Estimation Performance . . . 107
5.8 Summary . . . 109

6 Initiation and Initialisation with the PMHT 113
6.1 Innovation Homothetic Model . . . 114
6.2 Initiation Methodology . . . 116
6.3 Candidate Tests using the Standard PMHT . . . 118
6.3.1 Sum of Weights Quality Statistic . . . 118
6.3.2 Cost Function Increment . . . 119
6.4 PMHT with Hysteresis for Initiation . . . 121
6.4.1 Initiation with PMHT-ym . . . 122
6.4.2 Initiation with PMHT-ye . . . 123
6.5 Performance Analysis Method . . . 125
6.5.1 The Receiver Operating Characteristic Curve . . . 125
6.5.2 Other Performance Assessment Approaches . . . 126
6.5.3 Generating ROC Curves From Simulated Data . . . 126
6.6 Simulated Track Initiation Performance . . . 127
6.6.1 Divergent Tracks . . . 129
6.6.2 Uniform Clutter Distribution . . . 129
6.6.3 Chi-Squared Clutter Distribution . . . 132
6.6.4 Polynomial Clutter Distribution . . . 135
6.6.5 Initialisation Robustness . . . 136
6.6.6 Simulation Conclusions . . . 138
6.7 Summary . . . 138

7 Applying the PMHT to OTHR 141
7.1 Over the Horizon Radar Fundamentals . . . 141
7.1.1 Jindalee Facility at Alice Springs . . . 143
7.2 Specific Over the Horizon Radar Models . . . 143
7.2.1 Measurement Vector . . . 143
7.2.2 Target State Representation . . . 145
7.2.3 Clutter state model . . . 148
7.3 Initialising from Ambiguous Doppler Measurements . . . 149
7.3.1 Estimation Approach for Velocity Unwrapping . . . 153
7.3.2 Unwrapped Velocity as a Mixture Model . . . 153
7.3.3 Doppler Unwrapping Performance . . . 156
7.4 Using PMHT-c for Clutter Parameterisation . . . 157
7.4.1 Centralised Fusion Option . . . 159
7.4.2 Simulated Performance of PMHT-c for Clutter Parameterisation . . . 161
7.4.3 Simulation Results . . . 163
7.4.4 Summary of Clutter Modelling with PMHT-c . . . 164
7.5 Full Radar Algorithm . . . 166
7.6 Radar Data Performance . . . 166
7.6.1 Data Set Features . . . 167
7.6.2 Clutter Parameterisation . . . 167
7.6.3 Track Initiation . . . 168
7.7 Summary . . . 169

8 Comparison of the PMHT with the PDAF **171**
8.1 The Probabilistic Data Association Filter . . . 171
8.1.1 The Unified PDAF . . . 172
8.1.2 The Unified Joint PDAF . . . 173
8.2 JPDAF compared with PMHT . . . 173
8.2.1 Philosophical Differences . . . 173
8.2.2 Practical Differences . . . 174
8.2.3 Existing Comparison Studies . . . 175
8.3 Track Initiation on Simulated Data . . . 175
8.3.1 Initialisation Robustness . . . 177
8.4 Track Initiation on JFAS Data . . . 178
8.4.1 Initialisation Robustness . . . 178
8.5 Established Track Performance . . . 178
8.6 Summary . . . 179

9 Summary **181**
9.1 Classification Measurements . . . 181
9.2 Dynamic Assignment Prior . . . 181
9.3 Track Initiation . . . 182
9.4 Radar Data Performance . . . 183
9.5 Comparison with PDAF . . . 183
9.6 Future Work . . . 184
9.6.1 Classification Information . . . 184
9.6.2 Prior Dynamics . . . 184
9.6.3 Track Initiation and Initialisation . . . 184
9.6.4 OTHR Implementation . . . 185
9.6.5 PMHT and PDAF Comparison . . . 185
9.7 Conclusions . . . 185

A Innovation Homothetic Model **187**
A.1 Maximisation Step . . . 189
A.2 Target State Maximisation . . . 190

B Full Radar Algorithm **193**
B.1 Statement of Algorithm . . . 193

C Review of PDAF **197**
C.1 Nearest Neighbour Gating . . . 200
C.2 Target Visibility . . . 200
C.3 Augmented PDAF . . . 202
C.4 Sophisticated Clutter Models . . . 202
C.5 Sophisticated Target Models . . . 203
C.6 The Unified PDAF . . . 204
C.7 The Joint UPDAF . . . 204

List of Figures

1.1 Measurement Model BINs 4

2.1 Joint Tracking and Classification approach of Challa and Pulford 23
2.2 General initiation flow diagram 24
2.3 Generalised Pseudo Bayes, order 3 27
2.4 IPDA flow diagram 28
2.5 IMM flow diagram 29

4.1 Simulated scenarios 60
4.2 Examples of simulated trials 63
4.3 Bias of estimator $\hat{r}_t^1$, C known, crossing scenario 66
4.4 Variance of estimator $\hat{r}_t^1$, C known, crossing scenario 66
4.5 Bias of estimator $\hat{r}_t^1$, C known, turning scenario 67
4.6 Variance of estimator $\hat{r}_t^1$, C known, turning scenario 67
4.7 bias of estimator $\hat{r}_t^1$, C estimated, turning scenario 70
4.8 variance of estimator $\hat{r}_t^1$, C estimated, turning scenario 71
4.9 Bias of the estimator, $\hat{\alpha}$ 72
4.10 Variance of the estimator, $\hat{\alpha}$ 72
4.11 Bias of the estimator $\hat{r}_t^1$ with C mismatched, turning scenario 76
4.12 variance of the estimator $\hat{r}_t^1$ with C mismatched, turning scenario 77
4.13 Bias of the estimator $\hat{r}_t^1$ when $\beta = 0.7$ is assumed, turning scenario 78
4.14 Variance of the estimator $\hat{r}_t^1$ when $\beta = 0.7$ is assumed, turning scenario 78
4.15 Assignment weights for PMHT 79
4.16 Assignment weights for PMHT-c when $\beta = 0.7$ 80
4.17 Assignment weights for PMHT-c when $\beta = 0.9$ 80
4.18 Assignment weights for PMHT-c when $\alpha = 0.7$ 82
4.19 Assignment weights for PMHT-c when $\alpha = 0.9$ 82
4.20 Assignment weights for PMHT 83

5.1 Measurement Model BINs 86
5.2 BIN for PMHT with hysteresis 87
5.3 Flow diagrams for PMHTs with hysteresis 103
5.4 Example trial with $f = 1$ 105
5.5 Single trial prior estimation, $f = 1$ 108
5.6 PMHT-ym prior probability mass 108
5.7 Bias of estimator $\hat{\pi}_t^1$, $f = 1$ 110
5.8 Variance of estimator $\hat{\pi}_t^1$, $f = 1$ 110
5.9 Single trial prior estimation, $f = 4$ 111
5.10 Bias of estimator $\hat{\pi}_t^1$, $f = 4$ 112
5.11 Variance of estimator $\hat{\pi}_t^1$, $f = 4$ 112

6.1 Growth of the innovation covariance under poor initialisation 114
6.2 Correction for poor initialisation with innovation homothetic model 116
6.3 General initiation flow diagram 117
6.4 Alternative initiation flow diagram 118
6.5 Transition matrix, $\Delta_t^m \left(d_t^m = j | d_{t-1}^m = i\right)$, for $M_D = 51$ 124
6.6 Example tracker ROC curve 126
6.7 Uniform Clutter, low rate of false detections 131
6.8 Uniform Clutter, high rate of false detections 131
6.9 Example trial with target for chi-squared clutter 132
6.10 Chi-Squared Clutter, low rate of false detections 134
6.11 Chi-Squared Clutter, high rate of false detections 134
6.12 Non-uniform clutter densities 135
6.13 Polynomial Clutter, low rate of false detections 137
6.14 Polynomial Clutter, high rate of false detections 137

7.1 Extended radar range through electromagnetic refraction 142
7.2 Typical multipath propagation 143
7.3 Typical OTHR data processing 144
7.4 Data processing stages for the Jindalee OTHR 145
7.5 Observed Doppler frequency shift due to aliasing 151
7.6 Posterior probability density of radial velocity 151
7.7 Clutter Segmentation Process 158
7.8 Distributed Fusion Model 159
7.9 Centralised Fusion Model 159
7.10 Estimated feature pdf, $p\left(z_{tr}^{(f)}\right)$ 160
7.11 PMHT-c for clutter parameterisation, uninformative classifications 163
7.12 PMHT-c for clutter parameterisation, medium veracity classifications . . . 164
7.13 PMHT-c for clutter parameterisation, perfect classifications 165
7.14 ROC curves for different PMHT clutter approaches 168
7.15 PMHT clutter model sensitivity 169
7.16 Recorded JFAS data initiation ROC 170

8.1 Uniform Clutter, high density 176
8.2 Chi-Squared Clutter, high density 176
8.3 Polynomial Clutter, high density 177
8.4 ROC curves for JFAS data 179

List of Tables

4.1 Number of trials with correct tracking with known confusion matrix 65
4.2 Number of trials with correct tracking with estimated confusion matrix . . 69
4.3 Crossing target scenario 74
4.4 Turning target scenario 74

5.1 State estimation bias 106
5.2 State estimation variance 106
5.3 State estimation RMS error 107

6.1 Number of divergent trials 136

7.1 Proportion of correctly unwrapped velocities 157
7.2 Clutter parameters 162

8.1 Number of divergent trials 178

Abstract

Multitarget tracking is a state space estimation problem where false measurements, missed detections, and uncertainty in the source of measurements provide the challenge. The Probabilistic Multi-Hypothesis Tracker (PMHT) is an algorithm which solves the Multitarget tracking problem through application of the Expectation Maximisation algorithm. This algorithm has a number of advantages over traditional techniques, but has not undergone the same degree of development as more established algorithms. This thesis presents extensions to the PMHT which both generalise its fundamental problem formulation, and address practical issues arising in the use of real sensors.

The PMHT is extended to incorporate augmented measurements, which consist of the normal state observations, and classification measurements not considered under the standard PMHT. These classification measurements are interpreted as observations of the assignments, and a PMHT algorithm is derived. The classification measurements improve data association, and simulations are used to demonstrate the effect this has on state estimation accuracy.

The probabilistic assignment model, central to the PMHT, is generalised to allow for an assignment prior distribution which varies smoothly with time. The prior is modelled as a random process following a first order Markov chain. Simulations are used to demonstrate the performance of the algorithm under a time evolving assignment prior.

The PMHT assumption of a constant and known number of targets is relaxed by developing automatic track initiation schemes which are used to reject superfluous candidate models. Several approaches, similar to those used in other tracking algorithms, are considered, and the best of these is found by simulations involving various clutter conditions.

The above extensions are applied to the problem of Over the Horizon Radar (OTHR) tracking and a prototype OTHR tracker is developed. A number of OTHR specific problems are also addressed, and the performance of the PMHT extensions is measured on data recorded from an operational OTHR. The performance of the PMHT prototype is compared with the existing tracking algorithm, which is based on the Probabilistic Data Association Filter.

Declaration

This work contains no material that has been accepted for the award of any other degree or diploma in any university or 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.

I give my consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Signature

Date

Find rest, O my soul, in God alone;
for my hope comes from him.
He alone is my rock and my salvation;
he is my fortress; I will not be shaken.
My salvation and my honour depend on God;
he is my mighty rock, my refuge.
Trust in him at all times, O people;
pour out your hearts to him,
for God is our refuge.

Psalm 62, v 5-8, NIV translation

Acknowledgements

I would like to express my appreciation to all those who provided me with help and encouragement throughout my candidature.

To Professor Doug Gray, of the School of Electrical and Electronic Engineering at the University of Adelaide, for providing objective and constructive guidance, demanding a high standard of research, and encouraging a broad base of learning beyond a narrow problem focus.

To Dr Bren Colegrove, of the Defence Science and Technology Organisation, for providing a sounding board for my ideas, and keeping me in touch with the practical problems of real systems.

To the executive of Intelligence, Surveillance and Reconnaissance Division (formerly Surveillance Systems Division), Defence Science and Technology Organisation, for providing me with the opportunity to undertake this research, for giving me the latitude to find my own direction, and providing access to real sensor data, which has served as both motivation and benchmark for the work.

To the Cooperative Research Centre for Sensor, Signal and Information Processing, and the School of Electrical and Electronic Engineering at the University of Adelaide, for providing me with the facilities necessary to undergo the work, an environment of scholastic excellence, and the opportunity to meet researchers of international standing.

To Dr Roy Streit of the Naval Undersea Warfare Centre, Newport, Rhode Island, for inspiring me to strive for theoretical purity, and for illuminating discussions on PMHT and probability.

To Mik Newton for volunteering, and providing timely and thorough proof reading.

To my wife, Caprice, for unflagging support and believing in me, even when I did not. And to Anne, for sleeping just enough to give me time to finish this.

Abbreviations

BIN	Bayesian Inference Network
CRLB	Cramer-Rao Lower Bound
CSSIP	Cooperative research centre for Sensor Signal and Information Processing
DSTO	Defence Science and Technology Organisation
EM	Expectation Maximisation
ESM	Electronic Support Measure
HF	High Frequency
HMM	Hidden Markov Model
IMM	Interacting Multiple Model
IPDA	Integrated Probabilistic Data Association
IPDAF	Integrated Probabilistic Data Association Filter
ISRD	Intelligence, Surveillance and Reconnaissance Division
JFAS	the Jindalee Facility at Alice Springs
JORN	the Jindalee Over the horizon Radar Network
JPDA	Joint Probabilistic Data Association
JPDAF	Joint Probabilistic Data Association Filter
OTHR	Over The Horizon Radar
MAP	Maximum A Priori
MHT	Multiple Hypothesis Tracker
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
MM-UJPDAF	Multiple Model Unified Joint Probabilistic Data Association Filter
MM-UPDAF	Multiple Model Unified Probabilistic Data Association Filter

pdf	probability density function
pmf	probability mass function
PDA	Probabilistic Data Association
PDAF	Probabilistic Data Association Filter
PMHT	Probabilistic Multi-Hypothesis Tracker
PMHT-c	Probabilistic Multi-Hypothesis Tracker with Classification measurements
PMHT-v	Probabilistic Multi-Hypothesis Tracker with Visibility
PMHT-y	Probabilistic Multi-Hypothesis Tracker with Hysteresis
PMHT-ye	Probabilistic Multi-Hypothesis Tracker with estimated assignment state
PMHT-ym	Probabilistic Multi-Hypothesis Tracker with the assignment state modelled as missing data
ROC	Receiver Operating Characteristic
SNR	Signal to Noise Ratio
UJPDAF	Unified Joint Probabilistic Data Association Filter
UPDAF	Unified Probabilistic Data Association Filter

Symbols

$\{\cdot\}^{\mathsf{T}}$	Matrix transpose operator.
$\lvert \cdot \rvert$	Matrix determinant.
$.^{(i)}$	An estimate formed by the ith EM iteration.
$\hat{B}_t^e$	Estimated bias of the estimator e.
C	The confusion matrix, gives the probability of observing a particular class given the true class.
c_{ij}	An element of the confusion matrix, $\mathsf{C} = \{c_{ij}\}$.
$\hat{c}_{pq}^{(i)}$	The estimate of c_{pq} on the ith EM iteration.
$\operatorname{cov}\{\cdot\}$	Covariance operator.
$\boldsymbol{D}$	The set of all assignment states for the whole batch.
$\boldsymbol{D}_t$	The set of assignment states for scan t.
d_t^m	The assignment state for model $m - M_Y$ at scan t.
$E\{\cdot\}$	Expected value operator.
F_t^m	State transition matrix for model m at scan t.
f_c	Carrier frequency of the transmitted waveform.
f_w	Waveform repetition frequency of the transmitted waveform.
G_t^m	Process noise gain for model m at scan t.
H_t^m	Measurement matrix for model m at scan t.
I	The identity matrix.
I	The fixed number of nearest neighbours validated by the UPDAF.
$\mathbf{K}$	The set of all assignment indices over the data batch.
$\mathbf{K}_t$	The set of the assignment indices for all measurements at scan t.
k_{tr}	The assignment index for the rth measurement at scan t. Indicates which model is the true source for that measurement.

k_{tr}^1	The state model assignment part of k_{tr} when homothetic measurement models are used.
k_{tr}^2	The homothetic measurement model assignment part of k_{tr} when homothetic measurement models are used.
$L(\mathbf{O}, \mathbf{Z})$	The complete data likelihood.
M	The total number of models.
M_C	The number of classes.
M_D	The length on the assignment state space.
M_X	The number of target models.
M_Y	The number of clutter models.
M_μ	The number of dynamic component models for each target model.
m	A model index.
n_t	The number of measurements at scan t.
$\mathbf{O}$	The batch observer.
$\mathbf{O}_t$	The observer at scan t.
P_0	The covariance of the assumed distribution of the initial target state for the linear Gaussian case.
P_t^m	The covariance of the state estimate for model m at scan t.
$\mathsf{P}_{t\|t-1}^m$	The covariance of the predicted state for model m at scan t.
$\mathbf{P}_\mu$	The set of all dynamic model probabilities for OTHR velocity unwrapping.
$\mathbf{P}_\mu^m$	The set of all dynamic model probabilities for model m.
$P_\mu^m(p)$	The dynamic model probability for component p of model m.
Pd	The probability of detecting a target on any particular scan.
Pg	The probability that the target orientated measurement is within the validation region, given that it was detected.
Ps	The probability that the target orientated measurement is within the I nearest neighbours of the model, given that it was detected.
$Q\left(\cdot\|\cdot^{(i)}\right)$	The EM auxiliary function that is maximised to obtain the iterative parameter estimates. It is a function of the true parameters and their estimates from the previous EM iteration.
Q_C	The part of the auxiliary function dependent on the confusion matrix. This is maximised to find the confusion matrix estimate.

$Q_{t\rho}^m$	The part of the auxiliary function dependent on the homothetic mixing proportions rho_t^{mp}.
$Q_{t\pi}$	The part of the auxiliary function dependent on the assignment prior, $\mathbf{\Pi}_t$.
Q_D	The part of the auxiliary function dependent on the assignment state sequence.
Q_D^m	A separable approximation to the part of the auxiliary function dependent on the assignment state sequence for model m.
Q_X^m	The part of the auxiliary function dependent on the states of model m.
Q_t^m	The process noise covariance for model m at scan t.
q^m	A quality statistic for candidate model m.
R_t^m	The measurement covariance matrix for model m at scan t.
$\tilde{\mathsf{R}}_t^m$	The synthetic measurement covariance matrix for model m at scan t. The equivalent covariance used to implement multiple measurement estimation with a Kalman Filter.
r	A measurement index for measurements at a particular scan.
S_t^m	The innovation covariance matrix for model m at scan t. Represents the expected measurement scatter given the current state estimate and its covariance.
T	The number of scans in the batch.
t	A time index, indicating scan number t.
W_t^m	The Kalman Gain for model m at scan t.
w	An assignment weight. The posterior probability of a particular assignment given the current estimated parameters.
$\mathbf{X}$	A set of all of the states of all models over the entire batch.
$\mathbf{X}_t$	A set of the states of all models at scan t.
$\mathbf{X}^m$	A set of the states of model m over the entire batch.
$\mathcal{X}_{t1}^{t2}$	A set of the states of all models from scan $t1$ until scan $t2$.
$\boldsymbol{x}_t^m$	The state of model m at scan t.
$\hat{\boldsymbol{x}}_t^{m(i)}$	The state estimate at the ith EM iteration for model m at scan t.
$\boldsymbol{x}_t^{m,p}$	The state of the pth dynamic component of model m at scan t.
$\mathbf{Z}$	A set of all of the measurements for the entire batch.
$\mathbf{Z}^{(f)}$	A set of all of the feature measurements for the entire batch.

$\mathbf{Z}^{(x)}$	A set of all of the state observations for the entire batch.
$\mathbf{Z}_t$	A set of all of the measurements for scan t.
$\mathcal{Z}_{t1}^{t2}$	A set of all of the measurements from scan $t1$ until scan $t2$.
$\boldsymbol{z}_{tr}$	The rth measurement at scan t.
$\boldsymbol{z}_{tr}^{(f)}$	A feature measurement corresponding to $\boldsymbol{z}_{tr}$.
$\boldsymbol{z}_{tr}^{(k)}$	The classification part of z_{tr}.
$\boldsymbol{z}_{tr}^{(x)}$	The state observation part of z_{tr}.
$\tilde{\boldsymbol{z}}_t^m$	The synthetic measurement for model m at scan t. Used to estimate the target state via a Kalman Filter.
α	Classification veracity; the probability of the classifier being correct.
$\alpha_t\left(\boldsymbol{D}_t\right)$	Forwards part of Hidden Markov Model smoother; posterior assignment state probability.
β	The assumed classification veracity when there is mismatch in the assumed confusion matrix.
$\beta_t\left(\boldsymbol{D}_t\right)$	Backwards part of Hidden Markov Model smoother; retrodicted assignment state probability.
$\beta_t^m(i)$	An event probability for the PDAF.
$\gamma_t\left(\boldsymbol{D}_t\right)$	Probability of the assignment state given the whole batch.
Δ_{01}	Transition probability from invisible to visible.
Δ_{11}	Transition probability from visible to visible.
Δ_0^m	Prior assignment state probability mass function.
Δ_t^m	Assignment state evolution probability mass function.
Δ_r	Sensor resolution in range.
δ	The Kronecker delta function; an indicator function.
$\zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}\|\boldsymbol{x}_t^m\right)$	The measurement probability density for model m at scan t.
$\zeta_t^{mp}\left(\boldsymbol{z}_{tr}^{(x)}\|\boldsymbol{x}_t^m\right)$	The measurement probability density for homothetic component p of model m at scan t.
κ^{mp}	The scalar multiplier for the standard homothetic model for component p of model m.
λ	A Lagrangian multiplier.
μ	The set of all dynamic model indices for OTHR velocity unwrapping.

μ_t	The set of dynamic model indices at scan t.
μ_{tr}	The dynamic model index for measurement r at scan t.
μ_t^m	The dynamic model index for model m.
ν_t	The detection threshold at scan t.
ν_{tr}^m	The innovation for the rth measurement and model m at scan t. The difference between the predicted measurement and measurement $\boldsymbol{z}_{tr}$.
$\mathbf{\Pi}$	The set of all assignment priors for the batch.
$\mathbf{\Pi}_t$	The set of assignment priors for all models at scan t.
π_t^m	The assignment prior for model m at scan t.
ρ	The set of all homothetic mixing proportions.
ρ_t	The set of homothetic mixing proportions for all models at scan t.
ρ_t^{mp}	The mixing proportion for homothetic model p for model m at scan t.
σ_t^m	The assignment prior for clutter model m, given the measurement is due to clutter.
τ_t	The time at which scan t is observed.
$\phi_0^m\left(d_0^m\right)$	The prior probability mass function for the assignment state of model m.
$\phi_t^m\left(d_t^m \middle\| d_{t-1}^m\right)$	The evolution probability mass function (transition matrix) for the assignment state of model m at scan t
$\psi_0^m\left(\boldsymbol{x}_0^m\right)$	The prior probability density function for the state of model m.
$\psi_t^m\left(\boldsymbol{x}_t^m \middle\| x_{t-1}^m\right)$	The evolution probability density function for model m at scan t.

Publications

1. S J Davey, D A Gray, S B Colegrove, and R L Streit, *Simultaneous clutter parameterisation and tracking for skywave OTHR via the PMHT*, Workshop on Signal Processing Applications (Brisbane, Australia), December 2000.

2. S J Davey, D A Gray, and S B Colegrove, *Clutter characterisation for over the horizon radar with the PMHT*, Proceedings of the 4th International Conference on Information Fusion (Montreal, Canada), August 2001.

3. S J Davey, D A Gray, and R L Streit, *Incorporating classifications in the PMHT*, Defence Applications of Signal Processing (Adelaide, Australia), September 2001.

4. S J Davey and D A Gray, *Sensitivity of the PMHT-c algorithm to classifier model mismatch*, International Conference on Optimisation Techniques and Applications (Hong Kong), December 2001.

5. S J Davey and D A Gray, *A comparison of track initiation methods with the PMHT*, Information Decision and Control (Adelaide, Australia), February 2002.

6. S J Davey, D A Gray, and S B Colegrove, *A hidden markov model for track initiation with the PMHT*, Proceedings of the 5th International Conference on Information Fusion (Annapolis, Maryland, USA), July 2002.

7. S J Davey, S B Colegrove, and D A Gray, *A comparison of track initiation with PDAF and PMHT*, Radar 2002 (Edinburgh, UK), October 2002.

8. S J Davey, D A Gray, and R L Streit, *Tracking, association and classification - a combined PMHT approach*, Digital Signal Processing **12** (2002), no. 2, 372–382.

9. S B Colegrove and S J Davey, *PDAF with multiple clutter regions and target models*, IEEE transactions on Aerospace and Electronic Systems **39** (2003), no. 1, 110–124.

10. S J Davey and D A Gray, *The PMHT with hysteresis*, to appear in the Proceedings of the 6th International Conference on Information Fusion (Cairns, Australia), July 2003.

11. S B Colegrove, B Cheung, and S J Davey, *Tracking system performance assessment*, to appear in the Proceedings of the 6th International Conference on Information Fusion (Cairns, Australia), July 2003.

12. S B Colegrove, S J Davey, and B Cheung, *PDAF versus PMHT performance on OTHR data*, to appear in Radar (Adelaide, Australia), September 2003.

Chapter 1

Introduction

The term *tracking* is used to describe approaches used where the goal is to learn about an environment through the application of statistical models to corrupted and ambiguous data. This environment is a dynamic enigma, and it is usually trends within the model that provide information of interest. Tracking techniques can be applied to various problems, such as analysing stock prices, biomedical monitoring, mobile telecommunications, and remote sensing. This thesis is concerned with the last of these applications, the study of which is referred to as *target tracking* or *multi-target tracking* depending on the complexity of the problem at hand.

There is a variety of different types of sensors for which target tracking is applied. *Active* sensors observe distant objects by illuminating them with an energy source and measuring the reflected energy. Alternatively, *passive* sensors observe objects by measuring characteristic emissions of the object (such as engine noise or active sensor transmissions). Also, these sensors operate in different media, and use different types of radiated energy. Examples include underwater and atmospheric propagation, using electromagnetic and sonic energy. While these considerations have significant influence on the difficulty of the target tracking problem, and infer niceties that affect implementation, they do not alter the techniques used for the problems' solution. In all permutations, the sensor is fundamentally a device that measures incident energy over a spatial region, and the tracker is an algorithm that seeks to locate and characterise objects of interest in this region by modelling the temporal variation of this energy.

Solution of the multi-target tracking problem requires the simultaneous completion of two tasks: *estimation* and *data association*. Estimation is the task of finding the best model parameters to describe the observed data. The method used to complete this task is generally a function of the assumed model, resulting in a compromise between model fidelity and ease of model parameter optimisation. It is intuitive that independent objects in the sensor field of view should be represented by independent components in the data model. However, the sensor measurements (namely the observed incident energy, or a statistic of it) do not identify which object caused them. If data from one source is mistakenly used in the parameter optimisation of a component representing a different source object, then that optimisation becomes degraded. The task of assigning data to the components of the data model is data association. There are many different approaches for data association and this is generally the distinguishing feature that gives rise to different tracking algorithms.

This thesis is primarily concerned with a particular tracking algorithm called the Probabilistic Multi-Hypothesis Tracker (PMHT). This algorithm is a relatively new competitor among more established rivals, first introduced by Streit and Luginbuhl in 1995 [SL95].

The advantage offered by the PMHT is that the complexity of the algorithm is linear with time and with the number of components in the data model (the number of objects in the sensor field of view). This makes it realisable without compromising approximations, and gives the possibility of analysing data over a temporal batch, potentially increasing sensitivity and accuracy. The PMHT also uses a mixture paradigm which makes the use of sophisticated models simpler. Such models are often used for manoeuvring target tracking and in difficult interference conditions.

1.1 Motivation

A common shortcoming of most tracking approaches is that they suffer from a combinatorial growth in algorithmic complexity with the number of targets and the number of measurements. This growth is also exponential with time if batch processing is used. The reason for the complexity problem is that standard tracking algorithms assume a measurement model that allows, at most, one measurement per target. This infers a dependency between measurements, and the resulting assignment problem is NP complete. The explosion of computation requirements makes it impractical to implement such algorithms without making approximations that may degrade performance. In contrast, the PMHT has an algorithmic complexity that grows only linearly with these data size parameters. This makes the PMHT an attractive option for multi-target tracking applications.

The Jindalee Facility at Alice Springs (JFAS) is a skywave Over the Horizon Radar (OTHR) that provides wide area surveillance of Australia's Northern approaches. This sensor was first developed by the Defence Science and Technology Organisation (DSTO) and its continued enhancement is a research focus for Intelligence, Surveillance and Reconnaissance Division (ISRD). The current tracking algorithm used for JFAS is called the Unified Probabilistic Data Association Filter (UPDAF) [Col99, CD03] and is a single target tracking algorithm. This means that the algorithm assumes that the association of measurements with tracks can be performed independently for each track. This is a coarse approximation of the type described above, that allows linear complexity with the number of targets, at the cost of estimation accuracy when targets are close to one another. However, since JFAS is a surveillance sensor, timely algorithm performance is crucial, and estimation performance is secondary to execution time.

The single target approximation is valid provided that the targets are sufficiently separated that there is no contention for measurements. This is not always the case. In OTHR, the sensor resolution is much coarser than line of sight microwave radar (typically tens of kilometres). This increases the distance between targets at which the single target approximation breaks down. Also, skywave OTHR relies on propagation by refraction through the ionosphere, an area of charged particles in the atmosphere. This is a multipath medium, and often these paths may be closely spaced, due to the sensing geometry. Without a highly accurate ionospheric model, the tracking algorithm must treat each path as an independent target. Thus, apparent closely spaced multi-target scenarios may arise even when only one target is present due to the multi-path medium.

The need for a multi-target tracking algorithm that was capable of real time operation motivated the investigation of the PMHT. However, the PMHT is a young algorithm, and extensions have been required to produce an operationally practical tracking algorithm. The need for these extensions prompted the research described in this thesis.

1.2 Overview of Multi-target Tracking

Multi-Target Tracking, as described previously, is the problem of monitoring dynamic objects of interest in the field of view of a sensor. This is done by applying a state space model. At discrete intervals, referred to as *scans*, the sensor collects an image of the received energy over the field of view. According to standard practice, this image is put through localisation and thresholding, resulting in a set of point measurements. The set of measurements at scan t is denoted $\mathbf{Z}_t$. The state of the mth target at scan t is denoted by $\boldsymbol{x}_t^m$. The tracking problem is then to find the optimal estimate for $\boldsymbol{x}_t^m$, for all targets (all values of m), given all available information. Often the tracker is required to operate in real-time, that is the state at scan t should be estimated immediately that the scan is received. Under this requirement, the tracker has available measurements from scan 1 to t, namely $\mathbf{Z}_1, \dots \mathbf{Z}_t$. Thus the task of tracking in this case is to evaluate $P\left(\boldsymbol{x}_t^m | \mathbf{Z}_1, \dots \mathbf{Z}_t\right)$. Alternatively, the tracker may have a historical batch of data available, in which case there are measurements from scan 1 up to some $T \geq t$. In this case, the tracker is able to use future measurements to estimate the state, and the tracking task is to evaluate $P\left(\boldsymbol{x}_t^m | \mathbf{Z}_1, \dots \mathbf{Z}_T\right)$. The optimal estimator is then obtained from this probability density function (pdf), according to the optimality criterion.

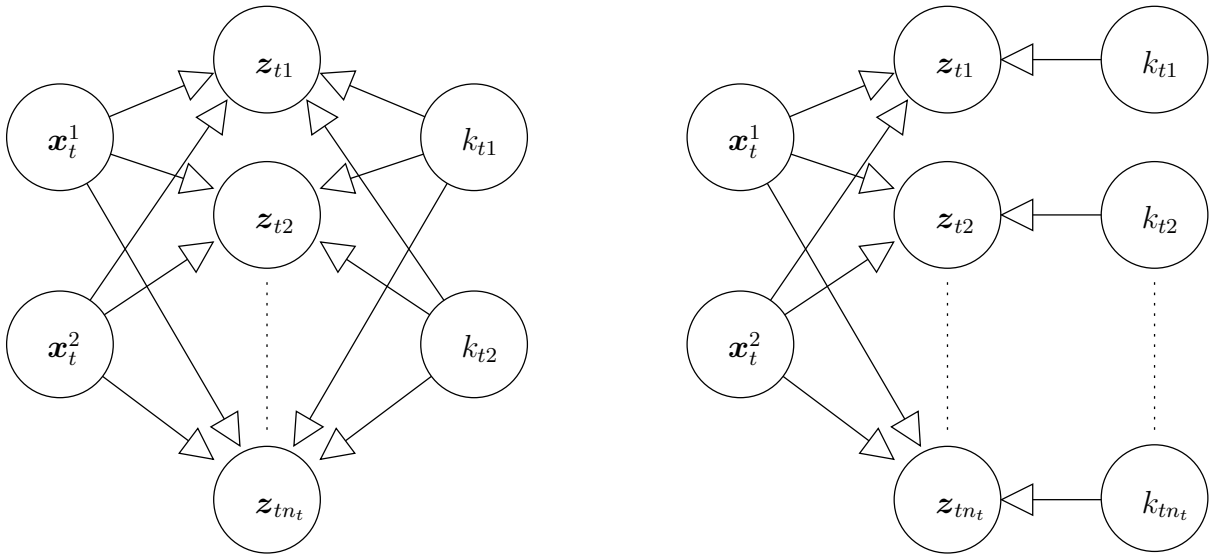
The difficulty in this task is that there are multiple targets present, the detector makes false detections, and targets are not always detected. So, it is not obvious to the tracker which measurements from the data available are caused by target m and which are due to other sources, such as other targets or various false detection processes referred to as *clutter*. The true assignment of measurements at scan t to targets and clutter is denoted as $\mathbf{K}_t$. If $\mathbf{K}_t$ were known, then the state estimation problem would be relatively easy, and could be solved using standard estimation techniques. However, $\mathbf{K}_t$ is unknown, and resolving this uncertainty is the data association problem, a key part of tracking algorithms.

1.2.1 PMHT Measurement Model

The fundamental difference between the PMHT algorithm and other tracking approaches is the assumed measurement model. Under the standard model, prior processing is assumed to extract sufficient statistics of the sensor data that essentially correspond to observations of the location of scatterers in the sensor field of view. It is assumed that this part of the measurement process produces at most one observation per scatterer. Thus, the standard assumption is that at most one observation can be due to a target track. This makes the track to observation association process dependent because the assignment of one observation may alter the possible assignment options for the next.

This is not the case under the PMHT model. The PMHT instead assumes that the true assignment of measurements is an independent random process with an unknown prior probability mass function (pmf). The result of this assumption is that the track to observation association is independent for different observations. This independence is what admits a reduced computational complexity for the PMHT.

The difference between the assumed measurement process for the standard tracking paradigm and the PMHT is highlighted by the Bayesian Inference Networks (BINs) shown in figure 1.1. In the BIN, each random variable is represented as a circle, and directed lines linking the circles indicate the dependence of one variable upon another. There are n_t different measurements in scan t, $\boldsymbol{z}_{t1}, \dots \boldsymbol{z}_{tn_t}$, and each of them is dependent on the model states and the assignment indices.



(a) standard measurement model (b) PMHT measurement model

Figure 1.1: Measurement Model BINs

Under the standard measurement model, there is an assignment index for each target model, indicating which measurement, if any, is the one due to that target. Unless merged measurements are allowed, the assignment indices of different target models are not allowed to indicate the same measurement. Thus, all of the measurements are dependent on the same indices, and all of the measurements need to be taken into account in the estimation of these indices.

In contrast, under the PMHT model, there is an independent index for every measurement. Each of these indices is an independent realisation of an underlying random process. Since each index only affects one measurement, only that measurement needs to be used to estimate the index. So, the PMHT model allows the algorithm to deal with each measurement independently, and this makes the assignment problem much less compuationally taxing.

1.3 Thesis Overview

The thesis deals with extensions of the Probabilistic Multi-Hypothesis Tracker. These extensions expand the tracking algorithm to accommodate a broader range of realistic applications.

> **The first key contribution of this thesis is the extension of the PMHT algorithm to incorporate augmented measurements that are treated as observations of the assignments.**

A key feature of the PMHT algorithm is that it assumes that the assignment of each measurement is an independent realisation of an underlying random process. When the probability mass of this process is unknown, then the PMHT is able to estimate it. However, it requires the limiting constraint that the probability mass is fixed or time independent.

The second key contribution of this thesis is the generalisation of the PMHT measurement model to assume a measurement assignment prior that is a random process which evolves according to an arbitrary discrete Markov Chain.

An important problem in practical tracking filters is automated track decisions. Over time, new targets enter the sensor surveillance region, and old targets leave. It is desirable for a practical filter to automatically initiate new tracks and terminate old tracks in such cases. The PMHT assumes that the number of measurement models is fixed and known.

The third key contribution of this thesis is the extension of the PMHT algorithm to include methods for automatic track initiation and termination. These methods include approaches related to model order estimation, and the use of a Hidden Markov Model to describe track quality.

In order to demonstrate the performance of the above PMHT enhancements, the PMHT algorithm was applied to the skywave OTHR tracking problem and tested using recorded radar data from the JFAS radar.

The fourth key contribution of this thesis is the application of the PMHT algorithm to tracking of data recorded from the JFAS Over the Horizon Radar.

The structure of the thesis is now described. The numbered references correspond to the publications listed on page xxv.

Chapter 2 presents a survey of relevant research in multi-target tracking. In particular, the various methods for data association are described and existing methods for automatic track decision making and augmented measurement data are reviewed. The chapter also gives a thorough summary of advances made to the original PMHT algorithm which improve estimation performance and provide solutions for the problems of manoeuvering targets and clutter.

Chapter 3 defines the multitarget tracking problem in clutter. The Kalman Filter and the Probabilistic Multi-Hypothesis Tracker are reviewed.

Chapter 4 derives a method for incorporating classification measurements into the PMHT framework. The resulting algorithm is referred to as the PMHT-c [3], [8].

Contribution: The development of an enhanced PMHT algorithm incorporating classification measurements, including the estimation of the unknown probability mass function of these measurements, the PMHT-c.

The performance of the PMHT-c for classification measurements of varying accuracy is investigated through simulation. The degradation in performance experienced when the probability mass of the classification measurements is estimated, rather than known, is examined. The sensitivity of the PMHT-c algorithm to a mismatch between the assumed probability mass and the true probability mass is also analysed by simulation [4].

Contribution: Analysis of the sensitivity of the PMHT-c to an incorrectly assumed probability mass for the classification measurements, and the algorithm's ability to adaptively estimate this probability mass.

Chapter 5 presents a generalised assignment model for the PMHT, referred to as the PMHT with Hysteresis. Under this model, the assignment prior probability has a discrete state (a Bayesian hyperparameter)which evolves according to a Markov Chain with an arbitrary state space and statistics. Thus, the assignment state is a Hidden Markov Model (HMM). Two PMHT variants are derived by treating the assignment state as missing information, and by estimating the assignment state [10].

Contribution: The development of two PMHT algorithms applicable to problems where the assignment prior varies smoothly with time. These algorithms utilise a discrete state space model for the assignment prior. The first, PMHT-ym, treats the assignment state as missing data and calculates its probability using the HMM Smoother, whereas the second, PMHT-ye, estimates the assignment state sequence using the Viterbi Algorithm.

Chapter 6 considers the problem of automated track initiation and termination with the PMHT. To provide automated track initiation, automatic initialisation must be performed. A scheme is presented based on a generalised version of the homothetic PMHT. This method uses the innovation covariance matrix as a means for automatic inflation of the measurement variance, thus assisting initialisation by smoothing the objective function at early iterations.

Contribution: An initialisation scheme for the PMHT based on modelling the measurement process as a mixture of the true measurement process and a process defined by the innovation covariance matrix.

Different approaches for automatic track initiation are then considered, each based on the formation of candidate tracks. The candidate tracks are assigned a quality statistic and this statistic is used to accept or reject the candidates. Model Order Estimation techniques are used to derive a candidate quality measure. A Hidden Markov Model approach is also developed by applying the PMHT with Hysteresis to the track initiation problem. The *visibility*[1] model used for initiation with the Integrated Probabilistic Data Association Filter (IPDA) is identified as a special case of the Hysteresis model, and thus the PMHT-ye and PMHT-ym are used to provide a measure of track quality integrated with the state estimation [6].

Contribution: Two algorithms for initiating tracks with the PMHT, utilising a Hidden Markov Model for track quality, analogous to the IPDA approach to track initiation.

Simulated statistics of the various candidate quality measures are used to produce estimated Receiver Operating Characteristic (ROC) curves for the track initiation decision [5]. These ROC curves are used to analyse the discrimination of false candidates and valid candidates for each initiation approach.

[1]This approach also referred to as *perceivability*, *observability* and *track existence* by different authors

Contribution: Comparison of the effectiveness of different track quality measures in the discrimination of false and valid tracks, through use of a Receiver Operating Characteristic curve for track initiation.

In chapter 7, an implementation of the PMHT for over the horizon radar is developed. The PMHT-c algorithm is incorporated for clutter density parameterisation [1]. The effectiveness of the PMHT-c for this application is compared with a direct feature tracker which bypasses the intermediate classification stage [2].

Contribution: Application of the PMHT-c to clutter density estimation for over the horizon radar.

A solution for the particular initialisation problems for Over the Horizon Radar ambiguous doppler measurements is developed, and the initiation schemes of chapter 6 are tested on recorded radar data from the JFAS radar.

Contribution: Application if the PMHT-c, initialisation, and initiation techniques to recorded radar data from the JFAS over the horizon radar.

Chapter8 presents a comparison of the performance of the PMHT and PDAF for track initiation. ROC curves are used to analyse the false and valid candidate discrimination for simulated data [7] and for data recorded from the JFAS radar. The PMHT performance on radar data is compared with the current JFAS tracking algorithm, which is an enhanced PDAF tracker [9]. A preliminary comparison of other features of the PMHT and PDAF algorithms is found in [12] following the technique in [11] but is not included in this thesis.

Contribution: Comparison of the track initiation performance of the PMHT and the PDAF on simulations and on recorded radar data.

Chapter 9 presents a summary of the thesis and completes with conclusions.

Chapter 2

Background

MULTI-TARGET tracking is fundamentally a problem of state estimation confounded by uncertainty in the observation process. The result of this uncertainty is that tracking algorithms must be able to simultaneously solve two very different problems. Firstly, the tracking algorithm must be able to assign observations to each target track. This process is often referred to as Data Association and is in essence an integer programming problem.

Given a collection of observations associated together, the tracking algorithm must secondly be able to estimate the state of the target. If the algorithm is provided with a batch of measurements consisting of observations at different times, then the estimator is commonly referred to as a smoother. A smoothing algorithm uses observations from the past and the future to estimate the current target state. If the state must be estimated in a time recursive fashion (so that the only data available is the observation from the current time and previous observations) then the estimator is commonly referred to as a filter. It is generally possible to derive a filter from the smoothing algorithm.

2.1 State Estimation

Given a particular collection of data, it is often possible to determine several optimal state estimates based on different optimality conditions. Usually, such optimality conditions can be stated in terms of the probability density function (pdf) of the states given the observations. Thus, the problem of state estimation can be formulated as the problem of calculating the probability density of the state.

Under general conditions, it is not possible to obtain a closed form solution to the problem of state estimation. However, there are some special cases where the solution is known. If the target state is a discrete variable, then it can be estimated using the Hidden Markov Model (HMM) Smoother [JR86]. The HMM Smoother is a finite length algorithm with dimension equal to the dimension of the state space. The HMM Smoother is generally not used in multi-target tracking because the target state (typically position and velocity) is continuous. It is possible to discretise the target state space using a sampling grid, but such an approach is an approximation, and will usually lead to a very large grid size since the smoother must have reasonable precision over a large area.

The second case where a closed form estimator exists is when the system is linear and the random elements are Gaussian random variables. Under these conditions, the optimal estimator is the Kalman Smoother [Kal60, KB61]. The Kalman Smoother is a finite length algorithm based on a recursion of the mean and covariance of the state probability density

function. Many target tracking algorithms use the Kalman Smoother or its filter form for state estimation [BSL95, Bla86]. However, most tracking problems are not linear, and the Kalman Smoother cannot be used directly. Instead, an approximate solution to the problem is achieved by linearising around a point and solving for the linearised system with a Kalman Smoother. This approach is referred to as the Extended Kalman Smoother and is the heart of many practical target tracking algorithms. The Kalman filter is usually specified as update equations for the mean and covariance of the target state probability density function. However, an equivalent filter can be written using a recursion for the information matrix (the inverse of the covariance) and this version is referred to as the information filter. There are numerous books that discuss the Kalman filter in detail, including [AM79, BSF88, CC99].

There also exist certain classes of non-linear problems where a closed form solution exist. An example of this is the Benes Filter [Ben81, FBR02]. However, these special non-linear classes of problem are not relevant to target tracking.

A recent approach to non-linear non-Gaussian estimation is referred to as the particle filter. The particle filter is a numerical approximation to the optimal non-linear non-Gaussian solution based on Monte Carlo integration techniques [DDFG01]. The particle filter uses a finite number of samples to represent the state probability density function and uses the exact non-linear system equations to propagate these samples. Since the filter is able to use the exact system rather than a functional approximation, the particle filter is also useful for problems containing constraints on the state space [Cha00, GR01].

2.2 Data Association

When targets are closely spaced, or when the sensor produces false detections, it becomes difficult to discern which measurement belongs to which target. Data Association is the process by which an algorithm assigns measurements to targets. In general, the assignment of measurements to targets can be *hard* or *soft*. Hard assignment is where the data association approach makes a decision about which measurement (if any) is due to a particular target and assigns that measurement to the target (and not others). The target state estimate (i.e. the track) is updated assuming that the assigned measurement is the correct measurement for this target. Soft assignment is where more than one measurement is assigned to each target with a certain probability. Rather than choose a single measurement to update the track, the track is updated using many possible assignments and the collection of updated states is combined using the assignment probabilities, usually in a Bayesian framework. If the data association is performed over a batch of data, it may be possible for the tracking algorithm to change the measurement assignment based on future data.

A data association approach is referred to as being either *single-target* or *multi-target*. A single-target association approach considers each target track in isolation and ignores all other tracks when assigning measurements. Single-target association inherently assumes that the assignment of measurements is independent for different tracks, which is generally not a valid assumption. Single-target association is used to simplify the assignment problem, but may cause significant performance degradation if the targets are closely spaced. In contrast, multi-target association algorithms jointly assign measurements to many tracks simultaneously. To avoid impractical computational requirements, it is usually necessary to partition the tracks into *clusters* that have little effect from other tracks outside the cluster. This is an approximation made for the sake of processing time, and

may again cause performance degradation if the tracks outside the cluster are too close to the cluster.

Data Association is described in detail in many tracking texts, for example [BSF88, BSL95, Bla86, BP99].

2.2.1 Nearest Neighbour

The simplest form of data association is nearest neighbour association [BSL95]. Using nearest neighbour, the observation that is closest to the predicted measurement for a track is assumed to be the correct measurement. Nearest Neighbour is a single-target, hard assignment rule.

Since not all measurement components will have the same accuracy (and some may be coupled) the distance is usually normalised using the target measurement covariance or the innovation covariance. When this is done, the *nearest* measurement is also the measurement with the highest likelihood given the current state estimate.

If the target is not detected on a particular scan, then the nearest neighbour will always be the wrong measurement to assign to the track. If the false measurement assigned to the track is distant from it, then updating the track with it will cause the track to diverge from the target trajectory. To avoid this, a validation gate may be defined. The validation gate determines a region over which it is acceptable to assign measurements to the track. Usually, this takes the form of a maximum distance to the predicted target measurement. Such a validation gate defines an ellipsoid in the measurement space outside of which assignment is not allowed. Measurements inside the validation gate are called validated measurements. The modified nearest neighbour rule is then to assign the closest validated measurement to the predicted target measurement.

Nearest neighbour association tends to perform poorly in cluttered environments where tracks are easily seduced by false detections [BSL95].

2.2.2 Track Split

On any scan where there is more than one measurement inside the validation gate for a track, there exists several possible assignment hypotheses. Either the target was not detected at all, or it caused one of the validated measurements. The nearest neighbour approach chooses one of these hypotheses - that the nearest measurement was due to the target. However, this may not necessarily be the correct hypothesis. For example, if the target manoeuvres, then the target measurement may be distant from the measurement predicted by the track. In such a case, there is a high probability that a false detection will be closer to the track than the true target detection, and the nearest neighbour approach will fail. One way of addressing this issue is to split the track into several tracks, each of which chooses a different assignment hypothesis. There now exists a separate track for each hypothesis, and the assignment decision can be deferred until a future scan. This approach is known as Track Split. The track split approach is a single-target hard association rule. More detailed descriptions of the track split approach can be found in [BSF88, BSL95, Bla86, BP99].

As the number of scans received increases, the number of tracks in the system will grow exponentially. For example, if we receive two validated measurements per scan, the number of tracks will double with every scan. To deal with this, a track scoring system is used to discard the least acceptable tracks and keep the number of tracks manageable. This is typically done by accumulating the squared innovations (the innovation

is the difference between the observed measurement and the predicted measurement). The accumulated squared innovations are proportional to the log of the likelihood of the measurement sequence, under Gaussian measurement statistics.

Track split is an ad hoc approach designed to allow hard assignment while still hedging bets over which measurement to use for that assignment. The track split association method has two major deficiencies. Firstly, it performs single-target association. If the scene contains closely spaced targets, then single-target association becomes confused and will perform poorly. Secondly, it produces a continual supply of new tracks from a single target. Because track split treats each of these spawned tracks independently, it is inevitable that the algorithm will produce duplicate tracks following the same target returns. Such duplicate tracks are redundant, and unless dealt with will ultimately consume the finite processor resources. To deal with redundant tracks, additional rules must be developed which themselves hinder tracking of closely spaced targets.

2.2.3 Multi-Hypothesis Tracker

The multi-hypothesis tracker (MHT) [Rei77, Rei79] is a data association formalism used to assign measurements over a batch of data to multiple tracks. Suppose we have a batch of T scans each containing n_t measurements, and M tracks. An association hypothesis is defined as an assignment of all of the batch measurements to tracks, so that no more than one measurement is assigned to each track from each scan, and no measurement is assigned to more than one track. Each association hypothesis represents a possible hard assignment of the measurements. The MHT finds the best hypothesis by enumerating all possible hypotheses and ranking them based on a scoring metric, typically the measurement likelihood. Since the MHT enumerates all assignment permutations, it is guaranteed to find the optimal hard assignment for the given scoring metric.

To achieve optimal performance on a continuous stream of measurements, the MHT must retain all the hypotheses and their associated scores. This quickly becomes impractical, because the number of hypotheses grows exponentially with time and combinatorially with the number of targets. To make the computation of the hypotheses and their scores feasible, it becomes necessary to approximate the MHT solution by reducing the number of hypotheses retained by the algorithm. Clustering (see discussion above) will greatly reduce the number of hypotheses, but generally more drastic measures are required with MHT. These amount to merging or discarding low scoring hypotheses. The process of deleting low scoring hypotheses is referred to as *pruning*, due to the intuitive representation of the hypotheses as a tree. Each node on the tree represents an assignment decision and the various branches from the node correspond to the different choices of assignment.

An alterative implementation of the MHT is referred to as the track-orientated MHT [Kur90, BP99]. In this method, the hypotheses are not retained over time, but reformed as each new scan is received. [BP99] contains a detailed explanation of MHT including the discussion of various implementation issues involved in producing a practical system.

2.2.4 Viterbi Algorithm

The Viterbi Algorithm is a linear programming approach usually used to find the optimal state sequence for a discrete Markov random process [Vit67, FJ73]. The algorithm is an efficient way of enumerating all possible state sequences which retains only a single sequence leading to each possible state per scan. This reduced sequence set can be achieved because of the Markov property of the state. For each possible state, $\boldsymbol{x}_t$, at scan t, there

are many possible sequences of previous states that may have lead to $\boldsymbol{x}_t$. However, due to the Markov property of the state, the future evolution does not depend on these previous state values. This means that the score of all sequences through $\boldsymbol{x}_t$ is the sum of the score before t and the score after t. Since we are interested in the best score, we need only keep that sequence leading to $\boldsymbol{x}_t$ with the best score - the score after t can be optimised independently.

It is possible to apply the Viterbi algorithm to target state estimation by discretising the state space, however this is usually undesirable. Instead, the algorithm has been used to perform data association [PL97]. Rather than a sequence of states, the algorithm is used to estimate a sequence of assignments. This approach can be seen as an optimal Multiple Hypothesis pruning method in the sense that it guarantees that the optimal sequence is never discarded. The algorithm of [PL97] also includes a model for automatic track initiation and termination.

The main shortcoming of the Viterbi approach is that the number of hypotheses still blows out with the number of tracks and this makes a multitarget implementation of the algorithm infeasible without further pruning. The algorithm used in [PL97] is only for single targets. [GMF02] showed that

2.2.5 Assignment Techniques

The Multi-Hypothesis Tracker is a brute-force approach to solving an integer assignment problem. It works by enumerating all possible assignment hypotheses and choosing the best hypothesis, based on a scoring method. However, there are many other approaches for solving the integer assignment problem. In the tracking field, these approaches are collectively referred to as assignment techniques. Assignment techniques were first applied to the target tracking problem in [Mor77], where they were used for data association and track initiation. A survey of assignment techniques for multi-target tracking can be found in [PPK00].

The measurement association problem (for a single scan), with M targets and n_t measurements, can be couched as a constrained optimisation of the form

$$\min \sum_{r=0}^{n_t} \sum_{m=0}^{M} c_{mr} \chi_{mr}, \tag{2.1}$$

where c_{mr} is the cost associated with assigning measurement r with track m and χ_{mr} is an indicator function taking the value zero or unity when measurement r is assigned to track m. The track denoted $m = 0$ is used to model false detections and the measurement denoted $r = 0$ is a dummy measurement used to model undetected targets. The χ_{mr} are constrained such that

$$\sum_{m=0}^{M} \chi_{mr} = 1 \qquad r = 1, \ldots, n, \tag{2.2}$$

$$\sum_{r=0}^{n_t} \chi_{mr} = 1 \qquad m = 1, \ldots, M. \tag{2.3}$$

These constraints ensure that each measurement is assigned to a single track, and that each track is assigned exactly one measurement (possibly the dummy measurement, $r = 0$). Note that there are no constraints on the dummy measurement or dummy track. This optimisation problem is referred to as 2-D assignment. If the costs are chosen to be the

logarithm of the measurement likelihood for each measurement and track, then the optimisation is a maximum likelihood assignment approach. This approach for multi-target tracking has been used as early as [Mor77] where the author repeats the optimisation for different numbers of tracks to solve the track initiation problem. There exist several algorithms for efficiently solving this optimisation, such as the auction algorithm [Ber79] and the Jonker, Volgenant, and Castanon (JVC) algorithm [JV87]. There are also algorithms that can determine the second and third (and so on) best assignment solutions, rather than simply the single best. It may be useful to retain a number of assignment options so that future data can be used to improve the decision. The Auction algorithm is generally regarded as the best approach in sparse problems [Ber88, PPK00, BP99]. Most tracking problems will be sparse, since tracks will tend to validate only a small proportion of the total collection of received measurements.

2.2.5.1 Auction

The auction algorithm [Ber79, Ber88] is an $O(n^3C)$ complexity algorithm where C is the range of the cost coefficients. The complexity depends on C because it dictates how quickly the bidding process will converge. As its name suggests, the auction algorithm sees tracks vying for contested measurements by making *bids*. Each track bids for measurements until the price becomes too high and the auction finishes. The auction algorithm is guaranteed to reach within a prescribed amount of the optimal cost, dependent on the overbidding parameter, ϵ, which is explained below. The auction algorithm deals with a maximisation problem obtained by making the price coefficients the negative of the cost coefficients, c_{mr}.

Initially, all measurements are assigned to the clutter track ($m = 0$) with an associated price, $p_r = -c_{0r}$. The tracks place a value on each measurement, given by $v_{mr} = -c_{mr} - p_r$. If this value is positive, then making the corresponding track-measurement assignment would improve the overall cost. A value of zero is given to the dummy measurement, i.e. $v_{m0} = 0$. Each track determines which measurement represents the highest value - if no measurement has a positive value, then the track is assigned to the dummy measurement indicating that the target was not detected. Those tracks with a nonzero value for at least one measurement bid for the measurement with the highest value. The tracks bid an amount γ_{mr} defined by

$$\gamma_{mr} = \max\{v_{mr}\} - \max_{r' \neq r}\{v_{mr'}\} + \epsilon, \tag{2.4}$$

where ϵ is an overbidding constant which is used to prevent ties in the event that two tracks have the same cost relating to a particular measurement. This bid is the value of the best measurement less the value of the second best measurement, plus the overbidding factor. The track which makes the highest bid for a measurement is assigned to that measurement, and all other tracks are unassigned. The prices of all assigned measurements are incremented by the bid value, $p_r = p_r + \max\{\gamma_{mr}\}$.

The unassigned tracks repeat the bidding process after redetermining the value of each measurement using the new prices. The assigned tracks do not bid again. Again, the highest bidder is assigned to a measurement, and any track previously assigned to that measurement becomes unassigned. The bidding process is continued until there are no more unassigned tracks. The use of the overbidding constant, ϵ, causes tracks to bid until measurements become slightly overpriced, and guarantees that the algorithm will eventually stop. Because the bidding increases the price of attractive measurements, the

less attractive measurements (i.e. those with a higher cost) will eventually offer better value to unassigned tracks.

The auction can equivalently be implemented with measurements bidding for tracks, except that any number of measurements is allowed to be assigned to the clutter.

2.2.5.2 S-D assignment

The auction and JVC algorithms described above are methods for solving the 2-D assignment problem. This corresponds to associating measurements from a single scan. When presented with a batch of data, the problem becomes more difficult. The general batch problem is referred to as S-D assignment, where $S-1$ scans of measurements are assigned to the tracks. This problem is solved using a technique referred to as Lagrangian relaxation [Fis81]. The S-D problem is an *NP hard* problem and the Lagrangian relaxation technique provides an efficient way of determining an approximate solution. The S-D assignment problem consists of an objective function, analogous to equation (2.1), and S hard constraints of similar form to (2.2) and (2.3). The Lagrangian relaxation works by relaxing one of the S hard constraints and replacing it by a Lagrangian penalty term in the objective function. In this way, the constraint is no longer enforced, but assignments that violate it will be penalised. The new problem is now an $S-1$ dimensional one with a modified objective function. Since there are now fewer constraints, the problem is simpler to solve. The relaxation of the hard constraints can be repeated until the problem is reduced to a 2-D assignment with $S-2$ Lagrangian penalty terms in the modified objective function. This 2-D problem can be solved with the auction algorithm (or others). Since the problem no longer enforces the constraints, it cannot be guaranteed to give the optimal solution. Detailed discussion of S-D assignment can be found in [PPK00] and [BP99].

2.2.6 Bayesian Data Association

Estimators are generally derived using an optimality criterion based on the probability density of the target states given the received measurements. Bayesian Data Association approaches are based on the use of Bayes Rule to simplify the target state probability density function. Let $\mathbf{X}_t$ denote the target states at scan t (possibly for multiple targets). At scan t, a set of n_t measurements, $\mathbf{Z}_t$, is observed by the sensor. The aim is to calculate the density of the $\mathbf{X}_t$ given the measurements from scan t and all earlier scans, and given an initial state density $P(\mathbf{X}_0)$. Let $\mathcal{Z}_1^t$ be the set of all measurements from scan 1 to scan t. This required density is

$$\begin{aligned} P\left(\mathbf{X}_t|\mathcal{Z}_1^t\right) &\equiv P\left(\mathbf{X}_t|\mathbf{Z}_1,\mathbf{Z}_2,\ldots,\mathbf{Z}_t\right) \\ &= \int P\left(\mathbf{X}_t|\mathbf{X}_0,\mathbf{Z}_1,\mathbf{Z}_2,\ldots,\mathbf{Z}_t\right) P\left(\mathbf{X}_0\right)\, \mathrm{d}\mathbf{X}_0. \end{aligned} \tag{2.5}$$

Under the assumption that the target states are independent first order Markov processes the density becomes

$$P\left(\mathbf{X}_t|\mathcal{Z}_1^t\right) = \int P\left(\mathbf{X}_t|\mathbf{X}_{t-1},\mathbf{Z}_t\right) P\left(\mathbf{X}_{t-1}|\mathcal{Z}_1^{t-1}\right)\, \mathrm{d}\mathbf{X}_{t-1}. \tag{2.6}$$

This now provides a recursion for the posterior state density, but the term $P\left(\mathbf{X}_t|\mathbf{X}_{t-1},\mathbf{Z}_t\right)$ is problematic because the assignment of each measurement in $\mathbf{Z}_t$ is unknown. We introduce an index variable $\mathbf{K}_t$ which denotes a particular assignment

hypothesis. Each value of $\mathbf{K}_t$ defines the assignment of all measurements in $\mathbf{Z}_t$ and the domain of $\mathbf{K}_t$ contains all possible assignment hypotheses. Given $\mathbf{K}_t$, the probability of the observations is known. We now write

$$\begin{aligned} P\left(\mathbf{X}_t|\mathbf{Z}_t,\mathbf{X}_{t-1}\right) &= \frac{P\left(\mathbf{X}_t,\mathbf{Z}_t|\mathbf{X}_{t-1}\right)}{P\left(\mathbf{Z}_t|\mathbf{X}_{t-1}\right)} \\ &\propto P\left(\mathbf{X}_t|\mathbf{X}_{t-1}\right)P\left(\mathbf{Z}_t|\mathbf{X}_t\right) \\ &= \sum_{\mathbf{K}_t} P\left(\mathbf{X}_t|\mathbf{X}_{t-1}\right)P\left(\mathbf{Z}_t|\mathbf{X}_t,\mathbf{K}_t\right)P(\mathbf{K}_t). \end{aligned} \tag{2.7}$$

Implementation of (2.7) would yield the optimal association strategy. However, the domain of $\mathbf{K}_t$ grows combinatorially with the number of targets and measurements. Further, (2.7) means that the density $P\left(\mathbf{X}_t|\mathcal{Z}_1^t\right)$ is a mixture (due to the sum). For every mode in $P\left(\mathbf{X}_{t-1}|\mathcal{Z}_1^{t-1}\right)$, (2.7) produces d_t modes in $P\left(\mathbf{X}_t|\mathcal{Z}_1^t\right)$ where d_t is the number of hypotheses (the size of the domain of $\mathbf{K}_t$). This means that the density $P\left(\mathbf{X}_t|\mathcal{Z}_1^t\right)$ is a mixture with a number of components that grows exponentially with time and the rate of growth is combinatorially dependent on the number of targets and measurements. This exponential growth makes it impractical to implement an exact Bayesian solution, so approaches have been developed to approximate it.

2.2.6.1 Gaussian Sum Filter

The Gaussian sum filter [Sal90] is an approximation to the Bayesian solution that uses a fixed (or bounded) number of Gaussian components to represent the target state distribution at each scan. When new data is received, the number of components in the updated state distribution is inflated. The components in the updated distribution are ranked and then merged together until the resulting approximation contains the desired number of components. The merged components are chosen in such a way as to preserve the moments of the distribution. The Gaussian sum filter ensures that the approximate distribution is a Gaussian mixture with a reduced number of components, which enables the use of a Kalman filter for state estimation. The Gaussian sum filter is also referred to as a mixture reduction algorithm because it works by reducing the number of components in the Gaussian mixture that is the pdf of the target state.

There are two different merging techniques used in [Sal90]. The first is termed *joining* and selects pairs of mixture components that have the closest means. Here the distance measure is normalised by the overall mixture covariance. The algorithm defines an acceptable degree of distortion, and continues to merge pairs until it reaches this limit. If the number of components is still larger than the pre-defined maximum number retained, then further pairs are joined until the number components is low enough. The second merging technique is termed *clustering* and collects together components that have low mixing proportion, merging them with more dominant components. The clustering method may merge together several components at once (rather than simply pairs). [Sal90] demonstrates that the clustering method gives better results, but with increased computational expense.

It is apparent that the Gaussian Sum approach is a numerical approximation to the probability density function. In fact, we can view the Gaussian as a kernel function and the mixture components as coefficients of a multi-resolution approximation to the true density function. As with all approximations, the performance of the filter will be acceptable if the approximations are within tolerable error. By adjusting the maximum

number of components, we have a trade-off between computation requirements and the degree of approximation used by the filter.

A particular special class of Gaussian Sum filters is referred to as the *Generalised Pseudo Bayes* (GPB) filters. The GPB filter of order N, referred to as GPBN, is a Gaussian sum filter that merges together mixture components at a fixed lag of N scans. The GPB terminology is also used for filters that contain a number of switching dynamics models which leads to a similar growth in the number of components in the state probability density. The advantage of a GPB filter is that the merging of components can be pre-computed analytically since it is fixed. The drawback is that the filter may waste resources updating components that make negligible contribution, or may merge two significant components because they arose from an ambiguous assignment at an earlier stage. Since the merging is analytically derived and hard coded into the filter, it is not possible to change the number of retained components at a later stage.

2.2.6.2 Probabilistic Data Association Filter

A special case of the Gaussian Sum Filter is the Probabilistic Data Association Filter (PDA or PDAF) [BST75]. The PDA is a popular association algorithm in practical tracking systems because of its simplicity and speed. The PDA is also the Generalised Pseudo Bayes filter of order 1. In the PDA, the target state density is approximated by a single Gaussian component. This is equivalent to representing the distribution by its first two moments. At each scan, the number of components in the target density grows according to the number of validated measurements received. The PDA then makes the assumption that the resulting mixture can be approximated by its first two moments, and produces an updated Gaussian pdf approximation. This approach has been demonstrated to significantly improve tracking performance over nearest neighbour association [BST75]. However, if the probability density of the target state is strongly multi-modal, then the PDA is clearly throwing away information, and the performance can be adversely affected.

The main advantage of the PDA approach is that it can be easily implemented in an efficient way. Since the target pdf is approximated at each scan by a single Gaussian, it is possible to reorganize the target density update so that the PDA is realised by a single Kalman filter. The filter is fed a centroid measurement, formed by taking the weighted average of all the validated measurements, and the filtered covariance is inflated an amount dependent on the scatter of the validated measurements.

The PDA is a single-target algorithm, but it has been extended to a multi-target version, referred to as the Joint-PDA (JPDA) [FBSS83]. Like the multi-hypothesis tracker, the complexity of the JPDA grows exponentially with the number of targets and clustering is usually required to achieve a physically practical algorithm. Various methods have been proposed to provide efficient JPDA algorithms including [ZB95] and [DC01]. Detailed discussion of the PDA and JPDA can be found in [BSL95].

2.2.7 Probabilistic Multi-Hypothesis Tracker

The Probabilistic Multi-Hypothesis Tracker (PMHT) [SL95] represents a very different approach to association. In the standard Bayesian approaches, or in the various hard assignment techniques, the true assignment of measurements to targets is viewed as an unknown parameter of the problem. This parameter is subject to constraints, namely each target is allowed to form at most one measurement. In the PMHT framework, the assignment of each measurement is treated as a random process, with an associated prob-

ability mass function. It is usual to treat the measurements as independent realisations of a random process, but the PMHT extends this to treat their sources as realisations of a random assignment process. This means that the hard constraints of the standard association approaches are violated. The constraints represent a form of dependence between the association of different measurements. The PMHT asserts that the association of measurements and sources is independent for different measurements. The problem of association and estimation then becomes a joint estimation process of two sets of random variables: continuous target states, and discrete measurement assignments.

The joint estimation of target states and random assignments is no easier than the standard MHT approach. However, the actual values of the assignments are not of any particular interest in most applications. The goal of the tracker is to estimate the target states, and the assignment problem merely arises as a complication. It is possible to treat the tracking problem as an estimation problem with incomplete data. The incomplete data consists of the measurements, whereas complete data would also include the measurement assignments. This form of problem can be solved using the Expectation-Maximisation (EM) algorithm [DLR77].

The EM algorithm provides a method for estimating the target states without estimating the assignments. This is achieved by maximising the conditional expectation over the assignments of the joint log likelihood of the states, assignments and measurements. This expectation is referred to as the EM auxiliary function. A detailed exposition of the EM algorithm can be found in [MK97].

The resulting algorithm is an iterative procedure that alternates between a data association step and a state estimation step. In the association step, the probability that each measurement is due to each target is calculated, using the state estimate from the previous iteration. In the estimation step, a new state estimate is produced by finding the maximum likelihood estimate of the state given measurements weighted by their association probabilities. The process is initialised by choosing a first state estimate and is halted when the auxiliary function converges.

The main advantage that the PMHT has over other association techniques is a linear complexity in the number of targets and in the number of scans used in a batch. These are a result of the probabilistic model used for the measurement assignments. However, this probabilistic model does not match realistic sensors that perform peak detection [DC99] to attempt to produce a single measurement per target. For such systems, the one-to-one assignment approach of other filters is more appropriate. In [RWS99] the modified measurement model was shown to degrade the Cramer-Rao Lower Bound (CRLB) for estimator variance. While this is important, it does not reflect the true algorithm performance since PMHT is an optimal algorithm (given the initial assumptions) and should achieve its CRLB whereas algorithms based on the standard measurement model are suboptimal.

The PMHT originally proposed by Streit and Luginbuhl in [SL95] is a multitarget filter for updating tracks in the absence of clutter. The algorithm assumes that the number of targets is known, and that it is possible to initialise the track states.

The problem of initialisation was addressed by the introduction of a *homothetic* measurement model in [RWS95a]. The homothetic model uses multiple measurement processes with a common mean and different covariances. Using the homothetic measurement model reduces the PMHT sensitivity to track initialisation. Alternatively, methods commonly used in other numerical optimisation algorithms can be employed. One such approach is covariance inflation which uses large covariance values for initial iterations, and progressively reduces them to the desired covariances for the final iterations. Such a

scheme was used by Kreig [Kre98]. More recently, [RW01a] introduced the *Turbo PMHT* which exploits a technique similar to *turbo coding* used in communication coding. The advantage of the turbo PMHT approach is that it introduces the track innovation covariance, which is a measure of the current estimate uncertainty. [RW01a] demonstrates significant improvement with the turbo PMHT over the homothetic PMHT.

Clutter has been incorporated into the PMHT in [RWS95a] and [DH97]. The incorporation of clutter is done by adding an extra model to the track list to represent clutter. This model has a different measurement process to the models representing targets that may be uniform, or at least diffuse. The homothetic measurement model also improves tracking performance in clutter by giving preference to measurements closer to the predicted track position. In [LSW01], the clutter pdf is modelled as an arbitrary mixture of uniform components, each with a different spatial support. The clutter pdf is adapted with the observed data

The PMHT algorithm has also been generalised to deal with measurements from more than one sensor [KG97b, Kre98].

2.2.7.1 Existing comparisons of the PMHT with other approaches

The first performance comparison for the PMHT was published by Rago, Willett and Streit in [RWS95a] and presents a comparison of the algorithm with the Joint PDA tracking algorithm. The Joint PDA algorithm is a variant of the PDA used to track several targets simultaneously. This paper presents only one simple simulated scenario with two tracks moving in parallel with constant speed, under varying clutter density. The results of the comparison indicate that the PMHT offers superior performance in this example. However, the comparison does not address the issue of track initiation, that is, starting new tracks from a collection of observations. For the comparison in [RWS95a], all tracks are initialised with the correct target states.

Dunham and Hutchins presented a PMHT formulated for tracking in clutter in [DH97] and [HD97]. These papers compare the PMHT with the Multi Hypothesis Tracking algorithm and the PDA. However, the PMHT is implemented as a batch algorithm over an extended data set (30 time scans) while the PDA and MHT are recursive algorithms. The comparison gives the expected result that a batch smoother produces state estimates with a smaller estimation error than a recursive filter. The comparison uses a strict initialisation criterion, which requires five consecutive target detections to start a track. Such a condition would be rarely met under a modest probability of detection. Disturbingly, the authors comment in [HD97] that the PMHT performed very poorly with less stringent initialisation conditions. The comparison also makes no mention of the false track performance of the approaches, which is important in any analysis of tracking in clutter. The comparison measures performance using RMS estimation error, which is an inappropriate metric for initiation.

Willett, Ruan and Streit presented a PMHT for manoeuvring targets in [WRS98a] that was compared with the Interacting Multiple Model PDA (IMM-PDA), a variant of the PDA used for manoeuvring targets. Both algorithms use a Markov process to cope with manoeuvring targets. The target is modelled as having two process states, one with high process noise (the manoeuvring state) and one with low process noise (the non-manoeuvring state). The IMM-PDA uses a PDF approximation (two moments) for each model. The PMHT estimated the state at each time instant using the forwards-backwards Markov chain approach. The comparison used a simulated target in the presence of clutter. This comparison is again favourable for the PMHT but it does not address initialisation.

In [RW01a], various PMHT algorithms are compared with the PDA in the problem of track maintenance. The number of trials where each filter manages to follow the target over a batch of data is observed for various clutter conditions. The turbo PMHT is demonstrated to give approximately the same performance as PDA, which is better than the homothetic approach. The problem of track initiation is not considered and tracks are initialised with the correct state.

2.2.8 Histogram PMHT

The two main problems with the PMHT are its sensitivity to initialisation, and the violation of the one to one assignment constraint. The PMHT is sensitive to initialisation because it is a hill climbing numerical optimisation approach and the objective function (the EM auxiliary function) has a problematic topology. Each measurement in a scan forms a local maximum in the auxiliary function and the overall global maximum is only achieved by temporal correlation of the in scan maxima. This is because the point detection process treats all measurements equally. The one to one assignment constraint is also a consequence of point detection and is really a fallacy. Typically, a sensor forms bins which form a discrete map of the received power in the measurement space. The backscattered power from targets will spread into more than one bin, either because of the physical target size compared with the bin resolution, or because of the point spread function of the sensor. This distributed target response is then put into a point detection process often referred to as *peak detection*, or *peak picking* [DC99], that attempts to produce an output of one measurement per distributed target response. It is this peak detection step that imposes the one to one assignment constraint. The PMHT fails to enforce the constraint because it models the sensor measurements as a mixture model, not a point detection process. So, it can be seen that the PMHT is not an incorrect model of the tracking problem, rather it fails to model a step in the sensor processing that the user chooses for the purpose of data reduction (and because other tracking algorithms require it).

The solution to the two main failings of the PMHT is then to remove the peak detection process. In [Str00b] Streit developed an algorithm referred to as *Histogram* PMHT (HPMHT) that tracks multiple targets in clutter using the entire collection of sensor bins. The naming of the algorithm stems from the theoretical development, which models the received radar image as a multi-dimensional histogram of a mixture. The histogram model is produced by quantising the bin powers at a particular resolution, $\hbar$, and treating each of the quantised values as a shot count. The artificial shots are independent of each other conditioned on the mixture model. The theoretical development of the algorithm concludes by taking the limit as the resolution becomes infinitely fine ($\hbar \to 0$) and the original sensor data are recovered. The HPMHT algorithm is robust to poor initialisation, and demonstrates good tracking on low power targets, even in the vicinity of other stronger targets [SGW01]. Since the HPMHT processes the entire sensor image, the processing requirements are significantly higher than the standard PMHT, and data storage may be an important issue, particularly with a stepped scanning sensor that may dwell on many different spatial regions. However, like the PMHT, the growth in computation remains linear with the number of targets and with the data size (batch length and size of each individual scan).

The HPMHT has also been extended to handle the case where targets have an extended signature in one or more of the sensor dimensions [Str01]. An example of this is passive sensing where the sensor measures the angle of arrival of signals and their spec-

tra. These spectra carry information about the identity of the target and can be used in classification [CP01]. This extension is called the Spectral HPMHT and provides a method for integrated tracking and classification. It also improves multi-target tracking performance by allowing for a superposition spectrum when targets become unresolvable.

2.2.9 Markov-chain Monte Carlo

Markov-chain Monte Carlo (MCMC) is a powerful numerical technique for solving difficult statistical problems, particularly involving integrals. The aim is to represent the target pdf by a collection of samples from it. This same technique is used in the particle filter [DDFG01] to solve nonlinear state dynamics or measurement processes. MCMC uses a numerical approximation to model the target pdf whereas the gaussian sum type approaches use a functional approximation. Each sample of the target state density is updated by a particular measurement and the resulting samples form an approximation of the updated state density. This approach was used in [HLP01, HLP02] to solve a highly non-linear multitarget bearings only tracking problem.

2.2.10 Probabilistic Least Squares Tracker

The Probabilistic Least Squares Tracker [Kre98] (PLST) is a multi-target soft association approach based on the principle of least squares optimisation. Whereas the PMHT and PDA approach the measurement association problem from a Bayesian probability theoretic viewpoint, the PLST takes more of a pragmatic stance. PLST assigns each measurement to each track with a certain weight. Rather than the probability of that measurement having been caused by the particular track, the PLST weight is chosen so as to optimise an iteratively re-weighted least squares problem. The formulation is much like the S-D assignment problem, but with the χ_{mr} indicator function replaced by a real valued α_{mr}. The α_{mr} must satisfy similar constraints to the χ_{mr}, namely they must sum to unity.

The optimal solution for the α_{mr} is a function of the unknown target states. The PLST solution requires an iterative approach, alternating between state estimation and optimal weighting. Thus the PLST is an example of an iteratively re-weighted least squares algorithm. Unlike the assignment approaches, the real valued α_{mr} results in an algorithm with linear complexity in the number of targets and in time.

The resulting PLST algorithm is functionally very similar to the PMHT but it uses a normalised distance ratio to assign measurements rather than a probability ratio. Under the usual assumptions of linear Gaussian statistics, the probability ratio of the PMHT tends strongly to extreme values (zero or unity) whereas the PLST distance ratio tends to be more variable [Kre98]. A detailed comparison of PLST and PMHT was performed in [KG97a, Kre98]. The PLST was found to be robust to mismodelling of the system covariances because it does not rely on assumed pdfs. However, the PMHT tended to give better steady state performance when the system statistics are known.

2.3 Tracking with Augmented Measurements

The tracking problem was introduced as two separate tasks: association and estimation. The tracking techniques described above make use of the same observations for both tasks. In general, there may be additional information available to the tracker which can

improve the measurement association, but has no bearing on the target state estimation. This information can be used to augment the measurements and give better tracking performance through improved assignment. The strategy of improved assignment through augmenting measurements was first proposed in [NSC84] where the *rank* of measurements was used to improved target to clutter discrimination. Usually a radar transmits a series of pulses and targets may be detected on each pulse with a particular probability, which depends on the target signal to noise ratio (SNR). The rank of the measurement is defined as the number of pulses in a scan where a detection is flagged at the corresponding radar cell. A certain minimum rank is required by the detector to declare a measurement. We would generally expect false measurements to have a lower rank than valid target measurements, hence the rank is a useful measurement feature for association. Another feature commonly used to improve discrimination of target measurements and false measurements is the measurement amplitude [Col87, Col89, LBS90, LBS01].

The above examples demonstrate additional information available to the tracker from alternative processing of the sensor data already used for state estimation. However, there may also be other sensors on the tracking platform that are able to provide information to the tracking system. An example of this is an airborne platform fitted with radar and ESM (electronic support measures) sensors. The ESM is a passive sensor which provides information about the emitters at various azimuths. This information can be incorporated into the tracker to improve association. [CP01] gives an algorithm for jointly tracking on ESM and radar data.

The algorithms presented in [NSC84, Col87, LBS90] are all examples of *augmented PDA* algorithms. A brief discussion of the augmented PDA can be found in [BSL95]. Any feature can be incorporated into the augmented PDA algorithm, so long as the designer can provide the probability density function of the feature for targets and clutter (or at least a reasonable model for it). The examples above demonstrate both real and integer valued features. In general, the augmented measurement set will contain continuous and discrete information. If the features provide discrimination between targets and clutter, or between different targets, then association will be improved.

A special case of this problem is where the tracker receives classification information. Here the additional measurements come from an enumerated set of classes and the probability of observing a particular class depends on intrinsic properties of the target.

2.3.1 Augmented Measurement Vector

The simplest way to incorporate additional measurement information into the tracking filter is to augment the measurement vector and assume fixed models for the distribution of the augmented measurements. The kinematic observations, such as range, azimuth and radial velocity, are dependent on the target state, but the augmented information depends on fixed distributions for targets and clutter. This is the method used in [NSC84] to incorporate measurement rank. [LBS90, LBS01] also use fixed models for target and clutter amplitude. The drawback with this approach is that most practical tracking problems contain different types of targets, which will have varying statistics. For example, the distribution of measurement rank depends on the probability of detecting the target on a particular pulse. This probability of detection is a function of the SNR which in turn is a function of radar cross section. Similarly, the observed amplitude also depends on radar cross section.

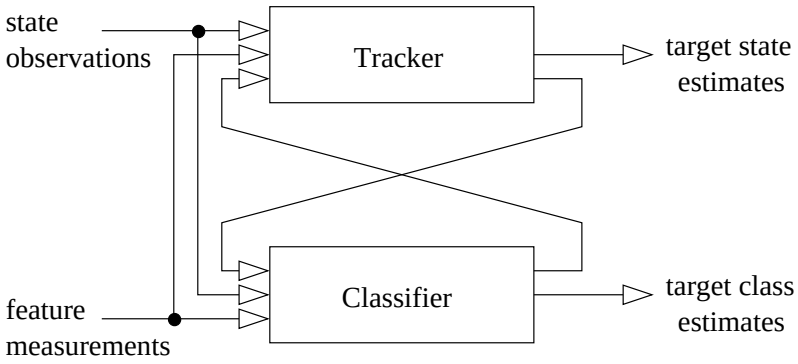


Figure 2.1: Joint Tracking and Classification approach of Challa and Pulford

2.3.2 Augmented State Vector

In order to account for targets with different characteristics, it is necessary to augment the state vector with parameters of the feature distribution. In this manner, the feature distribution for each target can be estimated. [Col87, Col89] is an example of this approach where the amplitude of each target is modelled as constant with Gaussian measurement error and estimated through the Kalman filter. The difficulty with this is that most features will not have Gaussian observation errors and their dynamics are unlikely to be linear. This means that the Kalman Filter is not really a suitable estimator. In [Col99] this problem is dealt with by using a non-linear transform to produce an error distribution which is somewhat similar to a Gaussian. This is analogous to pre-whitening a receiver spectrum before applying a matched filter.

2.3.3 Joint Classification and Tracking

A typical case where augmented measurements are provided to the tracker is where the sensor or platform is capable of providing estimated classifications. These classifications might be treated as augmented measurements, such as in [RCD97], or alternatively the tracker might be used to provide features to the classifier. Since the target class and its dynamics are linked (or else the above approaches would be pointless) it would be desirable to jointly perform both tasks. This is essentially an augmented state vector paradigm, however, this restrains the form of the classifier. The tracker requires an estimate of the probability density function of the data. Rather than perform the two tasks together, Challa and Pulford [CP01] proposed a coupled architecture where the tracker and classifier feedback into each other and each task is performed iteratively. This coupled architecture is shown in figure 2.1.

A more general discussion of integrated tracking and classification is found in [Dru99, Dru01]. These articles provide more of a philosophical discussion of how the problem might be approached rather than demonstrating a particular algorithm such as in [CP01].

2.4 Track Initiation

In the general multi-target tracking problem, the sensor receives data from an unknown number of targets at unknown locations within the surveillance region. Most tracking algorithms make use of data association techniques to assign measurements to existing target tracks and then refine the target tracks using those measurements. This assumes

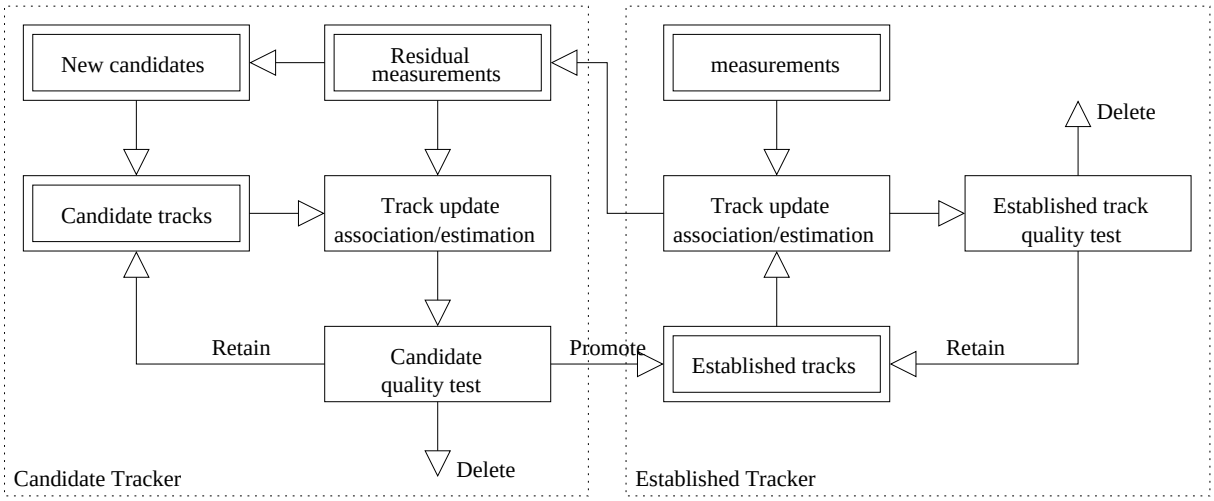


Figure 2.2: General initiation flow diagram

that the algorithm already has a state estimate of each target from a previous time and that no new targets appear. In a practical situation this will not be the case and the problem of initiating tracks on new targets arises.

In general, track initiation approaches can be expressed in a candidate track promotion form. Figure 2.2 shows a flow diagram for track initiation in the candidate track form. New tracks are formed based on unassigned measurements and these new tracks are added to a collection of candidate tracks. The candidate tracks are updated using a tracking filter and then put through a promotion test. Those tracks that pass the test are promoted to established tracks that can be shown to the sensor user. Those that do not are put through a second test, and are discarded if they fail it. The set of established tracks may interact with the candidate update perhaps through censoring the measurement list to prevent formation of multiple tracks on the same target.

The different methods used for track initiation can be viewed as different tests for promotion and retention of candidate tracks. There are broadly two approaches to this testing: the test statistic can be a byproduct of the tracking filter, or the filter can be specifically designed to provide a test statistic. The advantage of using byproducts of the filter is that the filter itself need not be altered in any way. This simplifies the design and makes it possible to retrofit automatic testing to existing filters. Unfortunately, this is a little like bolting new functions onto existing solutions. Better performance is obtainable by using a filter that provides a test statistic by design.

The candidate tracks can be started by initialising a track on a single measurement, or by using several measurements from consecutive scans to provide a more accurate initialisation. It is common to initialise tracks using measurements from two consecutive scans. This is referred to as *two point differencing*. A discussion of track initialisation can be found in [Str95].

[HLB97, LHB96] present comparisons of the performance of different track initiation methods.

The simplest initiation schemes are those that make use of tracking byproducts, since these require no modification of the tracking algorithm. Approaches of this type will be reviewed first.

2.4.1 M of N initiation

A commonly used test for track initiation is referred to as M of N initiation. Under this rule, candidate tracks must be assigned M measurements in N scans. Under low clutter conditions, the rule may require M validated measurements in N scans, a slightly easier condition. Because the rule is so simple, it is possible to generate analytic expressions for quantities such as the probability of promoting false and valid tracks, which are useful in quantifying tracking performance [BSCL90]. A problem with this approach is that all tracks are quite likely to have validated measurements when the clutter is dense. In addition, there is a limited range of promotion thresholds available, since M must be an integer no more than N. In order to achieve a very low probability of promoting false candidates, it may be necessary to use a large N which delays track initiation.

2.4.2 Accumulated Log-Likelihood

The first algorithms for track initiation [Sit64] used the measurement innovations to score the track. If the target statistics are gaussian, then the sum of the squared innovations is proportional to the negative of the log likelihood of the measurement sequence. The measurement innovations are the difference between the predicted measurement and the observed measurement. When used in a time recursive manner, the squared innovations are accumulated with time, producing a test statistic proportional to the negative of the accumulated log-likelihood of the measurements. Valid target tracks are expected to have a low scatter, and the sum of their squared innovations should be low. False tracks are expected to give high innovations. This is the same test that is used in the track split filter to discard excess tracks. If the target statistics are known (or assumed), then it is possible to determine analytically the probability of observing a particular innovation given the measurement is due to the target. This can be used to determine the probability of promoting valid tracks for a given threshold on the accumulated innovations [Sit64].

2.4.3 Model Order Estimation Techniques

The track initiation and termination problem is actually a model order estimation problem. This is particularly clear if we view tracking as parameter estimation of a dynamic mixture where the number of components in the mixture is unknown. Methods such as Maximum Likelihood and Least Squared Error fail in the model order estimation problem because the model fit to the data can always be improved by added further components. The pioneering work in model order estimation [Aka74, Sch78, Ris78] resulted in a number of different tests to determine the model order of a data sequence. Of these, the Minimum Description Length (MDL) of [Ris78] is generally accepted to be the best. This is because the test due to [Aka74] is dependent on the number of measurements of the system. When more observations are received, it will produce a higher model order estimate. Duplicating the data to artificially increase the number of measurements produces the same behaviour. The test due to [Sch78] has a penalty term that scales with the number of measurements, but the MDL also contains a penalty term dependent on the model complexity (not all order M models are the same).

The MDL test was derived by an information theoretic approach designed to find the statistical autoregressive moving average (ARMA) model with the minimum number of coefficients that fit the observed data. The test chooses the model that maximises the data likelihood penalised by a term dependent on the model order and by a term dependent on

the model complexity. Essentially, a higher order model is accepted only if it increases the likelihood by more than a prescribed amount. This test can be directly applied to track initiation, by promoting candidate tracks only when they sufficiently increase the data likelihood. An approach analogous to [Sch78] was used in [VLK00, VL00] to estimate the number of components in a (static) Gaussian mixture. The technique introduced a new component and used partial EM to optimise the state of the new mixture component. Partial EM updates a subset of the states while keeping the others fixed. The improvement in the EM auxiliary function, namely the increase in the log-likelihood, was tested against a threshold. This technique could be extended to dynamic mixtures, which is precisely the EM problem solved by PMHT.

A model order estimation approach such as MDL can be considered as an initiation scheme which uses byproducts of tracking. It is quite similar to the cumulative likelihood approach, however it uses the joint likelihood of all tracks, rather than considering each track in isolation. MDL requires the maximum data likelihood given the model, which is obtained by using the maximum likelihood estimator for the state. This is what the tracker does anyway, so MDL for tracking can be implemented as a candidate test after tracking, and the tracker is not modified.

2.4.4 Hidden Markov Model

When the tracker updates a false candidate track there is a mismatch between the data and the assumed model. The model assumes that there is a target present that produces a sequence of measurements, when in fact there is none. The initiation methods which rely on tracker byproducts attempt to detect this mismatch condition and thus reject false candidates. An alternative is to change the model so that it is capable of modelling the false track condition. Candidates would then be accepted or rejected based on the estimated state of the model. This is a fundamentally different approach because it alters the target state model and integrates the candidate quality estimation with the target state estimation.

An extended target state model was first proposed by Colegrove et al in [CDA86]. The extended target state contained a binary scoring parameter, referred to as the target *observability*. The target observability was a first order Markov process that was used to model the validity of candidate tracks with the PDA.

The term observability was not adopted by other authors due to the existing state modelling definition. However, the Markov model for track validity was. In [Col99] Colegrove adopted the alternative term *visibility* which will be used in this thesis. The same approach to initiation with the PDA has been referred to as *track existence* in [MES94] and *perceivability* in [LL01a, LL01b]. This PDA variation is often referred to as the Integrated PDA (IPDA), a term first used by Mušicki et al. in [MES94]. The Markov visibility model has also been used with other association algorithms such as in [RA00, CVW02]. With this assumed model, there are a number of different possible approaches to solve for the probability mass of the target visibility, these will now be discussed.

2.4.4.1 Generalised Pseudo Bayes

The generalised pseudo Bayes (GPB) filtering approach is a fixed lag method for estimating processes with switching modes [Blo84]. This switching may be in the target model, in the form of manoeuvre models, or in the form of the visibility model for target existence.

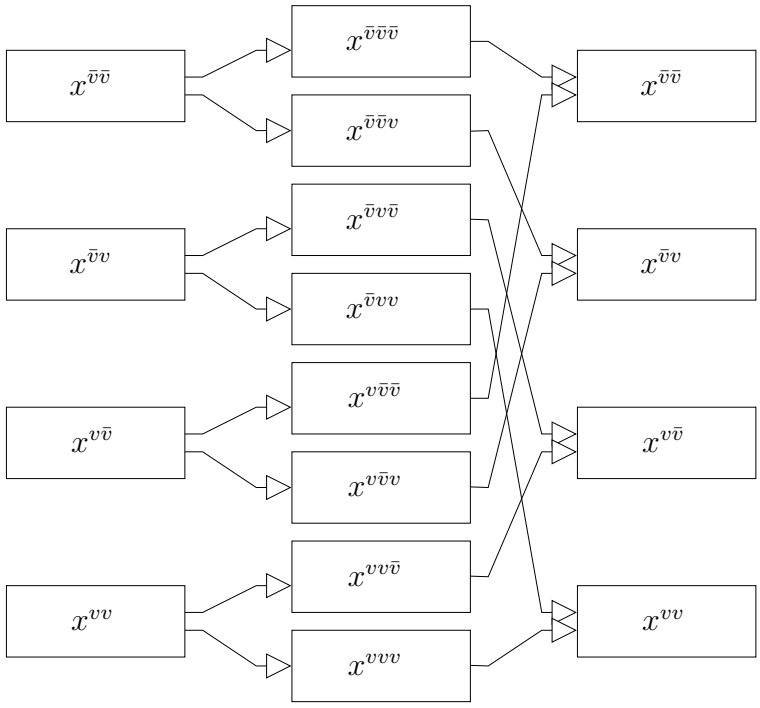


Figure 2.3: Generalised Pseudo Bayes, order 3

As described earlier, it can also be applied to data association. The GPB filter of order N, referred to as GPBN, uses a fixed lag of N scans to estimate the target state. In the visibility case, this is achieved by storing 2^{N-1} state estimates, each corresponding to a different chain of target visibilities. When the new scan is processed, these estimates grow to 2^N since each can either switch visibility state or remain the same. These 2^N state estimates are then combined to form 2^{N-1} estimates corresponding to shifting the beginning point of the sliding batch. Figure 2.3 shows a flow diagram for the GPB3 algorithm. The visibility chains are labelled v or $\bar{v}$ corresponding to whether the target is visible or invisible at that time. For example $x^{\bar{v}\ v}$ is the state estimate corresponding to the hypothesis that the target was invisible on the previous scan, but became visible on this one. Separate filtering must be done for each of the 2^N modes which therefore increases the computational cost of the algorithm.

2.4.4.2 Integrated Probabilistic Data Association

The simplest form of the generalised pseudo Bayes filter is the GPB1 filter. This is the form originally proposed in [CDA86] where the visibility model was introduced. This is the same form used for the trackers in [MES94] and [LL01a]. In [MES94] the name Integrated Probabilistic Data Association was adopted for this filter. The trackers of [Col99],[MES94] and [LL01a] are not identical (in particular they have very different models for clutter) but they all use the same approach for initiation. At each scan, the IPDA forms a single estimate of the target state by summing together the visible and invisible state estimates, weighted by their respective probabilities. The visible state estimate is found by the standard PDA approach: each validated measurement is used to update the state, and these state estimates are combined according to the probability that the particular measurement was caused by the target. The probability that measurement $\boldsymbol{z}_i$ was caused by the target is denoted as β_i and the probability that the target is visible

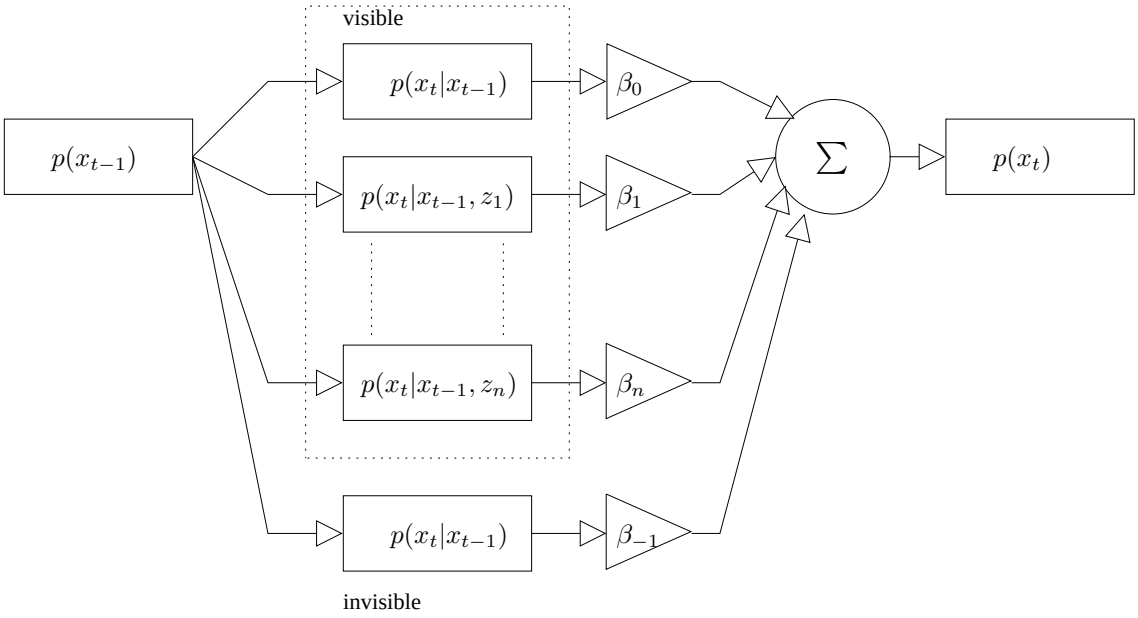


Figure 2.4: IPDA flow diagram

but did not cause any of the validated measurements is denoted as β_0. In addition to these standard PDA terms, there is the invisible target state, and the probability of the target being invisible is denoted as β_{-1}. Figure 2.4 shows a flow diagram of the state estimation when there are n_t validated measurements for the track. In addition to the target state estimate, the filter maintains an estimate of the probability that the target is visible which can be used to make track promotion and retention decisions. This probability of the target being visible also feeds into the mixing proportions which are used to sum the state estimates.

In [CDA86] the IPDA algorithm was demonstrated to give a significant improvement in false track rate over the standard PDA for similar target detection performance. The algorithm also has the advantage of being fairly simple and quick. In practice, most implementations of the IPDA algorithm assume the same dynamic models for visible and invisible targets. A consequence of this is that the flow diagram in figure 2.4 can be factorised, and the algorithm can be implemented using a single Kalman Filter with a synthetic measurement, much like the standard PDA. This measurement is formed by taking the weighted sum of the validated measurements, weighted by the associated β_i. The extra computations required beyond the standard PDA are minimal.

2.4.4.3 Interacting Multiple Model Probabilistic Data Association

The Interacting Multiple Model (IMM) filter [Blo84, BBS88] was first proposed as an approximation to the GPB2 filter. The IMM filter uses mixing of the state estimates to reduce the complexity of the GPB2 filter. A thorough description of the use of IMM-PDA for track initiation is given in [BSCB90]. Other descriptions of the IMM-PDA algorithm for track initiation and manoeuver tracking can be found in [MABSD98] and [BSL95]. The IMM algorithm is an heuristic approach which was originally justified with asymptotic arguments in [Blo84].

Rather than blowing out to four state estimates, the two estimates from the previous scan are mixed together to form predicted visible and invisible states. Each of these pre-

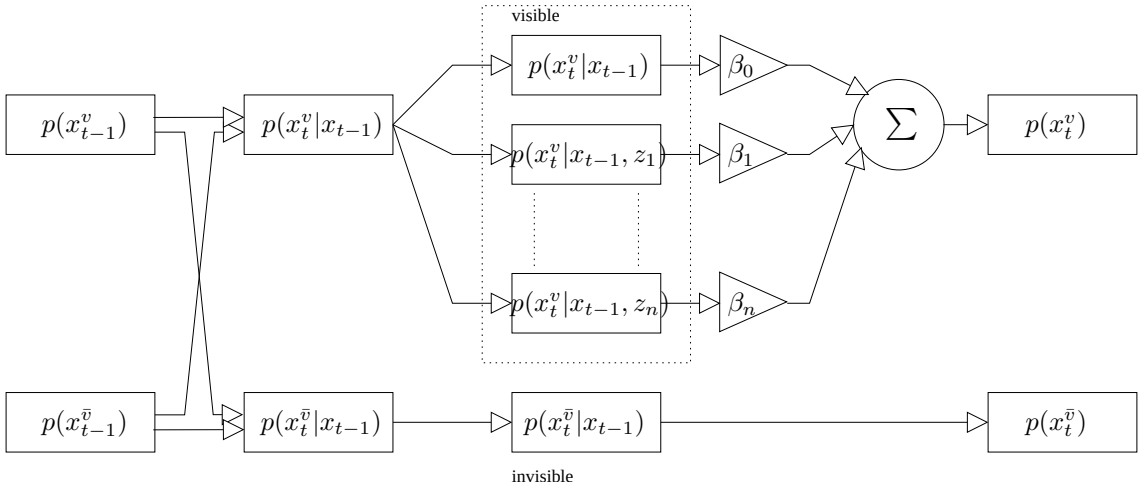


Figure 2.5: IMM flow diagram

dicted states is then updated. The predicted visible state is updated using standard PDA. The predicted invisible state is updated without any measurements (dead reckoning). The structure of the IMM-PDA filter is shown in figure 2.5.

The practical gain in efficiency of the IMM over GPB2 is not as substantial for track initiation as when it is used for manoeuvring target tracking. This is because the invisible model is not updated by measurements. Since the invisible model is not updated by measurements, the GPB2 algorithm only requires 2 Kalman filters (rather than 4) which is the same as the requirements of the IMM filter. [MES94] gives a comparison of IPDA and IMM-PDA where the two algorithms give almost identical performance. [LL01a] also concludes that the added complexity of the IMM algorithm does not provide a performance advantage for initiation.

2.4.5 Radon and Hough Transforms

The optimal strategy to detect a target would be to coherently integrate all of the energy that the sensor received from it to maximise the detector SNR. The difficulty with this is that we do not know the target trajectory. It is not feasible to test all conceivable trajectories, since there will be an infinite number of them (or at least a very large quantity). However, the Radon transform can be used to test all possible linear trajectories through the data. The Radon transform is a mapping from the position space into a parameterised line space. Peaks in the Radon space can be used to detect straight lines in position space. Usually the discrete version, referred to as the Hough transform [DH72, IK88] is used to detect lines in images. The Hough transform has been applied to detection and tracking for radar in [CEW94a, CEW94b, CEW94c] and more recently it has been integrated into the PMHT framework in [LSW01]. A drawback of this approach is that it uses the full data array received by the sensor and requires a number of scans of the same spatial region. This may result in a large memory requirement for a stepped scan sensor.

Chapter 3

Multi-Target Problem Formulation

THIS chapter defines the multi-target tracking problem in clutter and introduces the statistical framework that shall be used in the following chapters. It follows the method used by Streit and Luginbuhl in [SL95] to derive the Probabilistic Multi-Hypothesis Tracker. However, the same framework can also be used to derive other algorithms such as the Probabilistic Data Association Filter. The concept of the observer is reviewed, and the structure of the observer for the case of multi-target tracking in clutter is presented. Particular state models of interest are presented.

The standard PMHT algorithm is then stated, and some of the existing enhancements to it are discussed. Areas where the PMHT may be improved are highlighted, in particular those addressed in later chapters of this thesis.

3.1 Problem Definition

Suppose that a sensor observes a particular surveillance region and reports measurements at irregular intervals, referred to as scans. At each scan, there are multiple measurements. Some of these measurements are caused by signals that are of interest to the sensor operator, and are referred to as target detections. The other measurements may be caused by various undesired interference and noise processes collectively referred to as clutter. Let $\mathbf{Z}$ denote the collection of all measurements from the sensor over a batch from scan 1 to scan T. Let τ_t denote the time at which scan t was collected and n_t denote the number of measurements formed by the sensor in scan t. Assume that $\tau_{t+1} > \tau_t$ for all t. This merely implies that the measurements are ordered in time sequence and that all measurements observed at a particular time are collated in a single scan. Although it is possible to consider a more general problem, where measurements are received possibly out of sequence, this is beyond the scope of this thesis. Define $\mathbf{Z}_t$ to be the vector of all measurements in scan t.

$$\mathbf{Z}_t \equiv [\boldsymbol{z}_{t1}, \boldsymbol{z}_{t2}, \ldots, \boldsymbol{z}_{tn_t}], \tag{3.1}$$

where $\boldsymbol{z}_{tr}$ is the r^{th} measurement in scan t. The set of all measurements $\mathbf{Z}$ can be written as the set of the per scan vectors $\mathbf{Z}_t$.

$$\mathbf{Z} \equiv \{\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_T\}. \tag{3.2}$$

The probability density of the measurements, $P(\mathbf{Z})$, is a mixture of the measurement probability densities due to each of the various sources in the sensor scene, that is, $P(\mathbf{Z})$ is the weighted sum of the measurement probability densities of the sources. The weighting

for each source is referred to as the mixing proportion for that source and the measurement probability for that source is a component of the probability density function of $\mathbf{Z}$. Each of the source measurement probability densities has a known functional form with unknown time varying parameters, called the state. The goal is to find the dynamic mixture model that best describes the probability density function of the measurements $\mathbf{Z}$, namely the states that maximise the probability of the observed measurements. This is a Maximum Likelihood Criterion for estimating the states.

3.2 The Observer

The PMHT observer as introduced by Streit and Luginbuhl in [SL95] consists of two parts. Firstly, there is a collection of continuous states. These states are the dynamic parameters of the components of the measurement mixture density function. Secondly, there is a set of discrete assignments. These assignments label the true source of every measurement in the set $\mathbf{Z}$.

The continuous state part of the observer is composed of a number of models. Each model is used to represent an independent source of sensor measurements. Let the number of target models be M_X and the number of clutter models be M_Y. The total number of dynamic state models is

$$M = M_X + M_Y. \tag{3.3}$$

The distinction between target and clutter models is unnecessary at this stage since both are simply members of the dynamic mixture. The PMHT algorithm is not tied to any specific models for target or clutter behaviour. Instead, models are specified based on the problem at hand, and a particular solution is achieved for those models. The models do not have to occupy the same state space, and it is likely that it will be useful to have different state vectors for different models. For example, the state representation of clutter processes will most likely be different to the state representation of target processes. It is assumed that the dimension of the state vector for each model does not vary over the batch.

Let $\boldsymbol{x}_t^m$ denote the state of model m at time τ_t corresponding to scan t. Suppose there is some time τ_0 at which the probability density function of the state $\boldsymbol{x}_0^m$ is known. In a practical system, this density function may not be known and track initialisation techniques would be used to estimate it.

Define the temporal collection of model states over the batch for model m as

$$\mathbf{X}^m \equiv \left[\boldsymbol{x}_0^m, \boldsymbol{x}_1^m, \ldots, \boldsymbol{x}_T^m\right]. \tag{3.4}$$

The total collection of all states is defined as

$$\mathbf{X} \equiv \left[\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^M\right], \tag{3.5}$$

and the vector of all states at a particular scan is denoted

$$\mathbf{X}_t \equiv \left[\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \ldots, \boldsymbol{x}_t^M\right]. \tag{3.6}$$

Denote the (known) probability density function of the state of model m at time τ_0 as $\psi_0^m\left(\boldsymbol{x}_0^m\right)$. Assume that the evolution of the state of model m is a first order Markov process. That is, given the state of the model at the previous scan $\boldsymbol{x}_{t-1}^m$, the state of the model at the current scan $\boldsymbol{x}_t^m$ is conditionally independent of all prior states $\boldsymbol{x}_0^m \ldots \boldsymbol{x}_{t-2}^m$.

Denote the (known) probability density function of the state evolution as $\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)$. This function is implicitly dependent on the time elapsed between scans t and $t-1$, i.e. $\tau_t - \tau_{t-1}$.

The observer also contains the assignment of each measurement to the model that caused it. For the rth measurement, $\boldsymbol{z}_{tr}$, at scan t there is a corresponding assignment index, k_{tr}, that denotes the model that gave rise to that measurement. The assignment index is an integer in the range $[1, M]$. When the assignment index k_{tr} takes a value in the range $[1, M_Y]$ then this indicates that measurement $\boldsymbol{z}_{tr}$ was caused by clutter model k_{tr}. When it takes a value in the range $[M_Y + 1, M_X + M_Y]$ then this indicates that measurement $\boldsymbol{z}_{tr}$ was caused by target model k_{tr}. Define the set of all assignment indices at scan t as

$$\mathbf{K}_t \equiv [k_{t1}, k_{t2}, \ldots, k_{tn_t}], \tag{3.7}$$

and the set of assignment indices over the whole batch as

$$\mathbf{K} \equiv [\mathbf{K}_0, \mathbf{K}_1, \ldots, \mathbf{K}_T], \tag{3.8}$$

where $\mathbf{K}_0 = \emptyset$, i.e. the empty set, because there are no measurements at time τ_0.

Notice that by defining the association indices $\mathbf{K}$ there is an implicit assumption that each measurement is caused by only one of the models. There is no allowance made for the possibility that more than one model might contribute to a measurement, for example the case of two closely spaced targets that are not resolved by the finite aperture of a radar.

The observer at scan t is then defined as the combination of the model states and the assignment indices for scan t at time τ_t:

$$\mathbf{O}_t \equiv \{\mathbf{X}_t, \mathbf{K}_t\}, \tag{3.9}$$

and the batch observer is defined as the collection of model states and assignment indices over the entire batch:

$$\mathbf{O} \equiv \{\mathbf{X}, \mathbf{K}\} = [\mathbf{O}_0, \mathbf{O}_1, \ldots, \mathbf{O}_T]. \tag{3.10}$$

3.2.1 The Assignment Model

Under the standard assignment model for tracking, each target is able to form, at most, one measurement. It is assumed that the sensor uses a detection process that may not always form a measurement when a target is present. This missed detection may be due to random signal fluctuations, environmental factors, or simply low signal strength at the detector. When the target is detected, it may be distributed through several sensor cells, however the detector is assumed to produce a single measurement from these distributed returns. Each target thus produces either no measurement or exactly one measurement. This implies that no two assignment indices, for different measurements in the same scan, can indicate the same target model. That is, the number of indices which take the value $m \in M_Y \ldots M$ is either zero or one. Mathematically this is represented as:

$$\sum_{r=1}^{n_t} \delta\left(k_{tr} - m\right) \in \{0, 1\} \qquad M_Y + 1 \leq m \leq M, \tag{3.11}$$

where $\delta(.)$ is the Dirac delta, an indicator function that is unity at the origin and zero elsewhere. There is no constraint on the number of measurements assigned to the clutter models.

The constraints (3.11) cause the assignment problem to be difficult to solve and result in algorithms with greater than linear complexity in the number of measurements. This constrained assignment model is used in the Probabilistic Data Association Filter and the Multi-Hypothesis Tracker.

In the PMHT, the assignments are assumed to be independent identically distributed random variables. This assumption conflicts with the constrained assignments used for PDAF and MHT. Because PMHT uses this assumption, it allows each target to produce more than one measurement, and the constraint 3.11 can be violated. Let π_t^m denote the probability that the assignment index k_{tr}, corresponding to the rth measurement at scan t, takes the value m. Since the k_{tr} are identically distributed, π_t^m is the same for all r (r is an arbitrary indexing). Define the following sets of π_t^m:

$$\mathbf{\Pi}_t \equiv \left[\pi_t^1, \pi_t^2, \ldots, \pi_t^M\right], \tag{3.12}$$

$$\mathbf{\Pi} \equiv \left[\mathbf{\Pi}_1, \mathbf{\Pi}_2, \ldots, \mathbf{\Pi}_T\right]. \tag{3.13}$$

3.2.2 The Measurement Process

In addition to the observer, a measurement process is used to link the states and the assignments with the received measurements. The measurement process for each model defines the probability density function of measurements from that model over the surveillance region. The measurement processes are each functions only of the state of the corresponding models. All measurements are assumed to be independent identically distributed random variables drawn from the measurement probability density for the model given by the assignment index corresponding to the measurement. This means that the measurement process is a mixture process and the unconditional probability density of the measurements is a weighted sum of the individual model measurement densities.

Different models may have different measurement processes (indeed they must if the models have different state vectors). The measurement probability density function for model m at scan t is denoted by $\zeta_t^m\left(\boldsymbol{z}|\boldsymbol{x}_t^m\right)$. Note that it has also been implicitly assumed that the measurements at scan t are independent of the states at all other scans when conditioned on the states at scan t. The independence of measurements conditioned on the model states is a common assumption to tracking algorithms. The probability density function of measurement $\boldsymbol{z}_{tr}$ is then:

$$p\left(\boldsymbol{z}_{tr}|\mathbf{X}_t, k_{tr}\right) = \begin{cases} \zeta_t^1\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^1\right) & \text{if } k_{tr} = 1 \\ \zeta_t^2\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^2\right) & \text{if } k_{tr} = 2 \\ \vdots & \vdots \\ \zeta_t^M\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^M\right) & \text{if } k_{tr} = M. \end{cases} \tag{3.14}$$

3.3 Target and Measurement Models

Up to this stage, the tracking problem has been defined using completely general forms for the probability density functions of the target state evolution and the measurement process. Without specifying these processes in terms of statistical models, it is difficult to proceed to a solution for the estimation of their parameters.

One common assumption in many tracking applications is that of linear Gaussian statistics. A reason for the popularity of this model is that the probability density function can be defined using a finite parameter set, namely the mean vector and covariance matrix. Further, the solution to the estimation problem is well known for linear Gaussian statistics. That is, given a set of measurements, it is possible to find the optimal state estimate. The closed form solution for this estimate is the Kalman smoother. The state evolution and measurement models are now examined in more detail, presenting the linear Gaussian forms frequently used.

3.3.1 State Evolution Model

Recall that the target evolution is assumed to be a first order Markov process. Therefore, the most general form of the evolution can be written as

$$\boldsymbol{x}_t^m = \boldsymbol{f}_t^m \left(\boldsymbol{x}_{t-1}^m, \boldsymbol{u}_t^m, \boldsymbol{v}_t^m \right), \tag{3.15}$$

where $\boldsymbol{u}_t^m$ is a control input, $\boldsymbol{v}_t^m$ is random noise (referred to as the process noise), and $\boldsymbol{f}_t^m(\cdot)$ is some function. In the target tracking case, $\boldsymbol{x}_t^m$ generally represents the dynamic state of a non-cooperative target (perhaps with other non-kinematic attributes), so it is necessary to concede at this point that the control input $\boldsymbol{u}_t^m$ is unknown and it will be compensated for by using an artificially inflated noise process. Under the assumption of additive noise, the state model can be rewritten as

$$\boldsymbol{x}_t^m = \boldsymbol{f}_t^m \left(\boldsymbol{x}_{t-1}^m \right) + \boldsymbol{g}_t^m \left(\boldsymbol{v}_t^m \right), \tag{3.16}$$

where the functions $\boldsymbol{f}_t^m(\cdot)$ and $\boldsymbol{g}_t^m(\cdot)$ are arbitrary, and perhaps nonlinear. In many practical applications, such as the tracking of ballistic objects [CKBS01, YBSPD95], it is necessary to use nonlinear functions for $\boldsymbol{f}_t^m(\cdot)$ and $\boldsymbol{g}_t^m(\cdot)$. Nevertheless, in other cases, it is more usual to assume that they are both linear, and further that the noise process $\boldsymbol{v}_t^m$ is Gaussian. This allows the use of the Kalman Filter for state estimation, whereas the nonlinear, non Gaussian case requires an approach such as particle filtering.

Under the linear Gaussian model, the evolution density of the target state is a linear function of the target state at the previous scan corrupted with Gaussian noise. The state evolution is thus represented by

$$\boldsymbol{x}_t^m = \mathsf{F}_t^m \boldsymbol{x}_{t-1}^m + \mathsf{G}_t^m \boldsymbol{v}_t^m, \tag{3.17}$$

where $\boldsymbol{v}_t^m$ is a zero mean Gaussian random vector with covariance Q_t^m and F_t^m and G_t^m are appropriately sized matrices. Equation (3.17) is a general linear state model with additive noise. In the tracking application, the matrices F_t^m, G_t^m and Q_t^m are determined by the equations of motion and the motion model used. For example, a common model is the *almost constant velocity* model. Here, the target velocity is assumed to be constant except for accelerations introduced by the noise process, $\boldsymbol{v}_t^m$. For a one dimensional almost constant velocity model, the state vector consists of one dimensional position and velocity,

$$\boldsymbol{x}_t^m(\text{1D}) = \begin{bmatrix} r \\ \dot{r} \end{bmatrix}, \tag{3.18}$$

the process noise covariance, Q_t^m, is a scalar, and so:

$$\mathsf{F}_t^m(\text{1D}) = \begin{bmatrix} 1 & \tau_t - \tau_{t-1} \\ 0 & 1 \end{bmatrix}, \tag{3.19}$$

$$\mathsf{G}_t^m(1\mathrm{D}) = \begin{bmatrix} \frac{1}{2}(\tau_t - \tau_{t-1})^2 \\ \tau_t - \tau_{t-1} \end{bmatrix}. \tag{3.20}$$

The almost constant velocity model at higher dimensions can be composed simply from the matrices $\mathsf{F}_t^m(1\mathrm{D})$ and $\mathsf{G}_t^m(1\mathrm{D})$. For example, for a three dimensional model Q_t^m is a 3x3 symmetric positive definite matrix and:

$$\mathsf{F}_t^m(3\mathrm{D}) = \begin{bmatrix} \mathsf{F}_t^m(1\mathrm{D}) & 0 & 0 \\ 0 & \mathsf{F}_t^m(1\mathrm{D}) & 0 \\ 0 & 0 & \mathsf{F}_t^m(1\mathrm{D}) \end{bmatrix}, \tag{3.21}$$

$$\mathsf{G}_t^m(3\mathrm{D}) = \begin{bmatrix} \mathsf{G}_t^m(1\mathrm{D}) & 0 & 0 \\ 0 & \mathsf{G}_t^m(1\mathrm{D}) & 0 \\ 0 & 0 & \mathsf{G}_t^m(1\mathrm{D}) \end{bmatrix}. \tag{3.22}$$

assuming that the random variations in speed are independent for each velocity component.

Using (3.17) it is straightforward to show that the covariance of $\boldsymbol{x}_t^m$ given $\boldsymbol{x}_{t-1}^m$ is $\mathsf{G}_t^m\mathsf{Q}_t^m\mathsf{G}_t^{m'}$. It is clear from (3.20) that this covariance of the state evolution is singular. That is,

$$\operatorname{rank}\left\{E\left[\left(\boldsymbol{x}_t^m - \mathsf{F}_t^m\boldsymbol{x}_{t-1}^m\right)\left(\boldsymbol{x}_t^m - \mathsf{F}_t^m\boldsymbol{x}_{t-1}^m\right)^{\mathsf{T}}\right]\right\} = \operatorname{rank}\left\{\mathsf{Q}_t^m\right\} < \dim\left\{\boldsymbol{x}_t^m\right\}. \tag{3.23}$$

This is because the only random elements are the velocity perturbations and there is no random fluctuation of the position, except as an integration of the velocity. This means that the probability density function of the state evolution exists only on a manifold and is zero elsewhere. The position is completely dependent on the velocity fluctuations. While this is unimportant for Kalman Filtering (since the Kalman Filter does not require this inverse) it is important to recognise this in order to write down the proper expression for the evolution density, $\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)$. The only time that the evolution density is required is in the calculation of the PMHT auxiliary function that is used to test convergence. Since the random part of the evolution comes through the noise process, $\boldsymbol{v}_t^m$, the probability evolution density is simply the noise probability density. Under the almost constant velocity model, the process noise produces a deviation in the velocity, so the difference between the target velocity at t and $t-1$ determines the value of the noise, $\boldsymbol{v}_t^m$.

So the state evolution probability density is given by

$$\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) = |2\pi\mathsf{Q}_t^m|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(\boldsymbol{x}_t^m - \boldsymbol{x}_{t-1}^m\right)^{\mathsf{T}}\mathsf{A}^{\mathsf{T}}\mathsf{Q}_t^{m-1}\mathsf{A}\left(\boldsymbol{x}_t^m - \boldsymbol{x}_{t-1}^m\right)\right\}, \tag{3.24}$$

where the matrix A is used to pick out the velocity component of the state, for example

$$\mathsf{A}(3\mathrm{D}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.25}$$

This pdf applies provided the position deviation matches the velocity deviation, otherwise the probability of the state transition is zero.

In addition to the evolution probability density, the estimator requires the initial probability density function of the model states, $\psi_0^m\left(\boldsymbol{x}_0^m\right)$. In general, this function might take any form, but under the assumption of linear Gaussian statistics, $\psi_0^m\left(\boldsymbol{x}_0^m\right)$ is also a Gaussian function. Let the mean and variance of $\boldsymbol{x}_0^m$ be $\bar{\boldsymbol{x}}_0^m$ and P_0^m respectively. The initial state distribution is thus:

$$\psi_0^m\left(\boldsymbol{x}_0^m\right) = |2\pi\mathsf{P}_0^m|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}_0^m - \bar{\boldsymbol{x}}_0^m)^{\mathsf{T}}\mathsf{P}_0^{m-1}\left(\boldsymbol{x}_0^m - \bar{\boldsymbol{x}}_0^m\right)\right\}. \tag{3.26}$$

3.3.2 Measurement Model

Recall that there is a measurement probability density for each model, namely $\zeta_t^m(\boldsymbol{z}|\boldsymbol{x}_t^m)$. In general, this density is the result of the nonlinear measurement process

$$\boldsymbol{z}_{tr} = \boldsymbol{h}_t^m(\boldsymbol{x}_t^m, \boldsymbol{w}_{tr}), \tag{3.27}$$

where the source of measurement $\boldsymbol{z}_{tr}$ is model m as defined by the index k_{tr}, and $\boldsymbol{w}_t^r$ is random noise. Under the assumption of additive noise, and assuming that the noise vector, $\boldsymbol{w}_{tr}$, has the same dimension as the measurement vector, $\boldsymbol{z}_{tr}$, (3.27) becomes

$$\boldsymbol{z}_{tr} = \boldsymbol{h}_t^m(\boldsymbol{x}_t^m) + \boldsymbol{w}_{tr}. \tag{3.28}$$

In almost all tracking applications, the true measurement equation is nonlinear, because the sensor receives observations in a polar coordinate system (such as range, bearing and elevation) but the target state is represented in a Cartesian coordinate system (altitude, north, east) or in a geocentric coordinate system (altitude, latitude, longitude). If the target state is modelled in the radar polar coordinate system, then the state evolution will be nonlinear (since targets do not move at constant range or azimuth rate except in special cases). When the target state occupies a Cartesian coordinate system, and the sensor measurements are in a polar coordinate system, the bearing measurement function becomes an arctangent.

The modelling method for the measurement process depends very much on the degree of nonlinearity. For active sensors that operate at very long range, it is reasonable to assume that targets move at constant velocity in the radar polar coordinates. This makes life simpler, since the whole problem now becomes linear. This approach is appropriate for sensors such as Over the Horizon Radar which typically operate at ranges greater than 1000 km. At intermediate ranges, the nonlinearity cannot be ignored, but it is still not a large problem. Sensors such as active microwave radars typically use a first order Taylor approximation to the nonlinear function, namely the extended Kalman Filter. At close range, or with passive sensors where a range measurement is unavailable, it becomes necessary to use a nonlinear filter such as the particle filter [DDFG01].

Since the following analysis will be mainly concerned with association performance, and not estimation performance, it is possible restrict the problem to linear measurement models. These models are also appropriate for the physical application which will be explored later, namely Over the Horizon Radar.

The linear Gaussian measurement model replaces (3.28) with

$$\boldsymbol{z}_{tr} = \mathsf{H}_t^m \boldsymbol{x}_t^m + \boldsymbol{w}_{tr}, \tag{3.29}$$

where H_t^m is an appropriately sized matrix, and $\boldsymbol{w}_{tr}$ is zero mean Gaussian noise with covariance matrix R_t^m.

Under the linear Gaussian measurement model, the measurement probability density function is then given by:

$$\zeta_t^m(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m) = |2\pi\mathsf{R}_t^m|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{z}_{tr} - \mathsf{H}_t^m\boldsymbol{x}_t^m)^{\mathsf{T}}\mathsf{R}_t^{m-1}(\boldsymbol{z}_{tr} - \mathsf{H}_t^m\boldsymbol{x}_t^m)\right\}. \tag{3.30}$$

A commonly analysed problem is where the measurements contain only observations of the target position. For the almost constant velocity target model, this corresponds to:

$$\mathsf{H}_t^m(1\mathrm{D}) = [1 \quad 0]. \tag{3.31}$$

In a similar manner to the matrices F_t^m and G_t^m, the measurement matrix for higher dimensional problems can be expressed in terms of the one dimensional measurement matrix. For example,

$$\mathsf{H}_t^m(3\mathrm{D}) = \begin{bmatrix} \mathsf{H}_t^m(1\mathrm{D}) & 0 & 0 \\ 0 & \mathsf{H}_t^m(1\mathrm{D}) & 0 \\ 0 & 0 & \mathsf{H}_t^m(1\mathrm{D}) \end{bmatrix}. \tag{3.32}$$

3.4 The Kalman Filter

When the evolution and measurement processes are linear and the random elements are Gaussian, then the probability density function of the model state, given a collection of observations of that model, is also Gaussian. So, the posterior state pdf can be fully described by its mean and variance. The Kalman Filter is an optimal recursive estimator for the mean and variance of the posterior state pdf. It provides a method for calculating the mean and variance based on their values at the previous scan and on the measurement observed at the current scan. The Kalman Filter is analysed in detail in many texts (e.g. [CC99]) and will not be reviewed in detail here. However, it is a fundamental building block of most tracking algorithms, in particular the Probabilistic Data Association Filter and Probabilistic Multi-Hypothesis Tracker. For this reason, the Kalman Filter equations are now briefly reviewed.

Suppose, at scan $t-1$, the mean and variance of the posterior state pdf are known. Let these be denoted as:

$$\hat{\boldsymbol{x}}_{t-1}^m = E\left\{\boldsymbol{x}_{t-1}^m | \mathbf{Z}_1, \ldots, \mathbf{Z}_{t-1}\right\}, \tag{3.33}$$

$$\mathsf{P}_{t-1}^m = \mathrm{cov}\left\{\boldsymbol{x}_{t-1}^m | \mathbf{Z}_1, \ldots, \mathbf{Z}_{t-1}\right\}, \tag{3.34}$$

where both (3.33) and (3.34) are implicitly dependent on the known prior distribution of the state, $\psi_0^m(\boldsymbol{x}_0^m)$.

At scan t, the measurement $\boldsymbol{z}_{tm}$ is received. It is assumed to be known that the mth measurement is due to model m. The new mean and variance of the state pdf are calculated using a two step procedure. Firstly, the mean and variance are predicted based on the previous data, this is referred to as the prediction step. In the correction step (also referred to as the filtering or update step) , the predicted quantities are adjusted using the received measurement.

The prediction step extrapolates the state mean and covariance using the following equations:

$$\hat{\boldsymbol{x}}_{t|t-1}^m = \mathsf{F}_t^m \hat{\boldsymbol{x}}_{t-1}^m, \tag{3.35}$$

$$\mathsf{P}_{t|t-1}^m = \mathsf{F}_t^m \mathsf{P}_{t-1}^m \mathsf{F}_t^{m\mathsf{T}} + \mathsf{G}_t^m \mathsf{Q}_t^m \mathsf{G}_t^{m\mathsf{T}}. \tag{3.36}$$

The prediction step also calculates the mean and the covariance of the measurement $\boldsymbol{z}_{tm}$, at scan t, conditioned on the predicted state and its associated error covariance. The mean of the conditional measurement pdf (i.e. the expected measurement) is given by

$$\hat{\boldsymbol{z}}_t^m = \mathsf{H}_t^m \hat{\boldsymbol{x}}_{t|t-1}^m, \tag{3.37}$$

and its covariance is

$$\mathsf{S}_t^m = \mathsf{H}_t^m \mathsf{P}_{t|t-1}^m \mathsf{H}_t^{m\mathsf{T}} + \mathsf{R}_t^m. \tag{3.38}$$

This concludes the prediction step. The correction step adjusts the predicted state statistics from the new measurement, z_{tm}. In the correction step, the filter calculates the measurement *innovation*. The innovation is defined as the difference between the expected measurement and the actual observed measurement, and is denoted ν_t^m:

$$\nu_t^m = \boldsymbol{z}_{tm} - \hat{\boldsymbol{z}}_t^m. \tag{3.39}$$

Note that the innovation has a zero vector mean, because $\hat{\boldsymbol{z}}_t^m$ is defined as the mean of $\boldsymbol{z}_{tm}$, provided the initialisation is unbiased. Also, the covariance of the innovation is the covariance of the conditional measurement pdf, namely S_t^m. Hence, S_t^m is referred to as the innovation covariance matrix. This matrix is very important because it controls the gain of the filter.

Next, the Kalman gain, W_t^m, is calculated using

$$\mathsf{W}_t^m = \mathsf{P}_{t|t-1}^m \mathsf{H}_t^{m\mathsf{T}} \mathsf{S}_t^{m-1}. \tag{3.40}$$

The correction step then concludes by forming the new state mean and covariance:

$$\hat{\boldsymbol{x}}_t^m = \hat{\boldsymbol{x}}_{t|t-1}^m + \mathsf{W}_t^m \nu_t^m, \tag{3.41}$$

$$\mathsf{P}_t^m = \mathsf{P}_{t|t-1}^m - \mathsf{W}_t^m \mathsf{H}_t^m \mathsf{P}_{t|t-1}^m. \tag{3.42}$$

Notice that the Kalman gain and the state covariance do not depend on the actual measurements received. This means that these quantities could be pre-computed if the matrices F_t^m, G_t^m and H_t^m are known a priori. In practice, both F_t^m and G_t^m depend on the difference between consecutive scan times. So, if the scan rate is constant (or varies in a known predictable way) then F_t^m and G_t^m can be known a priori and the gains and covariances computed off line. For many sensors this will not be the case, since the scan rate may fluctuate.

The Kalman Filter can be presented in different formats. In particular, it is possible to write the whole filter as two equations, one for the mean and one for the covariance. However, this format has been chosen because data association at scan t will be performed using the predicted measurement, (3.37).

3.4.1 The Kalman Smoother

The Kalman Filter is used to estimate the state in a time recursive manner, however in some applications the estimator has access to a batch of data. In this case, future measurements can be used to improve the state estimate at scan t. The Kalman filter can be extended to handle batch data, and this extension is called the Kalman Smoother.

The Kalman Smoother is almost identical to the Kalman Filter, but it involves an extra correction step after the filter has calculated the recursive state estimates for the batch. Define the smoothed state mean and covariance as:

$$\hat{\boldsymbol{x}}_{t|T}^m = E\left\{x_t^m | \mathbf{Z}\right\}, \tag{3.43}$$

$$\mathsf{P}_{t|T}^m = \mathrm{cov}\left\{x_t^m | \mathbf{Z}\right\}. \tag{3.44}$$

The state mean and covariance at the final scan T are already conditioned on the whole batch, so

$$\hat{\boldsymbol{x}}_{T|T}^m = \hat{\boldsymbol{x}}_T^m, \tag{3.45}$$

$$\mathsf{P}_{T|T}^m = \mathsf{P}_T^m. \tag{3.46}$$

The mean and covariance at earlier scans are then corrected using the backwards recursions:

$$\hat{\boldsymbol{x}}^m_{t|T} = \hat{\boldsymbol{x}}^m_{t+1|T} + \mathsf{P}^m_t \mathsf{F}^m_{t+1}{}^{\mathsf{T}} \mathsf{P}^m_{t+1|t}{}^{-1} \left[\hat{\boldsymbol{x}}^m_{t+1|T} - \mathsf{F}^m_{t+1} \hat{\boldsymbol{x}}^m_t\right], \tag{3.47}$$

$$\mathsf{P}^m_{t|T} = \mathsf{P}^m_t + \mathsf{P}^m_t \mathsf{F}^m_{t+1}{}^{\mathsf{T}} \mathsf{P}^m_{t+1|t}{}^{-1} \left[\mathsf{P}^m_{t+1|T} - \mathsf{P}^m_{t+1|t}\right] \mathsf{P}^m_{t+1|t}{}^{-1} \mathsf{F}^m_{t+1} \mathsf{P}^m_t. \tag{3.48}$$

Notice that the measurements play no part in the smoother backwards correction. This is because the future state means already contain corrections for the measurements through (3.41). Like the Kalman Filter, the smoother has linear complexity with the number of scans.

3.5 The Probabilistic Multi-Hypothesis Tracker

The Probabilistic Multi-Hypothesis Tracker of [SL95] is a true multi-target tracking algorithm derived from the application of the Expectation-Maximisation (EM) technique of [DLR77]. A fundamental difference between the PMHT and other standard tracking approaches is that the PMHT assumes that the assignment indices for each measurement are independent random variables (see section 3.2.1). This introduces a further parameter of the problem, $\boldsymbol{\Pi}$, the prior probability mass function of the assignments.

The PMHT is an iterative algorithm: it asymptotically approaches a local maximum of the EM auxiliary function by refining estimates for the states, $\mathbf{X}$, and the parameters, $\boldsymbol{\Pi}$. At the ith iteration, denote the estimated states and parameters as $\mathbf{X}^{(i)}$ and $\boldsymbol{\Pi}^{(i)}$ respectively. The estimated states of model m at scan t are denoted $\boldsymbol{x}^{m(i)}_t$ and the estimated prior assignment probability for model m at scan t is $\pi^{m(i)}_t$. The iterations are repeated until the auxiliary function converges (or some other halting criterion is met) at iteration i^* and the estimated track state and parameters at the last iteration comprise the PMHT state and parameter estimates:

$$\hat{\boldsymbol{x}}^m_t = \boldsymbol{x}^{m(i^*)}_t,$$
$$\hat{\pi}^m_t = \pi^{m(i^*)}_t.$$

To start the iterative algorithm, the PMHT requires initial estimates $\mathbf{X}^{(0)}$, and $\boldsymbol{\Pi}^{(0)}$. These may be obtained from earlier data if it is available, or may be merely guesses. $\boldsymbol{\Pi}^{(0)}$ can be initialised as a uniform distribution, $\pi^{m(0)}_t = M^{-1}$.

The estimates $\mathbf{X}^{(i+1)}$ and $\boldsymbol{\Pi}^{(i+1)}$ are found by maximising the EM auxiliary function

$$Q\left(\mathbf{X}, \boldsymbol{\Pi} | \mathbf{X}^{(i)}, \boldsymbol{\Pi}^{(i)}\right) = \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k_{tr}=1}^{M} \log P\left(\mathbf{X}, \mathbf{K}, \mathbf{Z}; \boldsymbol{\Pi}\right) P\left(\mathbf{K} | \mathbf{X}^{(i)}, \mathbf{Z}; \boldsymbol{\Pi}^{(i)}\right), \tag{3.49}$$

with respect to $\mathbf{X}$ and $\boldsymbol{\Pi}$.

Recall that the states of all models are assumed to be first order Markov random variables and are therefore independent of all other prior information when conditioned on the prior state of that model. The assignments are independent identically distributed realisations of the probability mass $\boldsymbol{\Pi}$. The measurements are independent identically distributed realisations of the measurement processes ζ^m_t as defined by the respective assignment. These independence assumptions, combined with Bayes' Rule are used to simplify (3.49) into two terms:

$$Q\left(\mathbf{X}, \boldsymbol{\Pi} | \mathbf{X}^{(i)}, \boldsymbol{\Pi}^{(i)}\right) = \sum_{m=1}^{M} Q^m_X\left(\mathbf{X}^m | \mathbf{X}^{m(i)}, \boldsymbol{\Pi}^{(i)}\right) + \sum_{t=1}^{T} Q_{t\pi}\left(\boldsymbol{\Pi}_t | \mathbf{X}^{(i)}_t, \boldsymbol{\Pi}^{(i)}_t\right). \tag{3.50}$$

$Q_{t\pi}\left(\boldsymbol{\Pi}_t|\mathbf{X}_t^{(i)},\boldsymbol{\Pi}_t^{(i)}\right)$ must be maximised subject to the constraint that $\sum_{m=1}^{M}\pi_t^m=1$, namely $\boldsymbol{\Pi}_t$ should be a proper probability vector. This is achieved by introducing a Lagrangian multiplier and the resulting solution is

$$\pi_t^{m(i)}=\frac{1}{n_t}\sum_{r=1}^{n_t}w_{mtr}^{(i)}, \tag{3.51}$$

where $w_{mtr}^{(i)}$ is referred to as the assignment *weight* for measurement $\boldsymbol{z}_{tr}$ and model m. This weight can be interpreted as the posterior probability that measurement $\boldsymbol{z}_{tr}$ is due to model m. It is given by:

$$w_{mtr}^{(i)}=\frac{\pi_t^{m(i)}\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i)}\right)}{\sum_{s=1}^{M}\pi_t^{s(i)}\zeta_t^s\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{s(i)}\right)}. \tag{3.52}$$

From (3.52), it can be seen that the weight, $w_{mtr}^{(i)}$, captures the dependence upon the state and parameter estimates from the previous iteration.

The term $Q^m\left(\mathbf{X}^m|\mathbf{X}^{(i)},\boldsymbol{\Pi}^{(i)}\right)$ in (3.50) is a function of the states of target m over the batch and is independent of the other models. This implies that the state estimation for model m is independent of the state estimation of the other models. This term is given by:

$$\begin{aligned}&Q^m\left(\mathbf{X}^m|\mathbf{X}^{m(i)},\boldsymbol{\Pi}^{(i)}\right)=\\&\quad\log\psi_0^m\left(\boldsymbol{x}_0^m\right)+\sum_{t=1}^{T}\log\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)+\sum_{t=1}^{T}\sum_{r=1}^{n_t}w_{mtr}^{(i)}\log\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right),\end{aligned} \tag{3.53}$$

The weight equation (3.52) has a linear complexity in the number of models and the models are maximised independently, so the overall PMHT complexity is linear with the number of models.

The state estimate for model m at iteration i, $\mathbf{X}^{m(i+1)}$, is that state sequence that maximises $Q^m\left(\mathbf{X}^m|\mathbf{X}^{(i)},\boldsymbol{\Pi}^{(i)}\right)$. This is a maximum likelihood problem with weighted measurements. The objective likelihood function (3.53) is the same as the likelihood function of an unambiguous measurement problem, except for the measurement term. The unambiguous measurement case is maximised by the Kalman Smoother when there are linear Gaussian statistics. For linear Gaussian statistics, (3.53) becomes

$$\begin{aligned}Q^m\left(\mathbf{X}^m|\mathbf{X}^{m(i)},\boldsymbol{\Pi}^{(i)}\right)=&\log\psi_0^m\left(\boldsymbol{x}_0^m\right)+\sum_{t=1}^{T}\log\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)+C^m+\\&\sum_{t=1}^{T}\sum_{r=1}^{n_t}\frac{-w_{mtr}^{(i)}}{2}(\boldsymbol{z}_{tr}-\mathsf{H}_t^m\boldsymbol{x}_t^m)^{\mathsf{T}}\mathsf{R}_t^{m-1}\left(\boldsymbol{z}_{tr}-\mathsf{H}_t^m\boldsymbol{x}_t^m\right),\end{aligned} \tag{3.54}$$

where C^m is a constant factor due to the measurement density normalisation.

Through algebraic manipulation, (3.54) can be rewritten as

$$\begin{aligned}Q^m\left(\mathbf{X}^m|\mathbf{X}^{m(i)},\boldsymbol{\Pi}^{(i)}\right)=&\log\psi_0^m\left(\boldsymbol{x}_0^m\right)+\sum_{t=1}^{T}\log\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)+C^m+\\&\sum_{t=1}^{T}-\frac{1}{2}\left(\tilde{\boldsymbol{z}}_t^{m(i)}-\mathsf{H}_t^m\boldsymbol{x}_t^m\right)^{\mathsf{T}}\left(\tilde{\mathsf{R}}_t^{m(i)}\right)^{-1}\left(\tilde{\boldsymbol{z}}_t^{m(i)}-\mathsf{H}_t^m\boldsymbol{x}_t^m\right),\end{aligned} \tag{3.55}$$

where

$$\tilde{\boldsymbol{z}}_t^{m(i)} = \frac{1}{n_t \pi_t^{m(i)}} \sum_{r=1}^{n_t} w_{mtr}^{(i)} \boldsymbol{z}_{tr}, \tag{3.56}$$

and

$$\tilde{\mathsf{R}}_t^{m(i)} = \frac{1}{n_t \pi_t^{m(i)}} \mathsf{R}_t^m. \tag{3.57}$$

The likelihood function in (3.55) is the same as the likelihood function optimised by the Kalman smoother. However, instead of the measurement probability density for model m, $\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right)$, a modified measurement density with covariance $\tilde{\mathsf{R}}_t^{m(i)}$ appears. The measurement centroid, $\tilde{\boldsymbol{z}}_t^{m(i)}$, acts as a synthetic measurement. So, the model state estimate, $\boldsymbol{x}_t^{m(i)}$, can be found using a Kalman Smoother where the measurement probability density is given by

$$\tilde{\zeta}_t^m\left(\tilde{\boldsymbol{z}}_t^{m(i)}|\boldsymbol{x}_t^m\right) = |2\pi\tilde{\mathsf{R}}_t^{m(i)}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\tilde{\boldsymbol{z}}_t^{m(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\tilde{\mathsf{R}}_t^{m(i)}\right)^{-1} \left(\tilde{\boldsymbol{z}}_t^{m(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right)\right\}. \tag{3.58}$$

The term $Q_{t\pi}\left(\boldsymbol{\Pi}_t|\mathbf{X}_t^{(i)}, \boldsymbol{\Pi}_t^{(i)}\right)$ in (3.50) is a function of the parameter vector $\boldsymbol{\Pi}_t$ and is independent of the parameters at other scans and independent of the model states. It is given by

$$Q_{t\pi}\left(\boldsymbol{\Pi}_t|\mathbf{X}_t^{(i)}, \boldsymbol{\Pi}_t^{(i)}\right) = \sum_{m=1}^{M} w_{mtr}^{(i)} \log \pi_t^m. \tag{3.59}$$

The PMHT algorithm is then the repeated application of equations (3.52) and (3.51) along with Kalman smoothing to estimate the model states. At each iteration, the states and parameters are modified and this produces new weights through (3.52). When the algorithm converges, the changes in the estimates and weights will become small.

It is important to recognise that the weight equation (3.52) uses the measurement probability density $\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right)$, which is the probability density of the measurements when the model state is known. This is different from the PDAF which uses the probability density of measurements conditioned on the current state estimate (and its covariance).

3.5.1 Homothetic Measurement Models

An important extension of the PMHT is the introduction of homothetic measurement models [RWS95a]. The homothetic measurement model replaces the simple Gaussian model for $\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right)$. Rather than a single Gaussian, the homothetic PMHT uses a Gaussian mixture where each of the components has the same mean, but different covariance matrices. In published uses of the homothetic model, the covariance of each of the components is a scalar multiple of the measurement variance. It is possible to use a more general form of the homothetic model, and this is presented in chapter 6.

The homothetic mixture model comprises of P Gaussian components, each with mean $\mathsf{H}_t^m \boldsymbol{x}_t^m$, and with covariance $\kappa^{mp}\mathsf{R}_t^m$, where κ^{mp} is a scalar. Chapter 6 uses a more general model where the covariances of the different components are arbitrary time varying matrices. Since the measurement process for each target is now a mixture, the index variable k_{tr} must assign measurements to both the appropriate model and the homothetic mixture component of that model's measurement process. So the new assignment index is

$$\boldsymbol{k}_{tr} = \left\{k_{tr}^1, k_{tr}^2\right\}, \tag{3.60}$$

where $k_{tr}^1 \in [1, M]$ indicates the state model that corresponds to the true source of measurement z_{tr}, and $k_{tr}^2 \in [1, P]$ indicates the homothetic mixture component of that model that caused the measurement.

The homothetic mixture model for the measurement process for model m then gives the measurement density:

$$\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) = \begin{cases} \zeta_t^{m1}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) & \text{if } k_{tr}^2 = 1 \\ \zeta_t^{m2}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) & \text{if } k_{tr}^2 = 2 \\ \vdots & \vdots \\ \zeta_t^{mP}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^M\right) & \text{if } k_{tr}^2 = P, \end{cases} \tag{3.61}$$

where

$$\zeta_t^{mp}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) = |2\pi\kappa^{mp}\mathsf{R}_t^m|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{z}_{tr} - \mathsf{H}_t^m\boldsymbol{x}_t^m)^{\mathsf{T}}\left(\kappa^{mp}\mathsf{R}_t^m\right)^{-1}\left(\boldsymbol{z}_{tr} - \mathsf{H}_t^m\boldsymbol{x}_t^m\right)\right\}. \tag{3.62}$$

There is now a different assignment weight for each homothetic component, since they represent different measurement pdfs. These weights are given by

$$w_{mptr}^{(i)} = \frac{\pi_t^{m(i)} P_{\kappa t}^m(p)\, \zeta_t^{mp}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i)}\right)}{\sum_{s=1}^{M}\sum_{q=1}^{P} \pi_t^{s(i)} P_{\kappa t}^{k^1}(q)\, \zeta_t^{sq}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{s(i)}\right)}, \tag{3.63}$$

where $P_{\kappa t}^m(p)$ is the prior probability mass of the index k_{tr}^2 for model m, i.e. the mixing proportions for the components of the measurement density for that model. In [RWS95a, RWS95b], this mass was implicitly assumed to be known to be uniform. However, it may be desirable to assume a nonuniform probability mass, or adapt it with the data. If it is assumed unknown, then the probability mass $P_{\kappa t}^m(p)$ can be estimated in the same way as the probability mass $\mathbf{\Pi}$.

Under this new measurement model, the assignment prior is then estimated by

$$\pi_t^{m(i)} = \frac{1}{n_t}\sum_{r=1}^{n_t}\sum_{p=1}^{P} w_{mptr}^{(i)}. \tag{3.64}$$

If the homothetic model prior is to be estimated, then it is given by

$$P_{\kappa t}^{m(i)}(p) = \frac{\sum_{r=1}^{n_t} w_{mptr}^{(i)}}{\sum_{r=1}^{n_t}\sum_{s=1}^{P} w_{mstr}^{(i)}}. \tag{3.65}$$

The target state estimates can still be derived by a Kalman Smoother with synthetic measurements and covariances defined by

$$\tilde{\boldsymbol{z}}_t^{m(i)} = \left\{\sum_{r=1}^{n_t}\sum_{p=1}^{P}\frac{w_{mptr}^{(i)}}{\kappa^{mp}}\right\}^{-1}\sum_{r=1}^{n_t}\left(\sum_{p=1}^{P}\frac{w_{mptr}^{(i)}}{\kappa^{mp}}\right)\boldsymbol{z}_{tr}, \tag{3.66}$$

and

$$\tilde{\mathsf{R}}_t^{m(i)} = \left\{ \sum_{r=1}^{n_t} \sum_{p=1}^{P} \frac{w_{mptr}^{(i)}}{\kappa^{mp}} \right\}^{-1} \mathsf{R}_t^m. \tag{3.67}$$

The above equations assume that the homothetic models have covariances which are scalar multiples of each other. For the more general case of arbitrary covariance matrices, synthetic measurement and covariance equations are derived in appendix A.

In [RWS95a] the homothetic model was used to allow the PMHT to assign distant measurements to target models and improved track maintenance. Because of the overlapping covariances, the homothetic model also provides a means for the PMHT to prefer closer measurements. Under a simple constant clutter distribution, all measurements close to target models will have a much higher pdf under the target assignment rather than clutter. In this case, all such measurements are treated equally by the PMHT rather than preferring the closest one. In [RWS95b] a homothetic model was used, but the higher covariance components were lumped into the clutter distribution, not the target. This causes the target to pay less attention to more distant measurements.

[WRS98b] presented a comparison of a variety of different PMHT permutations. The homothetic measurement model was one of the best performing PMHT variants.

3.5.2 Manoeuvring Target Models

The inclusion of multiple dynamics models for each target is relatively straight-forward within the PMHT framework. Whereas the homothetic measurement model described in section 3.5.1 provides a mixture model for the measurement probability density function, the standard switching dynamic models used for manoeuver tracking can be represented as a mixture model for the state evolution process.

Manoeuver models for the PMHT were first proposed in [LKH97] and [WRS98a], and have continued to be an area of interest for research, for example [RW01b]. The approach described here is that of [WRS98a].

Let the target state evolution probability density be given by

$$\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) = \begin{cases} \psi_t^{m1}\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) & \text{if } \mu_t^m = 1 \\ \psi_t^{m2}\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) & \text{if } \mu_t^m = 2 \\ \vdots & \vdots \\ \psi_t^{mN}\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) & \text{if } \mu_t^m = N, \end{cases} \tag{3.68}$$

where $\mu_t^m \in [1 \dots N]$ is an index that denotes the component of the mixture distribution present at scan t, namely the index of the manoeuver model that is active. Each of the $\psi_t^{m\mu}\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)$ is a known function.

The manoeuvre model index, μ_t^m, is modelled as a Markov chain with a known transition matrix and prior probability vector. The index is treated as additional missing data, just as the homothetic model index is done. This means that the PMHT requires the conditional probability of the indices given the observed data and the current state estimate. This probability is more difficult to calculate than the probability of the homothetic model index because of the temporal dependence (through the Markov chain). Rather than the simple ratio in (3.65), the conditional probability of the manoeuvre index must be determined using the Hidden Markov Model Smoother.

A common approach to using multiple manoeuvre models is to use almost constant velocity models with different process noise covariance matrices, Q. When these matrices are chosen to be scalar multiples of a base matrix Q_0, then the resulting PMHT algorithm can be implemented using a scalar multiple of Q_0 in much the same way that the homothetic model in the previous section uses a scalar multiple of the measurement matrix. This assumption is not necessary, and more general manoeuvre models give rise to a solution analogous to the general homothetic measurement model presented in chapter 6.

3.5.3 PMHT Permutations

One of the main advantages of the PMHT algorithm is that its structure is amenable to extension. The homothetic measurement model and the manoeuvring target model described above are two examples of this. Many other variants were presented in [WRS98b]. Besides allowing for new additions, this structure also allows the combination of extensions such as the homothetic model and manoeuvring target models. In essence, the derivation of such combinations is simply a matter of careful notational accounting. The result is a superposition of algorithms, much as intuition would predict. An example of this combination of extensions is demonstrated in appendix B, where the various threads of this thesis are drawn together in a single über-algorithm.

3.6 Problem Areas in PMHT

Since the PMHT is a relatively young tracking algorithm, there are problem areas which have been addressed for other filters that remain unresolved for the PMHT. In addition to these problems, there are some areas that are more peculiar to the PMHT because of its data model and iterative nature. Some of these problem areas are described below.

3.6.1 Additional Sensor Information

The standard PMHT formulation includes measurements that are observations of the target state. In practice, there may be additional data available to the tracker. Sensors operate at a finite resolution, and the energy received from targets typically occupies several voxels[1] in the radar image. The standard measurements comprise of estimates of the mode (or mean) of this distributed energy pattern along a particular dimension. However, other measures of the pattern have been used with the augmented PDAF. Measurements such as the area of the pattern, the height of the mode, and the curvature of the spread have all been used in the Augmented PDAF context to improve the discrimination of target and clutter originated measurements [NSC84, Col87, LBS90, CC01]. These approaches are highly effective in reducing the false track rate of the PDAF.

As well as additional features of the sensor signal, some platforms provide the tracker with information from alternative sensors. An example of this is where an ESM receiver is used to provide classification information. This classification information can be used in the tracker to improve performance [CP01]. However, it is more usual for the tracking and classification problems to be dealt with independently.

Non-kinematic measurements are referred to as features using the standard classification terminology. Features with a known, or estimable probability density function are easily incorporated using an approach similar to the Augmented PDAF. The Spectral

[1] voxel: a contraction of **vo**lume pi**xel**. The quantum of a three or higher dimensional image.

Histogram PMHT [Str01] uses a nonparametric approach to estimate the feature distribution for a special kind of feature (namely spectra). This is the only PMHT approach to incorporate feature or classification information.

3.6.2 Dynamic Assignment Prior

A key difference between the PMHT and other tracking approaches is the assumed assignment model. As described in section 3.2.1, the PMHT assumes that the true source of each measurement is an independent realisation of a random process with probability mass $\mathbf{\Pi}_t$. This pmf can also be interpreted as the mixing proportion for each model where the sensor data is treated as a mixture process. If the assignment prior is unknown, then the PMHT can estimate it, but only under restrictive conditions. The standard PMHT algorithm is derived under the assumption that the $\mathbf{\Pi}_t$ are time independent. For many applications, the true mixing proportions of the component models varies smoothly over time, and are not independent from scan to scan. To properly model such cases, it is desirable to generalise the PMHT so that the assignment prior, $\mathbf{\Pi}_t$, is a random process with an arbitrary evolution probability density.

3.6.3 Track Initiation

One of the difficult problems that faces practical tracking systems is that the number of targets present within the surveillance scene is unknown and dynamic. Targets may enter or leave the scene at the boundary of the sensor footprint, but they may also appear and disappear due to the loss of signal propagation or target manoeuvre, especially around airports. It is highly desirable that practical tracking algorithms used for real time systems are able to automatically form new tracks when new targets appear, and terminate old tracks when targets vanish.

Like most tracking algorithms, the standard PMHT is predicated on a known and fixed number of target models. Furthermore, the algorithm requires a priori knowledge about the states of these targets. Since the PMHT is a hill climbing algorithm, the state estimate may converge to a local maximum if proper initialisation is not used. This assumption is contrary to the goal of automated track initiation and termination.

As reviewed in chapter 2, a number of different approaches have been used with other tracking filters to perform automated track decisions. In the following chapters, some of these techniques are incorporated into the PMHT framework and their performance is analysed for the problem of initiating target tracks in clutter with poor initialisation.

3.6.4 Application of PMHT to Operational Sensor Systems

A consequence of the youth of the PMHT algorithm is that there a very few implementations of the PMHT for real sensors. Work on the PMHT has mainly been studies using simulated data. While these are useful and provide a test-bed for theoretical advancement, stalwart practitioners are only convinced by performance on real systems. The goal of the following chapters is to develop a PMHT algorithm capable of handling the difficult problems encountered with practical data. Ultimately this algorithm is run on recorded radar data with better than real time speed. The performance of the prototype PMHT developed here is compared with the Unified PDAF (UPDAF), the current operational tracker for the Australian JFAS radar.

Chapter 4

The Probabilistic Multi-Hypothesis Tracker with Classification Measurements

The original formulation of the Probabilistic Multi-Hypothesis Tracker deals with measurements that are instantaneous observations of the state of a particular model. The data association problem arises because the particular model that caused any particular measurement is unknown. Thus the PMHT forms an estimate of the unknown model states based on a collection of state observations with uncertain origin. The algorithm estimates the model states by maximising the conditional expectation of the log likelihood with respect to the model to measurement assignments.

In practical applications, a sensor may be able to gather other information in addition to state observations. While this information may not be useful in estimating the state of the targets of interest, it may improve the tracking algorithm's ability to associate measurements with each target track and hence improve the overall performance. This chapter considers the case where the tracking filter has an estimate of the class of the target that caused each available state observation. This classification measurement is treated as an observation of the assignment of the corresponding measurement.

One physical example of a system where classification measurements exist is high resolution radar. At microwave frequencies, the wavelength of the transmitted waveform may be relatively short compared with the physical size of the target. This causes the target return to be distributed through several range bins. The distributed range response of a target is referred to as a *range profile*. The range profiles from various azimuth angles can be collated to form a radar image of the target. The location of primary scatterers and other features of this image can be used to classify the target [JO97]. Another example is where a radar platform has access to data from an electronic support measure (ESM) receiver [CP01]. An ESM receiver is a passive sensor that detects radiation from radar emitters on other platforms. The ESM receiver can provide classification information about emitters at various bearings that can be associated with radar detections aligned at the same bearings.

The PMHT algorithm derived in this chapter is designed to take advantage of classification measurements to improve data association and state estimation. This extension of the PMHT is referred to as the PMHT-c. It will be shown that the standard PMHT is a special case of the PMHT-c algorithm that is realised when there are no classification measurements available, or equivalently, when the classifications provide no information.

The derivation of the PMHT-c follows the same development as the original PMHT

algorithm. The derivation is presented in full here for completeness.

4.1 Derivation

Using the observer model developed in chapter 3, write the likelihood of the observer and the received measurements as

$$L\left(\mathbf{O},\mathbf{Z}\right)=P\left(\mathbf{O},\mathbf{Z}\right)=P\left(\mathbf{X},\mathbf{K},\mathbf{Z}\right). \tag{4.1}$$

The observer likelihood in (4.1) is implicitly dependent on the parameters, $\mathbf{\Pi}$, which provide the prior distribution of the assignments, $\mathbf{K}$. In general, neither $\mathbf{X}$, $\mathbf{K}$, nor $\mathbf{\Pi}$ are known and all must be estimated in order to achieve a state estimate $\hat{\mathbf{X}}$, which is the tracking goal.

It is not feasible to compute the maximum likelihood solution for $\mathbf{X}$, $\mathbf{K}$ and $\mathbf{\Pi}$ because the number of permutations of the assignments $\mathbf{K}$ grows exponentially with the batch length, T, and the number of models M. Instead, the Expectation-Maximisation (EM) algorithm is used to derive an iterative scheme that converges to the maximum likelihood estimates for the states, $\mathbf{X}$, and the assignment prior, $\mathbf{\Pi}$ without directly estimating the assignments themselves. Using EM terminology, the assignments $\mathbf{K}$ are the *missing data* and $\mathbf{O}$ and $\mathbf{Z}$ are collectively the complete data. The complete data likelihood is L.

Define an auxiliary function

$$Q\left(\mathbf{X},\mathbf{\Pi}|\hat{\mathbf{X}}^{(i)},\hat{\mathbf{\Pi}}^{(i)}\right)\equiv\sum_{\mathbf{K}}\log L\left(\mathbf{O},\mathbf{Z}\right)P\left(\mathbf{K}|\hat{\mathbf{X}}^{(i)},\mathbf{Z}\right), \tag{4.2}$$

where $\hat{\mathbf{X}}^{(i)}$ is the state estimate at iteration i and $\hat{\mathbf{\Pi}}^{(i)}$ is the estimate of the prior probability of the assignments at iteration i.

The summation denoted $\sum_{\mathbf{K}}$ is the sum over all possible permutations of the assignments, $\mathbf{K}$. Explicitly,

$$\sum_{\mathbf{K}}\{\cdot\}\equiv\sum_{k_{11}=1}^{M}\sum_{k_{12}=1}^{M}\cdots\sum_{k_{1n_t}=1}^{M}\sum_{k_{21}=1}^{M}\cdots\sum_{k_{Tn_T}=1}^{M}\{\cdot\}. \tag{4.3}$$

A recursive estimator for $\mathbf{X}$ and $\mathbf{\Pi}$ is achieved by finding the values of $\mathbf{X}$ and $\mathbf{\Pi}$ that maximise $Q\left(\mathbf{X},\mathbf{\Pi}|\hat{\mathbf{X}}^{(i)},\hat{\mathbf{\Pi}}^{(i)}\right)$. That is,

$$\begin{aligned}\hat{\mathbf{X}}^{(i+1)},\hat{\mathbf{\Pi}}^{(i+1)} &= \arg\max Q\left(\mathbf{X},\mathbf{\Pi}|\hat{\mathbf{X}}^{(i)},\hat{\mathbf{\Pi}}^{(i)}\right)\\ &= \arg\max\sum_{\mathbf{K}}\log L\left(\mathbf{O},\mathbf{Z}\right)P\left(\mathbf{K}|\hat{\mathbf{X}}^{(i)},\mathbf{Z}\right),\end{aligned} \tag{4.4}$$

4.1.1 Classification measurements

Suppose that each measurement $\boldsymbol{z}_{tr}$ at scan t has an associated classification measurement. Let $\boldsymbol{z}_{tr}^{(x)}$ denote the measurement vector produced by an observation of the state of an unknown model k_{tr}. Denote the classification measurement associated with $\boldsymbol{z}_{tr}^{(x)}$ as $\boldsymbol{z}_{tr}^{(k)}$. Let the total measurement vector be the collection of the state observation with its associated classification measurement,

$$\boldsymbol{z}_{tr}\equiv\left[\begin{array}{c}\boldsymbol{z}_{tr}^{(x)}\\ \boldsymbol{z}_{tr}^{(k)}\end{array}\right]. \tag{4.5}$$

The probability density function of the state observation $\boldsymbol{z}_{tr}^{(x)}$ is given by the measurement process for the model designated by k_{tr} as defined in chapter 3. Thus,

$$P\left(\boldsymbol{z}_{tr}^{(x)}|k_{tr},\mathbf{X}_t\right)=\zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^{k_{tr}}\right). \tag{4.6}$$

The classification measurement is a discrete variable that takes its value from an enumerated set of classes. The number of classes that define the possible outcomes for $\boldsymbol{z}_{tr}^{(k)}$ is, in general, not the same as the number of models, M. For example, if the classification measurement is due to an ESM receiver, the set of classes may be the set of radar platforms known to the ESM classifier, which will be a large set of vehicles. Alternatively, if the classification is due to poor resolution range profiles the classes may be broad groups such as { small, medium, and large}. Let the number of classes in the set of possible class measurements be M_C. The range of possible values of $\boldsymbol{z}_{tr}^{(k)}$ will be $[1\ldots M_C]$.

Assume that the classification measurement is independent of the model state, $\mathbf{X}$, and also independent of the state observation $\boldsymbol{z}_{tr}^{(x)}$. Thus,

$$\begin{aligned}P\left(\boldsymbol{z}_{tr}|k_{tr},\mathbf{X}_t\right)&=P\left(\boldsymbol{z}_{tr}^{(x)},\boldsymbol{z}_{tr}^{(k)}|k_{tr},\mathbf{X}_t\right)\\&=P\left(\boldsymbol{z}_{tr}^{(x)}|k_{tr},\mathbf{X}_t\right)P\left(\boldsymbol{z}_{tr}^{(k)}|\boldsymbol{z}_{tr}^{(x)},k_{tr},\mathbf{X}_t\right)\\&=P\left(\boldsymbol{z}_{tr}^{(x)}|k_{tr},\mathbf{X}_t\right)P\left(\boldsymbol{z}_{tr}^{(k)}|k_{tr}\right).\end{aligned} \tag{4.7}$$

This is equivalent to assuming that the available measurement data has been partitioned into dynamic and non-dynamic parts. These then comprise $\boldsymbol{z}_{tr}^{(x)}$ and $\boldsymbol{z}_{tr}^{(k)}$ respectively. If this is not the case, then the classification measurements provide information about the state and must be taken into account during the state optimisation.

The probability mass function $P\left(\boldsymbol{z}_{tr}^{(k)}|k_{tr}\right)$ can be represented by a matrix with M_C rows corresponding to possible class values for $\boldsymbol{z}_{tr}^{(k)}$ and M columns corresponding to the possible values of k_{tr}. Such a matrix is referred to as a *confusion matrix* in classification literature [TK99]. Denote the confusion matrix $\mathsf{C}=\{c_{ij}\}$ with $c_{ij}\equiv P\left(\boldsymbol{z}_{tr}^{(k)}=i|k_{tr}=j\right)$. So c_{ij} is the probability that the classification process will produce the class output i when the observation was truly caused by model j.

Using the confusion matrix elements, c_{ij}, and the model measurement processes, ζ_t^m, the total measurement probability is then given by

$$P\left(\boldsymbol{z}_{tr}|k_{tr},\mathbf{X}_t\right)=\zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^{k_{tr}}\right)c_{\boldsymbol{z}_{tr}^{(k)}k_{tr}}. \tag{4.8}$$

In some cases, the confusion matrix values may not be known by the tracking filter. If they are not known, then they must be treated as unknown parameters of the system. In this case, the auxiliary function becomes

$$Q\left(\mathbf{X},\mathbf{\Pi},\mathsf{C}|\hat{\mathbf{X}}^{(i)},\hat{\mathbf{\Pi}}^{(i)},\hat{\mathsf{C}}^{(i)}\right)\equiv\sum_{\mathbf{K}}\log L\left(\mathbf{O},\mathbf{Z}\right)P\left(\mathbf{K}|\hat{\mathbf{X}}^{(i)},\mathbf{Z}\right). \tag{4.9}$$

The iterative estimates for the confusion matrix elements are found by maximising the auxiliary function.

4.1.2 Complete data likelihood

The expression $L(\mathbf{O}, \mathbf{Z})$ in (4.1) is referred to as the complete data likelihood and is the joint probability of the states, the assignments and the measurements. This probability may be rewritten via Bayes' Rule as

$$L(\mathbf{O}, \mathbf{Z}) = P(\mathbf{X})P(\mathbf{K}|\mathbf{X})P(\mathbf{Z}|\mathbf{K}, \mathbf{X}). \tag{4.10}$$

The measurements are assumed to be independent conditioned on the states and the assignments, so

$$P(\mathbf{Z}|\mathbf{K}, \mathbf{X}) = \prod_{t=1}^{T} \prod_{r=1}^{n_t} P\left(\boldsymbol{z}_{tr}|k_{tr}, \mathbf{X}_t\right), \tag{4.11}$$

where $P\left(\boldsymbol{z}_{tr}|k_{tr}, \mathbf{X}_t\right)$ is defined by (4.8).

The assignments are assumed to be independent of each other and independent of the model states, so

$$P(\mathbf{K}|\mathbf{X}) = \prod_{t=1}^{T} \prod_{r=1}^{n_t} P\left(k_{tr}\right) = \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k_{tr}}. \tag{4.12}$$

The targets are assumed to be independent of each other and independent of the clutter models, so

$$P(\mathbf{X}) = \prod_{m=1}^{M} P\left(\mathbf{X}^m\right). \tag{4.13}$$

The models are assumed to have a first order Markov dependence, so the state at scan t is independent of the state at scans $t' < t-1$ when conditioned on the state at scan $t-1$. This means

$$\begin{aligned} P(\mathbf{X}) &= \prod_{m=1}^{M} \left\{ p\left(\boldsymbol{x}_0^m\right) \prod_{t=1}^{T} p\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) \right\} \\ &= \prod_{m=1}^{M} \left\{ \psi_0^m\left(\boldsymbol{x}_0^m\right) \prod_{t=1}^{T} \psi_T^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) \right\}. \end{aligned} \tag{4.14}$$

Substituting (4.8), (4.11), (4.12) and (4.14) into the complete data likelihood (4.10) gives

$$L(\mathbf{O}, \mathbf{Z}) = \prod_{m=1}^{M} \left\{ \psi_0^m\left(\boldsymbol{x}_0^m\right) \prod_{t=1}^{T} \psi_T^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k_{tr}} \zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^{k_{tr}}\right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}}. \tag{4.15}$$

4.1.3 Conditional probability of the assignments

Computing the auxiliary function in (4.9) requires the probability of the assignments conditioned on the states and the measurements, $P(\mathbf{K}|\mathbf{X}, \mathbf{Z})$. Using Bayes' Rule, write

$$\begin{aligned} P(\mathbf{K}|\mathbf{X}, \mathbf{Z}) &= \frac{P(\mathbf{X}, \mathbf{K}, \mathbf{Z})}{P(\mathbf{X}, \mathbf{Z})} \\ &= \frac{P(\mathbf{X}, \mathbf{K}, \mathbf{Z})}{\sum_{\mathbf{K}} P(\mathbf{X}, \mathbf{K}, \mathbf{Z})} \\ &= \frac{L(\mathbf{O}, \mathbf{Z})}{\sum_{\mathbf{K}} L(\mathbf{O}, \mathbf{Z})}. \end{aligned} \tag{4.16}$$

Both the clutter states and the target states are independent of the assignments, so the terms in $L(\mathbf{O}, \mathbf{Z})$ corresponding to these can be factored out of the sum in the denominator of (4.16). These terms then cancel with the same terms in the numerator

$$\begin{aligned} P(\mathbf{K}|\mathbf{X}, \mathbf{Z}) &= \frac{\prod_{m=1}^{M} \left\{ \psi_0^m \left(\boldsymbol{x}_0^m \right) \prod_{t=1}^{T} \psi_T^m \left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m \right) \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k_{tr}} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}}}{\sum_{\mathbf{K}'} \prod_{m=1}^{M} \left\{ \psi_0^m \left(\boldsymbol{x}_0^m \right) \prod_{t=1}^{T} \psi_T^m \left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m \right) \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k'_{tr}} \zeta_t^{k'_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k'_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k'_{tr}}} \\ &= \frac{\prod_{m=1}^{M} \left\{ \psi_0^m \left(\boldsymbol{x}_0^m \right) \prod_{t=1}^{T} \psi_T^m \left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m \right) \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k_{tr}} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}}}{\prod_{m=1}^{M} \left\{ \psi_0^m \left(\boldsymbol{x}_0^m \right) \prod_{t=1}^{T} \psi_T^m \left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m \right) \right\} \sum_{\mathbf{K}'} \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k'_{tr}} \zeta_t^{k'_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k'_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k'_{tr}}} \\ &= \frac{\prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k_{tr}} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}}}{\sum_{\mathbf{K}'} \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k'_{tr}} \zeta_t^{k'_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k'_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k'_{tr}}}, \end{aligned} \tag{4.17}$$

where $\mathbf{K}'$ is simply a dummy index.

The sum of products in the denominator of (4.17) is equivalent to a product of sums, namely

$$\sum_{\mathbf{K}'} \prod_{t=1}^{T} \prod_{r=1}^{n_t} f(.) \equiv \prod_{t=1}^{T} \prod_{r=1}^{n_t} \sum_{k'_{tr}=1}^{M} f(.). \tag{4.18}$$

Thus (4.17) can be simplified to

$$\begin{aligned} P(\mathbf{K}|\mathbf{X}, \mathbf{Z}) &= \frac{\prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k_{tr}} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}}}{\prod_{t=1}^{T} \prod_{r=1}^{n_t} \sum_{k'_{tr}=1}^{M} \pi_t^{k'_{tr}} \zeta_t^{k'_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k'_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k'_{tr}}} \\ &= \prod_{t=1}^{T} \prod_{r=1}^{n_t} \frac{\pi_t^{k_{tr}} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}}}{\sum_{k'_{tr}=1}^{M} \pi_t^{k'_{tr}} \zeta_t^{k'_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k'_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k'_{tr}}}. \end{aligned} \tag{4.19}$$

Define

$$w_{k_{tr}tr} = \frac{\pi_t^{k_{tr}} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}}}{\sum_{k'_{tr}=1}^{M} \pi_t^{k'_{tr}} \zeta_t^{k'_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k'_{tr}} \right) c_{\boldsymbol{z}_{tr}^{(k)} k'_{tr}}}, \tag{4.20}$$

then (4.19) becomes

$$P(\mathbf{K}|\mathbf{X}, \mathbf{Z}) = \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{tr}tr}. \tag{4.21}$$

The w_{mtr} is referred to as the association weight for measurement r at scan t and model m. It can be thought of as the posterior probability that the assignment $k_{tr} = m$ whereas the prior probability is, by definition, π_t^m. The weight w_{mtr} is implicitly iteration dependent because it is a function of the estimates at iteration i.

4.1.4 Expectation Step

The expectation step of the EM algorithm involves the calculation of the auxiliary function, Q. Recall that the auxiliary function is defined in (4.9) as

$$Q\left(\mathbf{X}, \mathbf{\Pi}, \mathsf{C} | \mathbf{X}^{(i)}, \mathbf{\Pi}^{(i)}, \mathsf{C}^{(i)}\right) = \sum_{\mathbf{K}} \log L(\mathbf{O}, \mathbf{Z}; \mathbf{\Pi}, \mathsf{C}) P\left(\mathbf{K} | \mathbf{X}^{(i)}, \mathbf{Z}; \mathbf{\Pi}^{(i)}, \mathsf{C}^{(i)}\right),$$

where the dependence on the probability mass functions $\mathbf{\Pi}$ and C is shown explicitly. The quantities $\mathbf{X}^{(i)}$, $\mathbf{\Pi}^{(i)}$, and $\mathsf{C}^{(i)}$ are the estimates found from the i^{th} iteration of the EM algorithm, and the estimates for the $i + 1^{th}$ iteration are found by maximising Q.

From (4.15) write

$$\begin{aligned}
\log L(\mathbf{O}, \mathbf{Z}; \mathbf{\Pi}, \mathsf{C}) &= \log \left\{ \prod_{m=1}^{M} \left\{ \psi_0^m\left(\boldsymbol{x}_0^m\right) \prod_{t=1}^{T} \psi_T^m\left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m\right) \right\} \times \right. \\
&\qquad \left. \prod_{t=1}^{T} \prod_{r=1}^{n_t} \pi_t^{k_{tr}} \zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}}\right) c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}} \right\} \\
&= \sum_{m=1}^{M} \left\{ \log \psi_0^m\left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T} \log \psi_T^m\left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m\right) \right\} \\
&\qquad + \sum_{t=1}^{T} \sum_{r=1}^{n_t} \left\{ \log \pi_t^{k_{tr}} + \log \zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}}\right) + \log c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}} \right\}. \quad (4.22)
\end{aligned}$$

Substituting (4.22) and (4.21) into (4.9) gives

$$\begin{aligned}
&Q\left(\mathbf{X}, \mathbf{\Pi}, \mathsf{C} | \mathbf{X}^{(i)}, \mathbf{\Pi}^{(i)}, \mathsf{C}^{(i)}\right) = \\
&\qquad \sum_{\mathbf{K}} \sum_{m=1}^{M} \left\{ \log \psi_0^m\left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T} \log \psi_T^m\left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m\right) \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{tr}tr}^{(i)} \\
&\qquad + \sum_{\mathbf{K}} \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \log \pi_t^{k_{tr}} \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{tr}tr}^{(i)} \\
&\qquad + \sum_{\mathbf{K}} \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \log \zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}}\right) \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{tr}tr}^{(i)} \\
&\qquad + \sum_{\mathbf{K}} \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \log c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}} \right\} \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{tr}tr}^{(i)}. \quad (4.23)
\end{aligned}$$

The notation $w_{k_{tr}tr}^{(i)}$ is used to indicate that the measurement to model assignment weight is calculated using the estimated states and parameters from iteration i, namely $\mathbf{X}^{(i)}$, $\Pi^{(i)}$, and $\mathsf{C}^{(i)}$.

The following two identities are used to simplify (4.23),

$$\sum_{\mathbf{K}} \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{tr}tr} = 1, \quad (4.24)$$

and

$$\sum_{\mathbf{K}\backslash k_{us}} \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{tr}tr} = w_{k_{us}us}, \tag{4.25}$$

where $\sum_{\mathbf{K}\backslash k_{us}}$ is the sum over all assignment indices except k_{us}.

Applying (4.24) and (4.25) to (4.23) gives

$$\begin{aligned} Q\left(\mathbf{X}, \mathbf{\Pi}, \mathsf{C}|\mathbf{X}^{(i)}, \mathbf{\Pi}^{(i)}, \mathsf{C}^{(i)}\right) &= \sum_{m=1}^{M} \left\{ \log \psi_0^m \left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T} \log \psi_T^m \left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m\right) \right\} \\ &+ \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log \pi_t^k w_{ktr}^{(i)} \right\} \\ &+ \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log \zeta_t^k \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^k\right) w_{ktr}^{(i)} \right\} \\ &+ \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log c_{\boldsymbol{z}_{tr}^{(k)}k} w_{ktr}^{(i)} \right\}. \end{aligned} \tag{4.26}$$

(4.26) can be grouped into the following three terms

$$Q\left(\mathbf{X}, \mathbf{\Pi}, \mathsf{C}|\mathbf{X}^{(i-1)}, \mathbf{\Pi}^{(i)}, \mathsf{C}^{(i)}\right) = \sum_{m=1}^{M} Q_X^m + \sum_{t=1}^{T} Q_{t\pi} + Q_C. \tag{4.27}$$

The Q_X^m term in (4.27) depends only on the target model states and the measurements and is given by

$$Q_X^m = \log \psi_0^m \left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T} \log \psi_t^m \left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m\right) + \sum_{t=1}^{T} \sum_{r=1}^{n_t} \log \zeta_t^m \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^m\right) w_{mtr}^{(i)}. \tag{4.28}$$

The $Q_{t\pi}$ term in (4.27) depends only on the prior probability $\mathbf{\Pi}$ and is given by

$$Q_{t\pi} = \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log \pi_t^k w_{ktr}^{(i)}. \tag{4.29}$$

The Q_C term in (4.27) depends only on the classification measurements $\boldsymbol{z}_{tr}^{(k)}$ and the confusion matrix C and is given by

$$Q_C = \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log c_{\boldsymbol{z}_{tr}^{(k)}k} w_{ktr}^{(i)}. \tag{4.30}$$

4.1.5 Maximisation Step

The EM maximisation step updates the estimates of the unknown states and parameters by maximising the auxiliary function Q. Because the auxiliary function is the sum of

independent terms, the overall function can be maximised by performing independent maximisation on each of the terms.

$$\begin{aligned}\mathbf{X}^{m(i+1)} &= \arg\max Q_X^m,\\ \mathbf{\Pi}^{(i+1)} &= \arg\max \sum_{t=1}^{T} Q_{t\pi},\\ \mathsf{C}^{(i+1)} &= \arg\max Q_C.\end{aligned}$$

4.1.5.1 State estimate

The maximisation of the state terms Q_X^m depends on the particular forms of the evolution probabilities $\psi_t^m(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m)$ and the measurement processes $\zeta_t^m(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m)$ and may be different for each model since the models may have functionally different probabilistic models. For the special case of linear Gaussian functions, Streit and Luginbuhl have shown that the solution for the states is a fixed interval Kalman smoother. The proof of this important result is repeated for completeness.

Let us examine the measurement term in (4.28) which is given by

$$\sum_{t=1}^{T}\sum_{r=1}^{n_t} \log \zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m\right) w_{mtr}^{(i)}.$$

For the case of linear Gaussian statistics, the measurement function for model m is

$$\zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m\right) = \frac{1}{|2\pi \mathsf{R}_t^m|^{1/2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{z}_{tr}^{(x)} - \mathsf{H}_t^m\boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1}\left(\boldsymbol{z}_{tr}^{(x)} - \mathsf{H}_t^m\boldsymbol{x}_t^m\right)\right\}, \quad (4.31)$$

where H_t^m is an appropriately sized matrix and R is the measurement covariance. More detail on linear Gaussian models is given in section 3.3.

Taking the log of (4.31) gives

$$\log \zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m\right) = \log\left\{\frac{1}{|2\pi R_t^m|^{1/2}}\right\} - \frac{1}{2}\left(\boldsymbol{z}_{tr}^{(x)} - \mathsf{H}_t^m\boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1}\left(\boldsymbol{z}_{tr}^{(x)} - \mathsf{H}_t^m\boldsymbol{x}_t^m\right). \quad (4.32)$$

The first term in (4.32) is independent of the target states and thus constant. It can be ignored for the purpose of the maximisation problem. The second term in (4.32) is quadratic in the target state. Using (4.32) write

$$\sum_{r=1}^{n_t} \log \zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m\right) w_{mtr}^{(i)} = \sum_{r=1}^{n_t} \frac{-w_{mtr}^{(i)}}{2}\left(\boldsymbol{z}_{tr}^{(x)} - \mathsf{H}_t^m\boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1}\left(\boldsymbol{z}_{tr}^{(x)} - \mathsf{H}_t^m\boldsymbol{x}_t^m\right) + A, \quad (4.33)$$

where A is independent of the target state $\boldsymbol{x}_t^m$ and can be ignored.

Expanding the quadratic terms and collecting all state independent expressions into the constant A gives

$$\begin{aligned}\sum_{r=1}^{n_t} \log \zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m\right) w_{mtr}^{(i)} &= -\frac{1}{2}\Bigg\{\left(\mathsf{H}_t^m\boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1}\left(\mathsf{H}_t^m\boldsymbol{x}_t^m\right)\sum_{r=1}^{n_t} w_{mtr}^{(i)}\\ &\quad - \left(\sum_{r=1}^{n_t} w_{mtr}^{(i)}\boldsymbol{z}_{tr}^{(x)}\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1}\left(\mathsf{H}_t^m\boldsymbol{x}_t^m\right)\\ &\quad - \left(\mathsf{H}_t^m\boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1}\left(\sum_{r=1}^{n_t} w_{mtr}^{(i)}\boldsymbol{z}_{tr}^{(x)}\right)\Bigg\} + A. \quad (4.34)\end{aligned}$$

Substituting

$$\tilde{\boldsymbol{z}}_{mt}^{(i)} \equiv \frac{\sum_{r=1}^{n_t} w_{mtr}^{(i)} \boldsymbol{z}_{tr}^{(x)}}{\sum_{r=1}^{n_t} w_{mtr}^{(i)}}, \tag{4.35}$$

into (4.34) gives

$$\begin{aligned}\sum_{r=1}^{n_t} \log \zeta_t^m \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^m\right) w_{mtr}^{(i)} &= -\frac{1}{2} \sum_{r=1}^{n_t} w_{mtr}^{(i)} \Big\{ \left(\mathsf{H}_t^m \boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1} \left(\mathsf{H}_t^m \boldsymbol{x}_t^m\right) \\ &\quad - \left(\tilde{\boldsymbol{z}}_{mt}^{(i)}\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1} \left(\mathsf{H}_t^m \boldsymbol{x}_t^m\right) \\ &\quad - \left(\mathsf{H}_t^m \boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1} \left(\tilde{\boldsymbol{z}}_{mt}^{(i)}\right) \Big\} + A. \end{aligned} \tag{4.36}$$

Completing the square, gives

$$\begin{aligned}\sum_{r=1}^{n_t} \log \zeta_t^m \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^m\right) w_{mtr}^{(i)} &= -\frac{1}{2} \sum_{r=1}^{n_t} w_{mtr}^{(i)} \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\mathsf{R}_t^m\right)^{-1} \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right) + A \\ &= -\frac{1}{2} \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\tilde{\mathsf{R}}_t^{m(i)}\right)^{-1} \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right) + A \\ &\propto \log \tilde{\zeta}_t^m \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} | \boldsymbol{x}_t^m\right), \end{aligned} \tag{4.37}$$

where

$$\tilde{\mathsf{R}}_t^{m(i)} = \frac{1}{\sum_{r=1}^{n_t} w_{mtr}^{(i)}} \mathsf{R}_t^m, \tag{4.38}$$

and

$$\tilde{\zeta}_t^m \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} | \boldsymbol{x}_t^m\right) = \frac{1}{\left|2\pi \tilde{\mathsf{R}}_t^{m(i)}\right|^{1/2}} \exp\left\{-\frac{1}{2} \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right)^{\mathsf{T}} \left(\tilde{\mathsf{R}}_t^{m(i)}\right)^{-1} \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} - \mathsf{H}_t^m \boldsymbol{x}_t^m\right)\right\}. \tag{4.39}$$

Ignoring constant terms, write (4.28) as

$$Q_X^m = \log \psi_0^l \left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T} \log \psi_t^m \left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m\right) + \sum_{t=1}^{T} \log \tilde{\zeta}_t^m \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} | \boldsymbol{x}_t^m\right). \tag{4.40}$$

Observe that (4.40) is exactly the log-likelihood of a linear Gaussian state estimation problem with measurements $\tilde{\boldsymbol{z}}_{mt}^{(i)}$ and measurement probability density $\tilde{\zeta}_t^m \left(\tilde{\boldsymbol{z}}_{mt}^{(i)} | \boldsymbol{x}_t^m\right)$. Thus, the Kalman smoother can be employed to obtain $\mathbf{X}^{(i+1)}$ because the Kalman smoother is the Maximum Likelihood estimator for this problem.

For the case of linear Gaussian statistics, the state estimate for iteration $i+1$ is found by running a Kalman smoother with the synthetic measurements

$$\tilde{\boldsymbol{z}}_{mt}^{(i)} \equiv \frac{\sum_{r=1}^{n_t} w_{mtr}^{(i)} \boldsymbol{z}_{tr}^{(x)}}{\sum_{r=1}^{n_t} w_{mtr}^{(i)}},$$

and with a modified measurement covariance given by

$$\tilde{\mathsf{R}}_t^{m(i)} = \frac{1}{\sum_{r=1}^{n_t} w_{mtr}^{(i)}} \mathsf{R}_t^m.$$

4.1.5.2 Assignment prior estimate

The $Q_{t\pi}$ term in (4.27) must be maximised subject to the constraint $\sum_{k=1}^{M} \pi_t^k = 1$. This is achieved by using the Lagrangian

$$L_{t\pi} = \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log \pi_t^k w_{ktr}^{(i)} + \lambda_t^\pi \left(1 - \sum_{k=1}^{M} \pi_t^k \right), \tag{4.41}$$

where λ_t^π is the Lagrange multiplier.

Differentiating the Lagrangian with respect to π_t^k and solving for stationary points gives

$$\pi_t^{k(i)} = \frac{1}{\lambda_t^\pi} \sum_{r=1}^{n_t} w_{ktr}^{(i)}. \tag{4.42}$$

Reapplying the constraint gives

$$\lambda_t^\pi = \sum_{r=1}^{n_t} \sum_{k=1}^{M} w_{ktr}^{(i)} = n_t. \tag{4.43}$$

So the estimate for π_t^k at iteration $i+1$ is given by the unique stationary point of the Lagrangian

$$\pi_t^{k(i+1)} = \frac{1}{n_t} \sum_{r=1}^{n_t} w_{ktr}^{(i)}. \tag{4.44}$$

4.1.5.3 Confusion matrix estimate

The Q_C term in (4.27) must be maximised subject to the constraint $\sum_{i=1}^{M_C} c_{ij} = 1$. This is achieved by using the Lagrangian

$$L_C = \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log c_{\boldsymbol{z}_{tr}^{(k)} k} w_{ktr}^{(i)} + \sum_{j=1}^{M} \lambda_j^c \left(1 - \sum_{i=1}^{M_C} c_{ij} \right), \tag{4.45}$$

where λ_j^c is the Lagrange multiplier.

Rewrite (4.45) as

$$L_C = \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \sum_{i=1}^{M_C} \delta \left(\boldsymbol{z}_{tr}^{(k)} - i \right) \log c_{ik} w_{ktr}^{(i)} + \sum_{j=1}^{M} \lambda_j^c \left(1 - \sum_{i=1}^{M_C} c_{ij} \right), \tag{4.46}$$

where $\delta(t)$ is the Kronecker delta function, an identity function that is unity at the origin and zero elsewhere.

Differentiating the Lagrangian with respect to c_{ij} and solving for stationary points gives

$$c_{ij}^{(i)} = \frac{1}{\lambda_j^c} \sum_{t=1}^{T} \sum_{r=1}^{n_t} \delta\left(\boldsymbol{z}_{tr}^{(k)} - i\right) w_{jtr}^{(i)}. \tag{4.47}$$

Reapplying the constraint gives

$$\begin{aligned} \lambda_j^c &= \sum_{t=1}^{n_t} \sum_{r=1}^{n_t} \sum_{i=1}^{M_C} \delta\left(\boldsymbol{z}_{tr}^{(k)} - i\right) w_{jtr}^{(i)} \\ &= \sum_{t=1}^{n_t} \sum_{r=1}^{n_t} w_{jtr}^{(i)}. \end{aligned} \tag{4.48}$$

So the estimate for c_{ij} at iteration $i+1$ is given by the unique stationary point of the Lagrangian

$$c_{ij}^{(i+1)} = \frac{\sum_{t=1}^{T} \sum_{r=1}^{n_t} \delta\left(\boldsymbol{z}_{tr}^{(k)} - i\right) w_{jtr}^{(i)}}{\sum_{t=1}^{n_t} \sum_{r=1}^{n_t} w_{jtr}^{(i)}}. \tag{4.49}$$

The denominator of (4.49) is the sum of all the weights for model j over the whole batch. This is the posterior maximum likelihood estimate of the number of measurements truly caused by model j. The delta function in the numerator effectively controls the range of the summation in the numerator by selecting only those weights which corresponded to classification measurements with value i. So, the numerator is the posterior MLE of the number of measurements truly caused by model j with classification i. The ratio in (4.49) is thus an estimate of the fraction of measurements due to model j which gave rise to an observed classification i. In other words, the estimator for each confusion matrix entry (which is a probability) is the estimated relative frequency of the associated event. This is an intuitively appealing result.

4.2 Summary of the PMHT-c algorithm

The Probabilistic Multi-Hypothesis Tracker with classification measurements (PMHT-c) can be summarised by the following steps.

1. Initialise the state and parameter estimates, $\mathbf{X}^{(0)}$, $\Pi^{(0)}$ and $\mathsf{C}^{(0)}$. Proper initialisation is crucial since the EM algorithm is a hill climbing approach that can guarantee only local convergence. Since the classification measurements, $\boldsymbol{z}_{tr}^{(k)}$ are observations of the assignments, it is possible to use their probabilities to define $w_{mtr}^{(-1)} = c_{\boldsymbol{z}_{tr}^{(k)} m}$ and form the initial estimates using these weights.

2. Calculate assignment weights $w_{mtr}^{(i)}$ for each measurement r and model m at each time sample t defined by (4.20) as

$$w_{mtr}^{(i)} = \frac{\pi_t^{m(i)} \zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{m(i)}\right) c_{\boldsymbol{z}_{tr}^{(k)} m}}{\sum_{l=1}^{M} \pi_t^{l(i)} \zeta_t^l\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{l(i)}\right) c_{\boldsymbol{z}_{tr}^{(k)} l}}.$$

3. Calculate the refined estimate of the states $\mathbf{X}^{(i+1)}$ using the weights $w_{mtr}^{(i)}$ and the observations $\boldsymbol{z}_{tr}$.

   For the case of linear Gaussian statistics, the state estimate for iteration $i+1$ is found by running a Kalman smoother with the synthetic measurements

$$\tilde{\boldsymbol{z}}_{mt}^{(i)} \equiv \frac{\sum_{r=1}^{n_t} w_{mtr}^{(i)} \boldsymbol{z}_{tr}^{(x)}}{\sum_{r=1}^{n_t} w_{mtr}^{(i)}},$$

   and with a modified measurement covariance given by

$$\tilde{R}_t^{m(i)} = \frac{1}{\sum_{r=1}^{n_t} w_{mtr}^{(i)}} R_t^m.$$

   where R_t^m is the true measurement covariance, i.e. the covariance of the function ζ_t^m.

4. Calculate the refined estimate of the assignment prior, $\Pi^{(i+1)}$, using

$$\pi_t^{k(i+1)} = \frac{1}{n_t} \sum_{r=1}^{n_t} w_{ktr}^{(i)}.$$

5. Refine the estimate of the confusion matrix, $\mathsf{C}^{(i+1)}$ using

$$c_{ij}^{(i+1)} = \frac{\sum_{t=1}^{T} \sum_{r=1}^{n_t} \delta\left(\boldsymbol{z}_{tr}^{(k)} - i\right) w_{jtr}^{(i)}}{\sum_{t=1}^{n_t} \sum_{r=1}^{n_t} w_{jtr}^{(i)}}.$$

6. Calculate the value of the auxiliary function at the new estimates,

$$\begin{aligned} Q\left(\mathbf{X}^{(i+1)}, \mathbf{\Pi}^{(i+1)}, \mathsf{C}^{(i+1)} | \mathbf{X}^{(i)}, \mathbf{\Pi}^{(i)}, \mathsf{C}^{(i)}\right) = \\ \sum_{m=1}^{M} \left\{ \log \psi_0^m \left(\boldsymbol{x}_0^{m(i+1)}\right) + \sum_{t=1}^{T} \log \psi_T^m \left(\boldsymbol{x}_t^{m(i+1)} | \boldsymbol{x}_{t-1}^{m(i+1)}\right) \right\} \\ + \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log \pi_t^{k(i+1)} w_{ktr}^{(i)} \right\} \\ + \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log \zeta_t^k \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k(i+1)}\right) w_{ktr}^{(i)} \right\} \\ + \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{k=1}^{M} \log c_{\boldsymbol{z}_{tr}^{(k)} k}^{(i+1)} w_{ktr}^{(i)} \right\}. \end{aligned}$$

7. Repeat steps 2 ... 6 until $Q\left(\mathbf{X}^{(i+1)}, \mathbf{\Pi}^{(i+1)}, \mathsf{C}^{(i+1)} | \mathbf{X}^{(i)}, \mathbf{\Pi}^{(i)}, \mathsf{C}^{(i)}\right)$ converges.

4.3 Special Cases

Consider two special cases of the PMHT-c filter. In both cases, it is assumed that the PMHT knows the true confusion matrix C.

4.3.1 Uninformative classification measurements

The PMHT-c is a generalised version of the original PMHT. The original PMHT is derived without classification measurements, so it is expected that the PMHT-c will simplify to the original PMHT in the case that the classification measurements are uninformative.

Suppose that the classification measurements were uninformative, that is each model is equally likely to give rise to each class output. In this case, the confusion matrix is constant across rows. That is, $c_{ij} = c_{ik} \forall i, j, k$. Denote c_i as the constant value of the confusion matrix for row i. This situation is equivalent to having no classification measurements at all since the probability of the observation $\boldsymbol{z}_{tr}^{(k)}$ is independent of the value of k_{tr}, i.e. $c_{\boldsymbol{z}_{tr}^{(k)} k_{tr}} = c_{\boldsymbol{z}_{tr}^{(k)}}$.

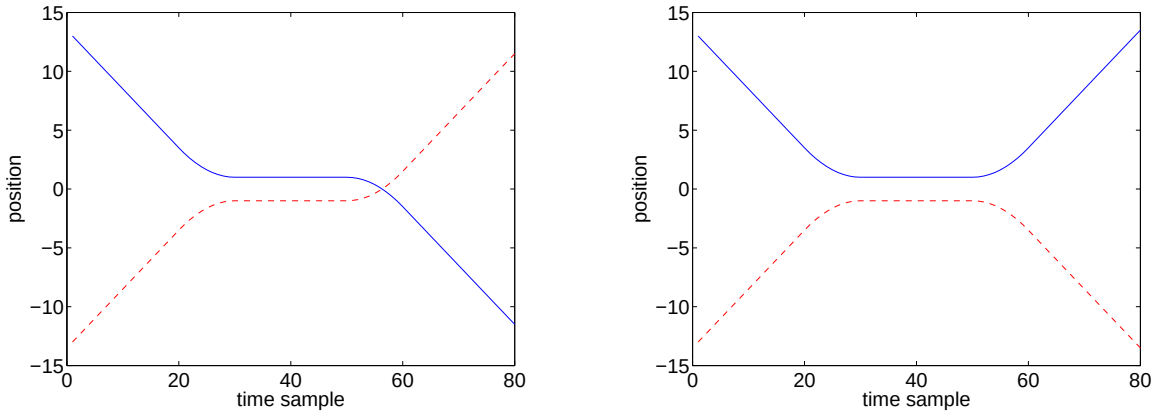
In this case, the weight equation for model m becomes

$$\begin{aligned} w_{mtr}^{(i)} &= \frac{\pi_t^{m(i)} \zeta_t^m \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{m(i)} \right) c_{\boldsymbol{z}_{tr}^{(k)}}}{\sum_{l=1}^{M} \pi_t^{l(i)} \zeta_t^l \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{l(i)} \right) c_{\boldsymbol{z}_{tr}^{(k)}}} \\ &= \frac{c_{\boldsymbol{z}_{tr}^{(k)}} \pi_t^{m(i)} \zeta_t^m \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{m(i)} \right)}{c_{\boldsymbol{z}_{tr}^{(k)}} \sum_{l=1}^{M} \pi_t^{l(i)} \zeta_t^l \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{l(i)} \right)} \\ &= \frac{\pi_t^{m(i)} \zeta_t^m \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{m(i)} \right)}{\sum_{l=1}^{M} \pi_t^{l(i)} \zeta_t^l \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{l(i)} \right)}. \end{aligned}$$

The term due to the classification measurements cancels and the weight equation is the same as that for the standard PMHT. Thus, when the classification measurements are uninformative, the algorithm simplifies to the standard PMHT.

4.3.2 Perfect classification measurements

Another special case of interest is when the output of the classifier is perfect. Suppose there is a separate class for each model and that the classification measurements are error free. In this case, the confusion matrix is the identity matrix and the individual elements may be written as identity functions $c_{ij} = \delta(i - j)$. Intuitively, it is expected that the algorithm should use the classifications as hard associations.



(a) crossing targets (b) turning targets

Figure 4.1: Simulated scenarios

In this case, the weights become

$$\begin{aligned} w_{k_{tr}tr}^{(i)} &= \frac{\pi_t^{k_{tr}(i)} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}(i)} \right) \delta \left(\boldsymbol{z}_{tr}^{(k)} - k_{tr} \right)}{\sum_{l=1}^{M} \pi_t^{l(i)} \zeta_t^{l} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{l(i)} \right) \delta \left(\boldsymbol{z}_{tr}^{(k)} - l \right)} \\ &= \frac{\pi_t^{k_{tr}(i)} \zeta_t^{k_{tr}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{k_{tr}(i)} \right) \delta \left(\boldsymbol{z}_{tr}^{(k)} - k_{tr} \right)}{\pi_t^{\boldsymbol{z}_{tr}^{(k)}(i)} \zeta_t^{\boldsymbol{z}_{tr}^{(k)}} \left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{\boldsymbol{z}_{tr}^{(k)}(i)} \right)} \\ &= \delta \left(k_{tr} - \boldsymbol{z}_{tr}^{(k)} \right). \end{aligned}$$

This weight equation effectively means that the filter no longer uses probabilistic assignment of the observations and models. This should be expected since the perfect classification measurements are providing the filter with the true values of $\mathbf{K}$. The weights are now hard assignments independent of the model states so there is no need for recursive estimation of the states or the probability parameter $\mathbf{\Pi}$. The filter simplifies to independent state estimation of the different models using the assignments provided by the classifier.

4.4 Performance Analysis

The improvement in tracking performance obtainable by using classification measurements is now illustrated through some simulated examples. These examples focus on estimation performance in a manoeuvring target setting. Two different scenarios are considered. In each scenario, there are two targets that follow one dimensional trajectories that are closely spaced, making for a difficult tracking problem. In the first scenario, the targets move together and then cross over before eventually diverging. In the second scenario, the targets move together and then turn back. These two scenarios are shown in figure 4.1. The trajectory of target 1 is shown as a solid line and the trajectory of target 2 is shown as a dashed line.

In both cases, the targets are simulated without clutter and a single measurement is observed in each scan. The measurement is taken from either target with equal probability.

The sensor receives measurements of position and classification measurements. The classification measurements are an observation of the true source of the position observation. The confusion matrix for the classification measurements is of the form

$$\mathsf{C} = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\alpha & \alpha \end{bmatrix}, \tag{4.50}$$

where α is the probability that the correct classification is reported. α will be referred to as the classification veracity. The scenarios are tested with various values of α and the performance observed. For the case of $\alpha = 1$, the classification measurements are perfect and the filter performance will be the best possible for the scenario. Thus, the case of $\alpha = 1$ provides a performance bound. For the case of $\alpha = 0.5$, the classification measurements provide no information and the PMHT-c simplifies to the standard PMHT. Thus, the performance when $\alpha = 0.5$ shows the standard PMHT performance on the scenario. The cases of known and unknown C were both considered.

4.4.1 Simulation Details

The particulars of the simulations used for performance evaluation are now presented. For each combination of parameters, 1000 Monte Carlo realisations are generated and the filter performance is gauged by averaging over these realisations.

4.4.1.1 Target Trajectories

In both scenarios, the target trajectories are composed of constant velocity and constant acceleration segments. This means that the position is a piecewise combination of linear and quadratic functions. The trajectory of target 2 is a mirror image of the target 1 trajectory. The difference between the turn scenario and the crossing scenario is the sign of the acceleration in the second constant acceleration segment.

For the crossing scenario, the true state of target 1 is given by:

$$\dot{r}^1_\tau = \begin{cases} -v & 0 \le \tau < 20 \\ -v + \frac{v}{10}(\tau - 20) & 20 \le \tau < 30 \\ 0 & 30 \le \tau < 50 \\ -\frac{v}{10}(\tau - 50) & 50 \le \tau < 60 \\ -v & 60 \le \tau < 80, \end{cases} \tag{4.51}$$

$$\begin{aligned} r^1_\tau &= r^1_0 + \int_0^\tau \dot{r}^1_t \mathrm{d}t \\ &= \begin{cases} r^1_0 - v\tau & 0 \le \tau < 20 \\ r^1_0 - 20v - \frac{v}{20}(\tau - 20)^2 & 20 \le \tau < 30 \\ r^1_0 - 25v & 30 \le \tau < 50 \\ r^1_0 - 25v - \frac{v}{20}(\tau - 50)^2 & 50 \le \tau < 60 \\ r^1_0 - 30v - v(t - 60) & 60 \le \tau < 80 \end{cases}, \end{aligned} \tag{4.52}$$

where the constants $v = 0.5$ and $r^1_0 = 1 + 25v = 13.5$ were chosen. The function given in (4.52) defines the solid curve shown in figure 4.1(a).

4.4.1.2 Measurements

At each scan, a single measurement is observed, $n_t = 1$. The scans are taken at regular intervals with a sampling rate of 1. This means that $\tau_t = t$ and $\tau_t - \tau_{t-1} = 1$. A batch

length of $T = 80$ is used. The measurement at each scan is taken from either target with equal probability, so $\pi_t^m = 0.5$. Π is assumed to be known by the tracker.

Each measurement is the current position of one of the targets corrupted with white Gaussian noise, with variance $\mathsf{R} = 1$.

The classification measurement at each scan is generated according to the confusion matrix, C, as defined in (4.50). The classification veracity, α, is varied.

4.4.1.3 Target Models

The tracker uses a one dimensional almost constant velocity model[1] for the targets. This means that the motion is assumed to be constant in velocity with Gaussian perturbations. In the one dimensional case, these perturbations are scalars with zero mean and variance Q. This variance is assumed to be constant and the same for both targets. A value of $\mathsf{Q} = (v/10)^2 = 0.0025$ was chosen so that the filter would give an adequate dynamic response to the acceleration segments. The target state vector is

$$\boldsymbol{x}_t^m \equiv \begin{bmatrix} r_t^m \\ \dot{r}_t^m \end{bmatrix}. \tag{4.53}$$

The true measurement process is linear and Gaussian. It is assumed that the filter knows the measurement process. This measurement process is, in the Gaussian position only case, described in 3.3. So, the measurement probability density used by the filter is

$$\zeta^m\left(\boldsymbol{z}_t^{(x)}|\boldsymbol{x}_t^m\right) = \frac{1}{\sqrt{2\pi\mathsf{R}}}\exp\left\{-\frac{1}{2\mathsf{R}}\left(\boldsymbol{z}_t^{(x)} - \mathsf{H}\boldsymbol{x}_t^m\right)^2\right\}, \tag{4.54}$$

where $R = 1$ is a scalar, and $H = [1, 0]$.

The filter knows that the confusion matrix, C, takes the form given in (4.50). Both cases of known and unknown classification veracity, α, are considered. When the value of α is unknown, the confusion matrix must be estimated. However, the filter does know the form of the matrix, which can be exploited by incorporating the constraint $c_{11} = c_{22}$. When this constraint is incorporated into the iterative maximisation of Q_C, then the estimator for α is given by

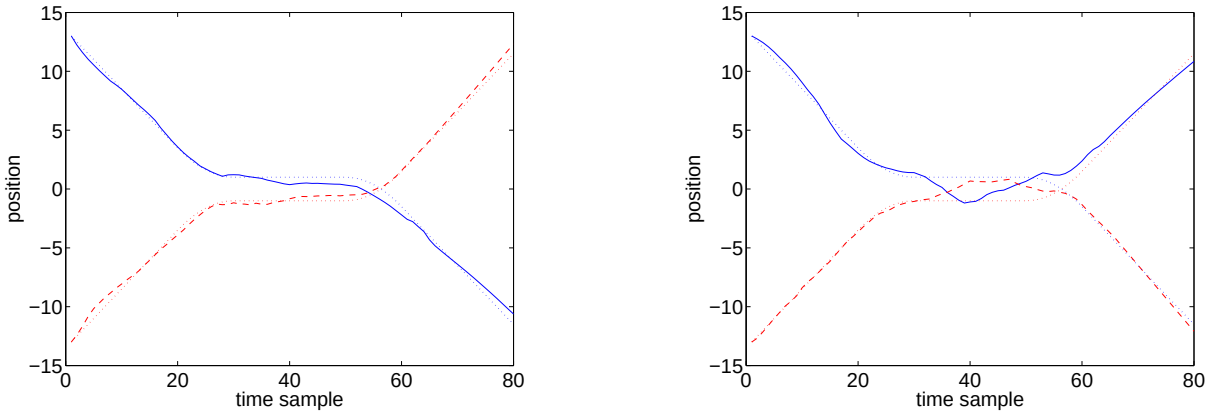
$$\hat{\alpha}^{(i+1)} = \frac{1}{T}\left\{\sum_{t=1}^{T} w_{1t}^{(i)}\delta\left(\boldsymbol{z}_t^{(k)} - 1\right) + w_{2t}^{(i)}\delta\left(\boldsymbol{z}_t^{(k)} - 2\right)\right\}. \tag{4.55}$$

The target state estimates for the filter are initialised by using the correct initial state and projecting a constant velocity path from this point. This scheme is unrealistic in practice, but it avoids irrelevant track initiation issues. Track initiation is considered later in chapter 6. The prior state distribution $\psi_m(\boldsymbol{x}_0^m)$ is Gaussian centred on the correct initial state with covariance matrix $P_0 = \mathrm{diag}(R, 0.5)$.

4.4.2 Performance Metrics

The performance of the PMHT-c is quantified by estimation accuracy for $\mathbf{X}$ and C as α is varied. An additional measure of performance is the probability that the filter follows the target trajectories properly. Since the measurements make the target paths ambiguous, it is possible that the filter may lose the targets, or may switch tracks. Figure 4.2 shows examples of trials with correct and incorrect tracking. In the left plot, the filter has correctly identified the crossing trajectories, but in the right plot, it assigns measurements from target 2 to the track for target 1 and so incorrectly produces turning trajectories.

[1] refer to section 3.3 for more details on the almost constant velocity model



(a) crossing trial, correct tracking (b) crossing trial, incorrect tracking

Figure 4.2: Examples of simulated trials

The probability of correctly following the trajectories is estimated by counting the number of random trials where both of the state estimates remain within a prescribed distance of the respective true states. If the tracks move farther away from the true state than this distance at any stage, then the trial is declared incorrectly tracked. A gate distance of 3 units was used.

The estimation performance for $\mathbf{X}$ is quantified by the bias and variance of the estimator output. These statistics are calculated using their sample estimates for trials with correct tracking. The estimation of target 1 is investigated and it is assumed that similar results would be obtained for target 2 (particularly given the symmetrical nature of the geometry). Spot checks confirm this assumption.

The instantaneous position bias is estimated from the simulations by using the sample mean of the estimates. That is,

$$\hat{B}_t^r = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{r}_t^{1(n)} - r_t^1\right), \tag{4.56}$$

where $\hat{r}_t^{1(n)}$ is the position estimate for track 1 at scan t from the nth correctly tracked random trial. N is the total number of trials with correct tracking.

There are two sources of estimation bias. Firstly, the model used by the PMHT-c does not properly match the target dynamics. The true target model switches between constant velocity to constant acceleration during the scenario. The tracker is attempting to model the trajectory using an almost constant velocity model, which means it will be inclined to estimate a trajectory with gradual changes in speed. During the periods of acceleration, the model is mismatched and the tracker models the manoeuvre as a more gradual acceleration over a longer time period. This leads to estimator bias. The second source of bias is the presence of measurements from the other target. The measurements truly due to target 2 will be given a non-zero assignment weight for target 1, and will cause a bias towards target 2. This bias depends on the target geometry because the association probability will be very low when the targets are widely separated. If the tracker were able to perfectly assign measurements, then the second target would cause no bias. These two sources of bias will be referred to as mismatch bias and assignment bias. The biases may counteract each other. For example, in the turn scenario, the mismatch bias on target 1 is positive (the expected value of the estimate is above the true

trajectory) but the assignment bias is negative. In such a case, the mismatch bias may become masked if the assignment bias is high (when the classifier veracity is low). The biases are not inherent deficiencies of the PMHT approach, but inevitable errors that any estimator would encounter using the above target models.

The instantaneous position variance is estimated from the simulations by using the sample variance. That is,

$$\hat{\sigma}_t^r = \frac{1}{N}\sum_{n=1}^{N}\left[\hat{r}_t^{1(n)} - \left(\hat{B}_t^r + r_t^1\right)\right]^2. \tag{4.57}$$

Estimator variance is caused by the randomness of the measurements. When the variability of the measurements assigned to a track is high, then the estimator variance is high. Such a condition is encountered when the two targets are close together. If the tracker performed perfect assignment (such as when the classifier veracity is unity), then the measurement variability would always be the target measurement noise, and the estimator variance would be constant with time.

The estimation of the confusion matrix is quantified by examining the bias and variance of the estimator $\hat{\alpha}$ obtained when the PMHT-c converges. These are estimated using their sample values, namely

$$\hat{B}^\alpha = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{\alpha}^{(n)} - \alpha\right), \tag{4.58}$$

$$\hat{\sigma}^\alpha = \frac{1}{N}\sum_{n=1}^{N}\left[\hat{\alpha}^{(n)} - \left(\hat{B}_t^\alpha + \alpha\right)\right]^2. \tag{4.59}$$

The bias and variance of the estimator $\hat{\alpha}$ are not time varying because $\hat{\alpha}$ produces one estimate for the whole batch.

For each scenario, the number of trials with correct tracking is an estimate of the true probability that the scenario is correctly tracked for a particular value of α. Let us denote the true probability of correctly tracking the scenario as ρ. The estimator, $\hat{\rho}$, used to form the tables of results is

$$\hat{\rho} = \frac{1}{N}\sum_{i=1}^{N} d_i, \tag{4.60}$$

where d_i is the result of the i^{th} trial (zero or unity) and N is the number of trials, i.e 1000.

It can be shown that the variance of this estimator is given by

$$\text{var}\ (\hat{\rho}) = \frac{1}{N}\left(\rho - \rho^2\right). \tag{4.61}$$

This is maximized when $\rho = 0.5$. If $\rho = 0.5$ then the variance of $\hat{\rho}$ using 1000 samples is 2.5×10^{-4}. Thus, the standard deviation of the results given in tables 4.1 ... 4.4 is less than 0.016, i.e. 16 trials.

4.4.3 Improvement with known confusion matrix

First investigate the performance gain obtained by using classification measurements when the confusion matrix C is known. The number of trials with correct tracking for various values of α is given in table 4.1.

	classification veracity, α					
	0.5	0.6	0.7	0.8	0.9	1
crossing targets	634	613	694	863	972	1000
turning targets	115	355	695	911	987	1000

Table 4.1: Number of trials with correct tracking with known confusion matrix

As would be expected, the number of trials with correct tracking generally increases as the classification veracity improves. What is surprising is how much improvement is obtained even for very inaccurate classifications.

The standard PMHT (namely the PMHT-c with $\alpha = 0.5$) performs poorly on the turn scenario because the filter is predisposed to the crossing explanation. The filter tends to overshoot, causing the tracks to cross in the stationary part of the scenario. Since the tracks have already crossed, it is unlikely that they will cross again, which is required for the turn output. This behaviour can be seen from the estimator bias at the middle of the scenario (refer to figure 4.5). The bias that causes the overshoot is the assignment bias, which reduces as the classification veracity improves. The assignment bias actually improves the performance for the crossing scenario. Table 4.1 shows that the performance of the PMHT-c is better for $\alpha = 0.5$ than for $\alpha = 0.6$ on the crossing scenario. This is because the assignment bias for the filter predisposes the filter to the crossing case.

The position estimation bias for the crossing scenario is plotted in figure 4.3 for various values of α. The variance for the crossing scenario is plotted in figure 4.4. Figures 4.5 and 4.6 show the bias and variance for the turning scenario respectively.

The algorithm uses smoothing which means that measurements from earlier time samples and measurements from later time samples contribute to the state estimate. Because the smoother does not have these extra measurements near the end of the batch, the variance rises towards the end. At the beginning of the batch, the variance is very low. This is due to the initialisation which uses the true value of the target state. This behaviour is common for all classification veracity values, α. When $\alpha = 1$, the true source of each measurement is known, and the variance is roughly constant over the rest of the batch. There is a small rise in variance during the target manoeuvres, where the assumed model does not fit the true dynamics, and other fluctuations are due to the finite ensemble size. For other values of α, the variance peaks around the middle of each scenario. This is when the two targets are close together, and there is most ambiguity in the assignment of measurements. The height of this peak increases as α decreases. In the turning scenario, the peak of the variance is roughly symmetrical in time, but it is not in the crossing scenario. This is because the target trajectories are symmetrical in the turning scenario, but not in the crossing one.

As discussed in section 4.4.2, there are two sources of estimator bias, namely the mismatch of the target dynamics and the assumed model dynamics during target manoeuvres, and incorrectly associated measurements from the other target. The first source is most clearly seen in the curves for perfect classification, $\alpha = 1$. In this case, there is no association errors, and the only source of bias is the model mismatch. The algorithm assumes an almost constant velocity model, but the true trajectory follows a constant acceleration law during the manoeuvres. This means that the algorithm over-smoothes the estimated trajectory, rounding off the corners and introducing bias towards the inside of the turn. For the crossing scenario, this means a positive bias for the first manoeuvre, and a neg-

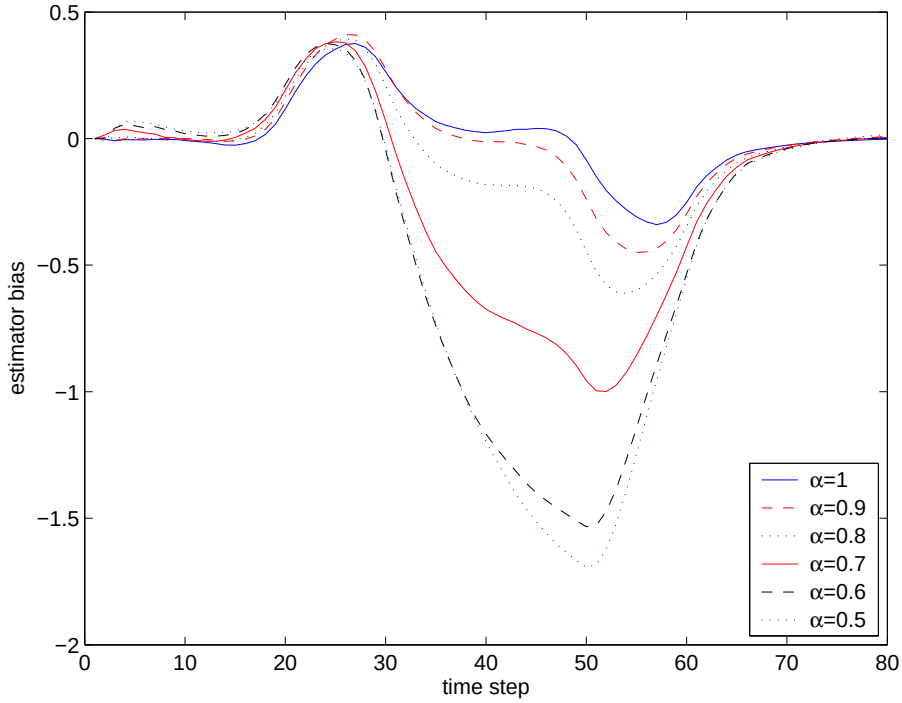


Figure 4.3: Bias of estimator $\hat{r}_t^1$, C known, crossing scenario

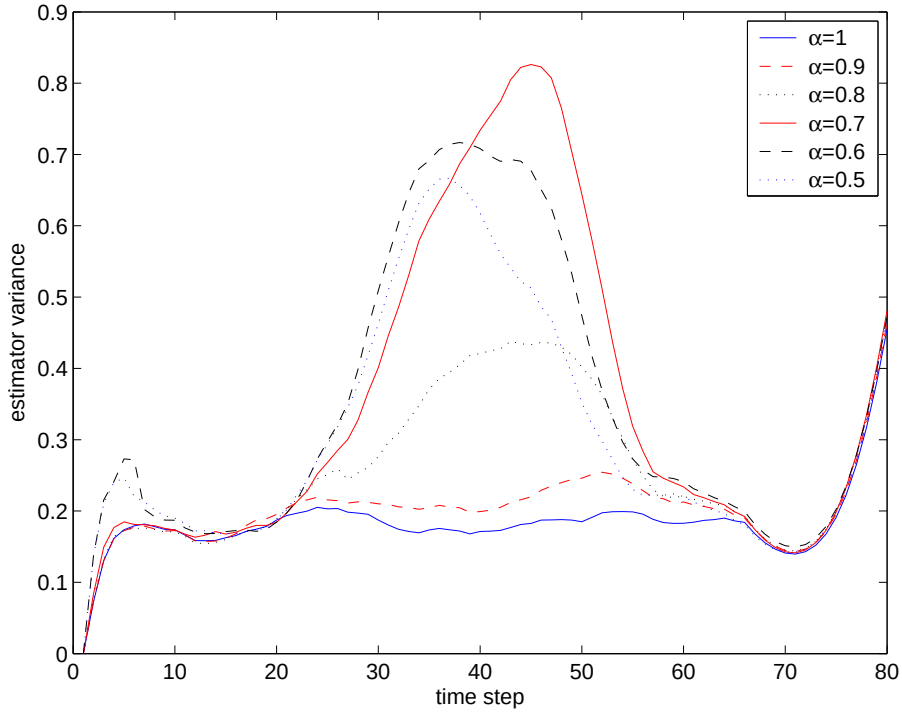


Figure 4.4: Variance of estimator $\hat{r}_t^1$, C known, crossing scenario

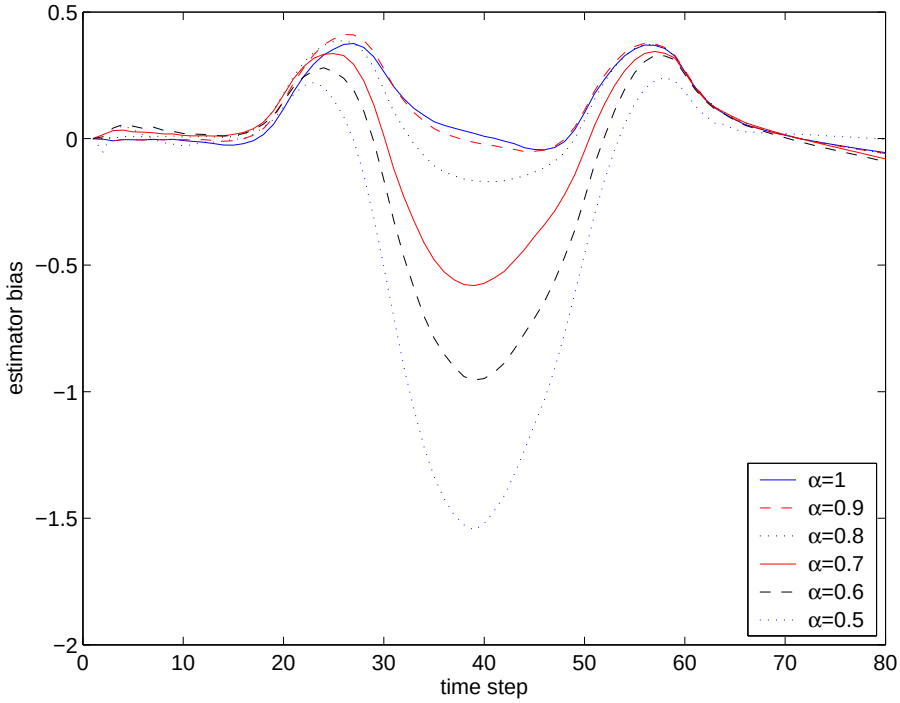


Figure 4.5: Bias of estimator $\hat{r}_t^1$, C known, turning scenario

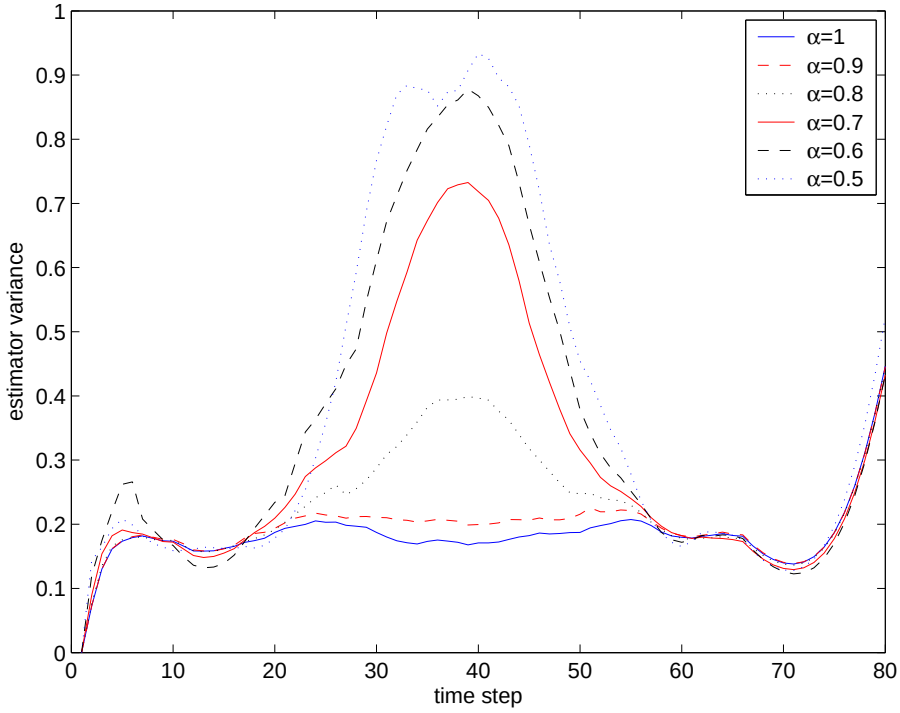


Figure 4.6: Variance of estimator $\hat{r}_t^1$, C known, turning scenario

ative bias for the second, whereas positive bias is produced for both manoeuvres in the turning scenario, as is expected due to the symmetry. The second source of bias has an increasing influence as the classifier veracity decreases. This bias is maximised when the targets are closest together, at the middle of each scenario. Incorrectly associated measurements bias each state estimate towards the other target, so the bias is negative for target 1. When this bias counteracts the mismatch bias, such as during both manoeuvres for the turning scenario, the overall bias may be reduced at that instant in time. Thus the $\alpha = 0.5$ case gives less bias than $\alpha = 1$ during the manoeuvre, but more in the constant velocity sections. Like the variance, the crossing scenario bias is asymmetrical because of the asymmetrical target trajectories.

Generally, both the bias and variance of the state estimate increase as the classification accuracy reduces. Although there is significant improvement from $\alpha = 0.7$ to $\alpha = 1$, the results for low veracity classifications are still important. These show that with fairly vague information, significant improvement in performance can be obtained over the standard PMHT algorithm.

4.4.4 Improvement with unknown confusion matrix

The performance of the PMHT-c algorithm is now examined for the case where the classification veracity is unknown. As for the known confusion matrix case, the number of trials with correct tracking, the state estimator bias, and the state estimator variance provide performance measures. In addition to these, the bias and variance of the estimated veracity $\hat{\alpha}$ are also used. The PMHT-c knows the form of the confusion matrix, but does not know the particular value of the classification veracity, α. The confusion matrix parameter is estimated using (4.55)

$$\alpha^{(i+1)} = \frac{1}{T}\left\{\sum_{t=1}^{T} w_{1t}^{(i)}\delta\left(\boldsymbol{z}_t^{(k)} - 1\right) + w_{2t}^{(i)}\delta\left(\boldsymbol{z}_t^{(k)} - 2\right)\right\}.$$

The estimated value for α is a function of the assignment weights and relies on the PMHT-c to be able to determine the true source of the measurements. When the weights are exactly correct (i.e. zero or unity corresponding to the true measurement source) then (4.55) is the optimal estimator for α. If the weights deviate from their correct values, the estimate becomes degraded. During the middle time of both scenarios, there is much confusion over the true source of measurements and the weights for both targets tend to be close. Under this condition, the estimate of α is poor. However, at the start and end of the scenarios, the targets are well separated and the weights will converge to zero or unity, giving a good estimate of α. To ensure that the estimator is not corrupted by the confused middle segment of the scenarios, the estimate of α is only calculated when the larger of the target weights is more than 0.8.

A further benefit obtained by estimating the confusion matrix is an automatic check for correct tracking. If α is estimated from the data at the end of the scenario, then it is expected that the estimate would be less than 0.5 if the tracks have swapped targets. If the estimate is unbiased, then the mean of $\hat{\alpha}$ would be $1 - \alpha$. This could be used to implement an ad hoc rule to correct for the incorrectly tracked scenarios. However, no such rule has been used in this analysis.

The number of trials with correct tracking for the two scenarios is given in table 4.2. These results are compared with the standard PMHT and the PMHT-c when the confusion matrix is known. The unknown α performance is somewhat poorer than when the confusion matrix is known, but the results are still much better than the standard PMHT.

	α					
	0.5	0.6	0.7	0.8	0.9	1
crossing targets						
standard PMHT	634	634	634	634	634	634
PMHT-c α known	634	613	694	863	972	1000
PMHT-c α estimated	589	612	727	858	940	1000
turning targets						
standard PMHT	115	115	115	115	115	115
PMHT-c α known	115	355	695	911	987	1000
PMHT-c α estimated	210	343	636	858	926	1000

Table 4.2: Number of trials with correct tracking with estimated confusion matrix

Figures 4.7 and 4.8 show the bias and variance of the PMHT-c when the classification veracity is estimated on the turn scenario. Similar results are obtained for the crossing scenario and are not shown here. The plots also show the statistics for the standard PMHT (i.e. $\alpha = 0.5$), and the PMHT-c when the classification veracity is known. There is very little degradation in performance between the PMHT-c curves for known and unknown classifier veracity.

From these results, it appears that the PMHT-c does not suffer from significant performance degradation when the confusion matrix is unknown.

4.4.4.1 Confusion matrix estimation performance

As stated earlier, the estimation performance of the classification veracity, α, is quantified by the bias and variance of its estimator. These quantities are plotted in figures 4.9 and 4.10 respectively. The figures show the bias and variance for the turn and cross scenarios. When the targets are incorrectly tracked for a particular trial, the information used to estimate α is systematically in error since the tracks follow the wrong targets. This bias tends to make the estimated α closer to 0.5 since the incorrect segments cancel out the correct segments. Hence, the bias and variance are calculated using only the trials with correct tracking.

The estimator bias is approximately symmetrical about the point $\alpha = 0.5$. This is to be expected. If the tracker were to take the opposite of the classification measurement, then it would be correct with probability $1 - \alpha$, so the veracities α and $1 - \alpha$ are essentially equivalent. For both scenarios, the bias tends to push the estimated veracity away from 0.5. This may be because the estimator only uses those measurements where polarised weights are obtained. If the true states are fairly close, then a correctly classified measurement may be given a strong weight when an incorrectly classified one is given an indecisive one. This causes slightly more correctly classified measurements to be counted in the estimator than is representative. When α is less than 0.5, the converse occurs. There is a small positive average bias.

The estimator variance is also approximately symmetrical about $\alpha = 0.5$ and is highest at $\alpha = 0.5$ when the classifications provide the least information. For a binomial process

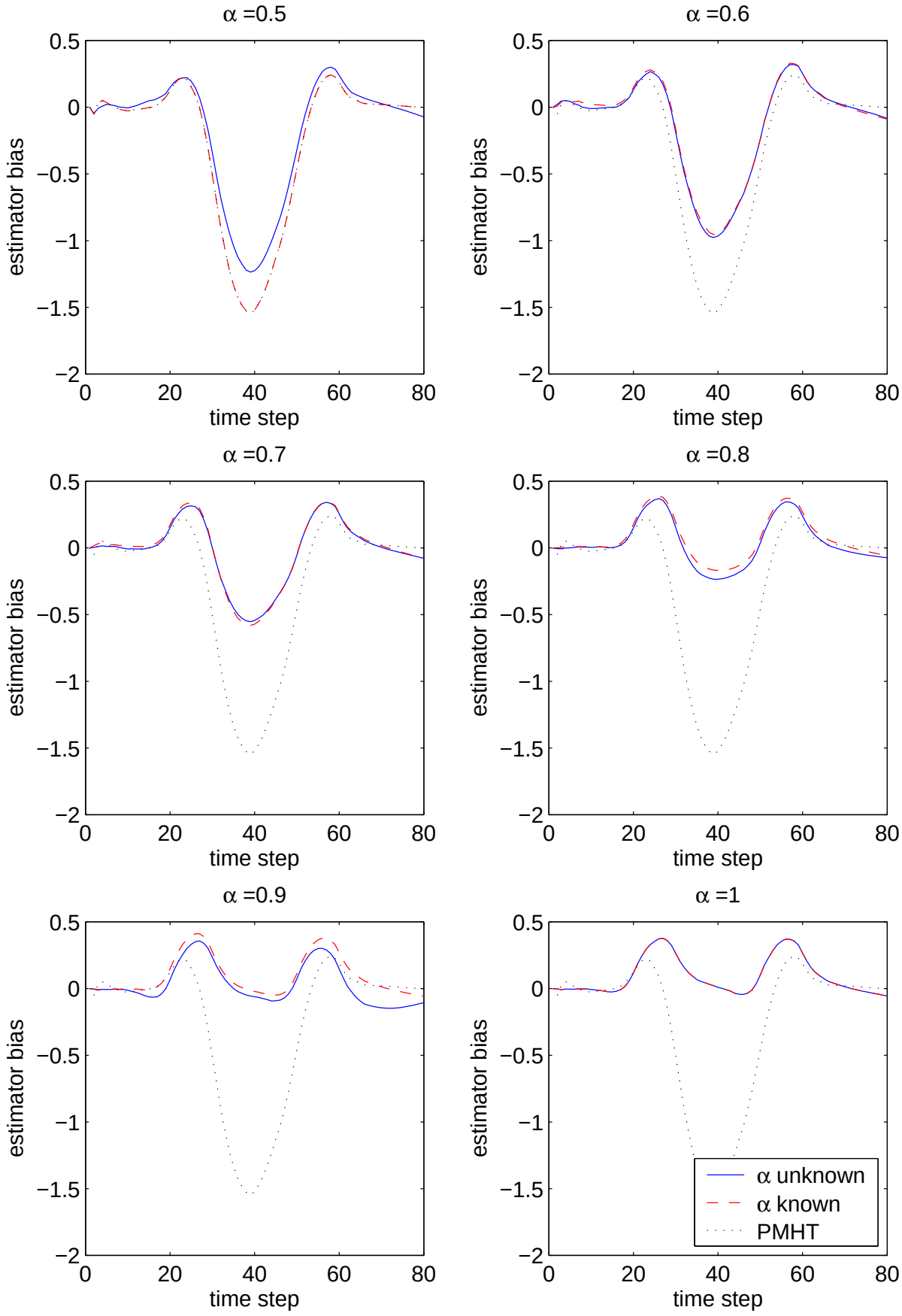


Figure 4.7: bias of estimator $\hat{r}_t^1$, C estimated, turning scenario

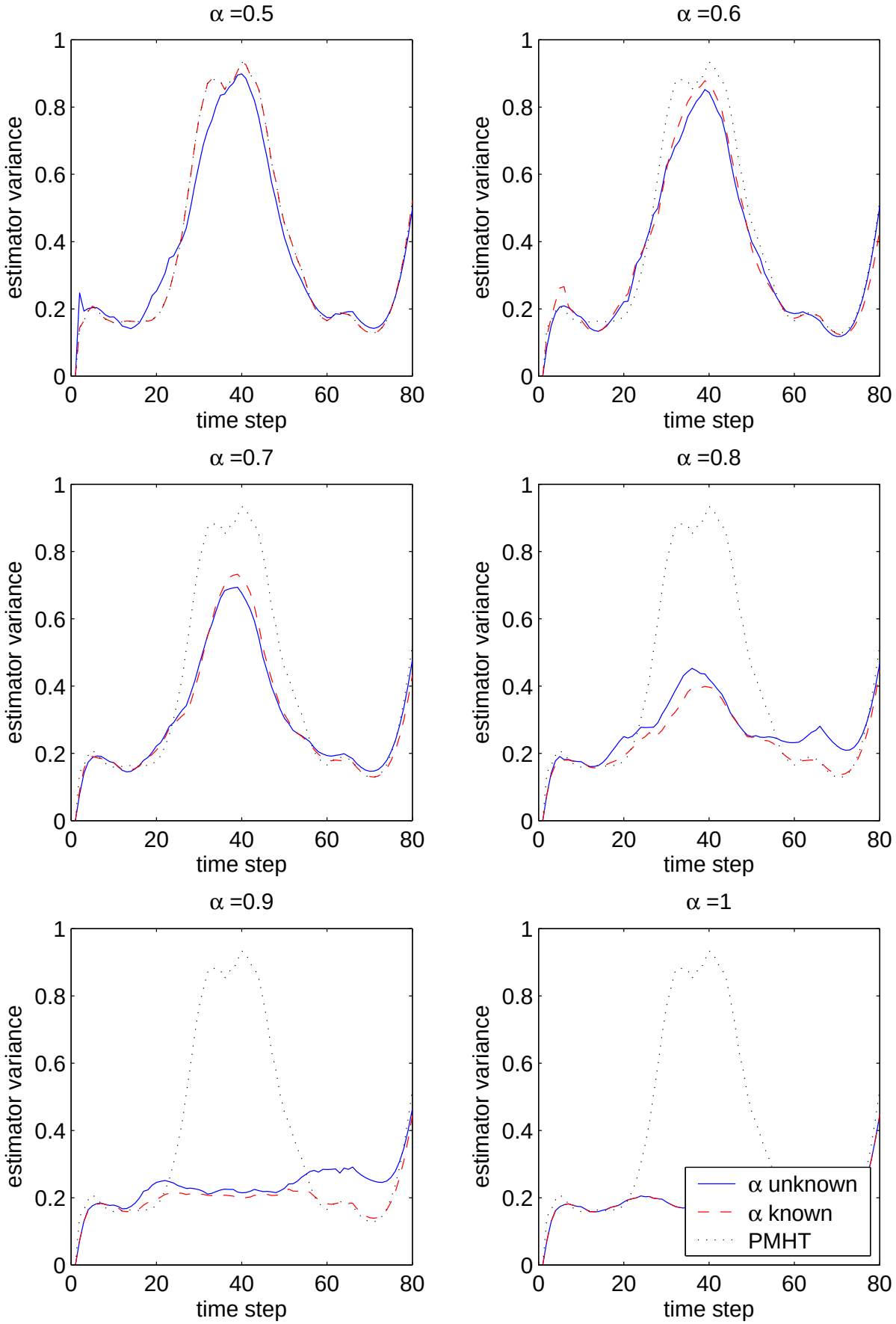


Figure 4.8: variance of estimator $\hat{r}_t^1$, C estimated, turning scenario

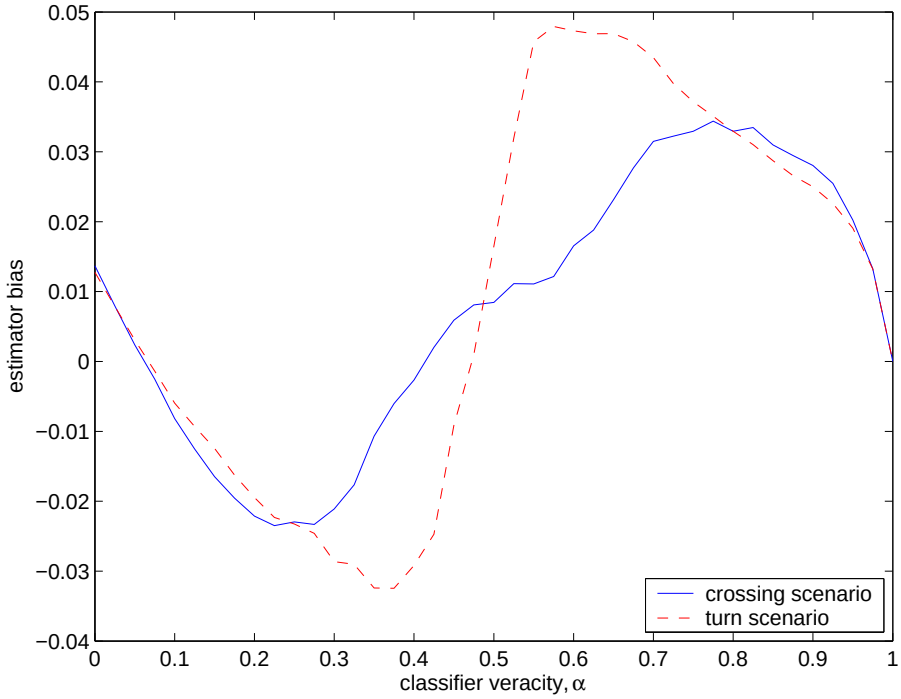


Figure 4.9: Bias of the estimator, $\hat{\alpha}$

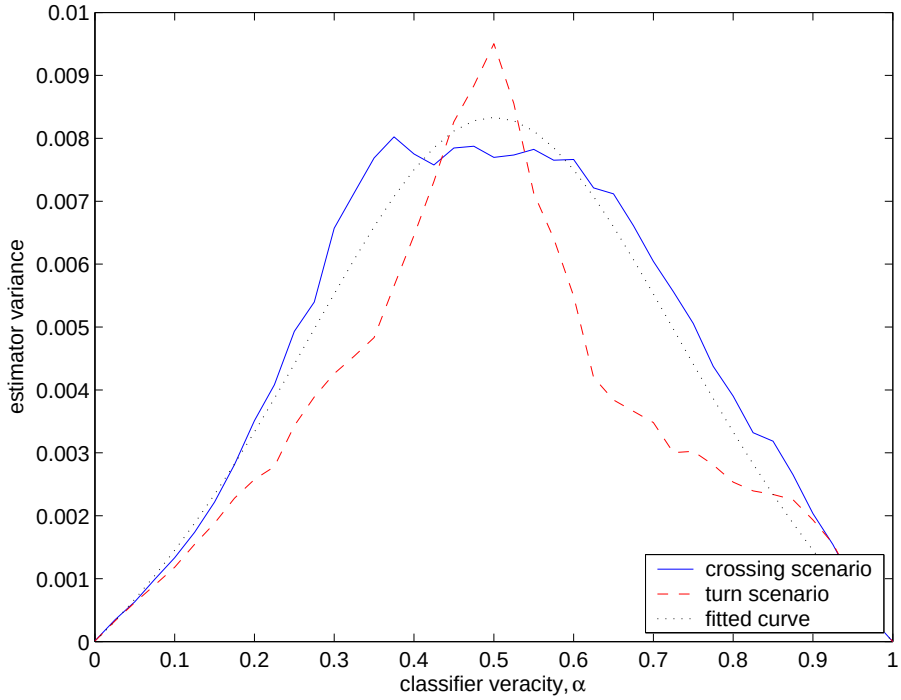


Figure 4.10: Variance of the estimator, $\hat{\alpha}$

with parameter ρ, the variance of the estimator of ρ is given by

$$\text{var}(\hat{\rho}) = \frac{\rho - \rho^2}{N},$$

where N is the number of data points. The classifications are a binomial process, but they are indirectly observed. Since the estimator $\hat{\alpha}$ only uses those classification measurements where the weights are polarised, the number of data points that contribute to the estimator varies with the classifier veracity. When the veracity is very high (or equivalently very low) then most of the data points are used, and N is approximately the batch length, 80. When the veracity is close to 0.5, then many of the measurements have indecisive weights and the number of data points used for the estimator is much lower. So the variance of the estimated veracity, $\hat{\alpha}$, should be

$$\text{var}(\hat{\alpha}) = \frac{\alpha - \alpha^2}{N(\alpha)}, \tag{4.62}$$

where $N(\alpha)$ is the expected number of data points that contribute to the estimator. For the crossing scenario, an empirically derived function,

$$N(\alpha) = 80 - 200\alpha(1 - \alpha), \tag{4.63}$$

was found to agree well with the observed variance. This function corresponds to all 80 batch points contributing to the estimator at extreme α and only 30 points contributing when $\alpha = 0.5$. The quadratic transition models the experimental results adequately. This fitted variance curve is also plotted in figure 4.10.

The variance is generally lower for the turn scenario than the crossing scenario because in the crossing scenario, the targets remain close to each other for a longer portion of the batch. This means there are less scans where the targets are well separated. The result is that the number of measurements that contribute to the veracity estimator, $N(\alpha)$, is generally higher for the turn scenario, giving a lower variance.

4.5 Sensitivity of the PMHT-c to an Incorrectly Assumed Confusion Matrix

The results shown in section 4.4.3 assume that the PMHT-c knows the true classifier confusion matrix, and in section 4.4.4 the confusion matrix is unknown. In general, it is likely that the confusion matrix assumed to be correct is (at least slightly) in error. In this section, the PMHT-c is analysed for the case where the assumed confusion matrix is incorrect. In particular, the PMHT-c will assume a classifier veracity that is not the same as the true classifier veracity, α. The assumed veracity will be denoted β. Thus the PMHT-c assumes that the confusion matrix is $\mathsf{C} = \begin{bmatrix} \beta & 1-\beta \\ 1-\beta & \beta \end{bmatrix}$.

The crossing and turning target scenarios are repeated for a variety of true veracities, α, and assumed veracities, β. As previously, the performance is specified using the bias and variance of the estimate for the position of target 1, and by the number of trials with correct tracking.

The number of trials with correct tracking for various α and β is given in table 4.3 for the crossing trial, and table 4.4 for the turning trial. The table also includes the results obtained in section 4.4.4 when the confusion matrix is estimated. Each column of tables

	true α					
assumed β	0.5	0.6	0.7	0.8	0.9	1.0
0.5	634	634	634	634	634	634
0.6	573	613	643	686	717	725
0.7	547	641	694	784	852	906
0.8	529	654	772	863	931	982
0.9	503	690	835	928	972	999
1.0	0	7	60	207	526	1000
adapted	589	612	727	858	940	1000

Table 4.3: Crossing target scenario

	true α					
assumed β	0.5	0.6	0.7	0.8	0.9	1.0
0.5	115	115	115	115	115	115
0.6	244	355	485	612	723	851
0.7	355	511	695	844	936	984
0.8	442	595	768	911	974	1000
0.9	465	643	818	941	987	1000
1.0	0	1	30	169	479	1000
adapted	210	343	636	858	926	1000

Table 4.4: Turning target scenario

4.3 and 4.4 corresponds to a particular true classification veracity, α. Each row shows how the performance varies when a particular confusion matrix is assumed and the true classification veracity changes.

All of the entries in the first row of table 4.3 are the same, and similarly for table 4.4. This is because this row corresponds to the PMHT-c assuming that the confusion matrix is uniform and that the classification measurements are useless. In this case, the PMHT-c simplifies to the standard PMHT and the performance is identical, independent of classification accuracy, since the classifications are ignored.

The row corresponding to $\beta = 1.0$ gives very bad performance except when the classifications are perfect, or almost so. This is because the algorithm assumes that the classifications are perfect and uses them blindly to assign the measurements. This means that any false classifications will cause measurements from the wrong target to be assigned to a track and the state error will become larger the farther apart the targets move.

What is quite remarkable is the trend shown in both tables: performance is improved the more accurate the classifications are assumed to be, provided they are not assumed to be perfect. For almost every value of classification veracity, the assumed veracity $\beta = 0.9$ gives the best performance. This result is completely counter intuitive, since one would expect the best performance to occur when the algorithm assumes the correct classification veracity. This result also leads to the interesting conclusion that, if the PMHT-c is provided with the true classification veracity, it is better off not using it! These surprising results are explained by detailed examination of the algorithm in the following section.

The performance even improves with higher β when the classification measurements

are uninformative, in the turn scenario. This occurs essentially due to the predisposition of the PMHT to favour the crossing explanation. It would be fairly straightforward to generate four short tracks on either scenario (two at the start and two at the end). The task of determining the target trajectories would then be equivalent to matching the tracks from the start and the end. This could be done by picking a single classification measurement from the start of the batch and one from the end. For the uninformative case, this would give the correct track segment association half of the time - the same as an arbitrary decision. The uninformative classifications do not improve the situation, however, the performance of the PMHT is much worse than 50% for the turning scenario because of the estimator bias. So, assuming that these uninformative measurements actually have high veracity does not improve the solution over what might have been possible using segments of tracks. Rather, it reduces the impact of the PMHT's predisposition to choose the crossing case. This effect is the same for the crossing scenario and so the performance is degraded (the bias enables the correct decision).

For various mismatched conditions, the state estimation bias and variance are plotted in figure 4.11 and figure 4.12 respectively, for the turning scenario. Similar results are obtained for the crossing scenario, and are not shown. Each individual plot in these figures shows the performance for a single true confusion matrix, α, as the assumed confusion matrix, β, varies. There is no curve shown for $\beta = 1$ since assuming this confusion matrix very rarely gives correct tracking. The bias and variance obtained when α is estimated are also shown. In each case, the performance when α is estimated is similar to the performance when the correct value is obtained (as has already been demonstrated with figures 4.7 and 4.8). The number of trials with correct tracking generally improved when a larger value of β was assumed, and figures 4.11 and 4.12 demonstrate that the state estimation performance follows a similar trend. The curves corresponding to $\beta = 0.5$ are independent of the true veracity, because this is where the algorithm has assumed the classifications to be uninformative and ignores them, i.e. the standard PMHT.

The state estimation bias for varying α when $\beta = 0.7$ is plotted in figure 4.13, and the variance in figure 4.14. Both the bias and variance are reduced as the veracity of the classifications improves (α increases) even though the assumed confusion matrix is the same. Similar plots are obtained for other assumed values of β, except $\beta = 0.5$ when the classifications are ignored. This indicates that improving the quality of the supplied data always improves the quality of the track output, which is a desirable result.

4.5.1 Improved Performance with Mismatch

The most intriguing conclusion of the previous section, is that the performance of the PMHT-c improves when it assumes the classification measurements are very accurate, even if they are not. This would seem to imply that it is better to always assume a high probability of correct classification, even if it is known that this assumption is invalid. This is counter-intuitive, but can be understood if the functioning of the algorithm is examined closely.

The key is in the function of the assignment weights. A simplified example will be considered. The conclusions drawn from this example will explain the performance improvements observed with high values of β.

Consider one scan with a single measurement, where the target state estimates from the previous EM iteration are $\hat{\boldsymbol{x}}_1 = -0.5$ and $\hat{\boldsymbol{x}}_2 = 0.5$. The true target states are $\boldsymbol{x}_1 = -1$ and $\boldsymbol{x}_2 = 1$, and the measurement functions are both centred on the target state with unity variance. For the standard PMHT, the assignment weight is simply a normalised

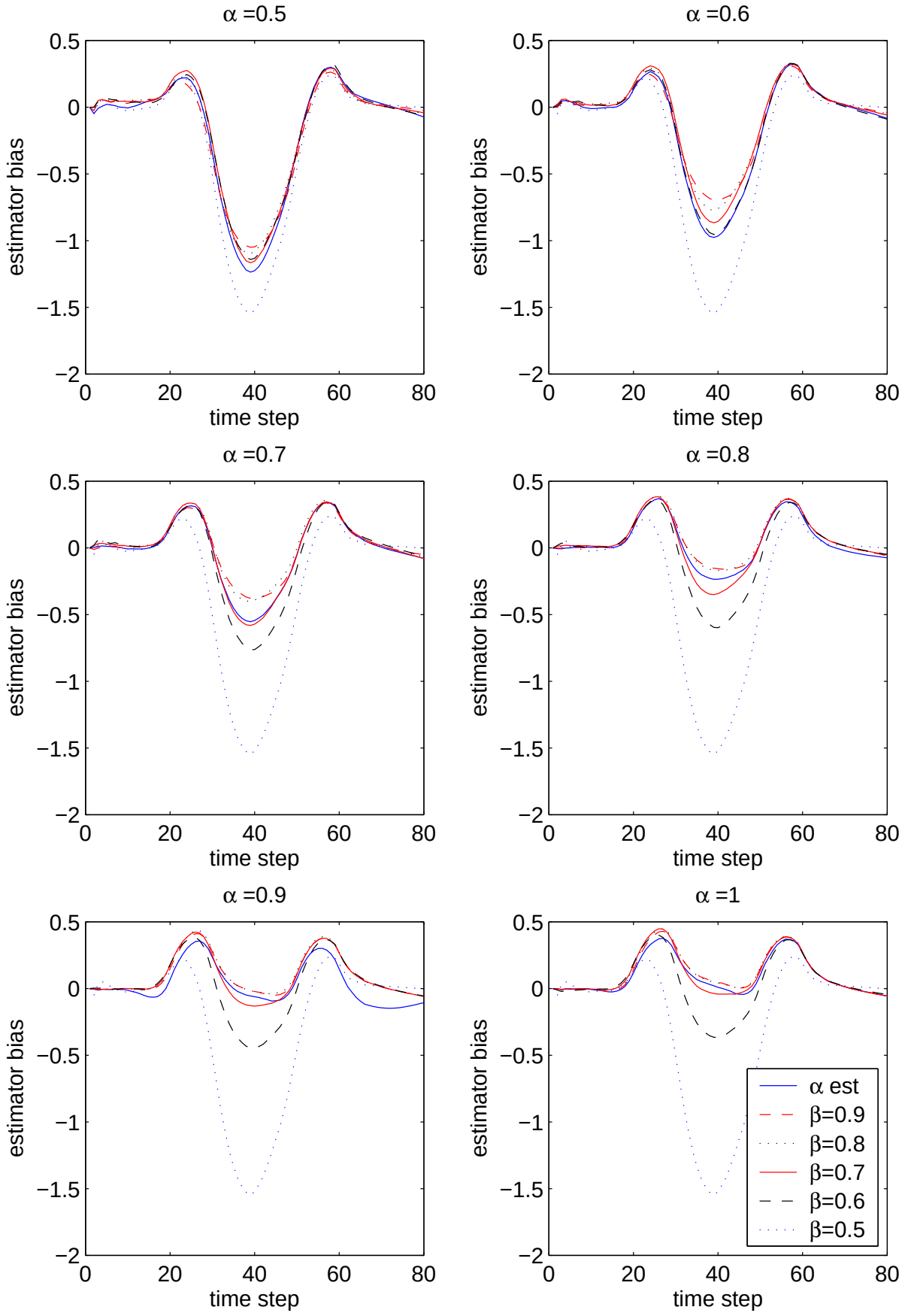


Figure 4.11: Bias of the estimator $\hat{r}_t^1$ with C mismatched, turning scenario

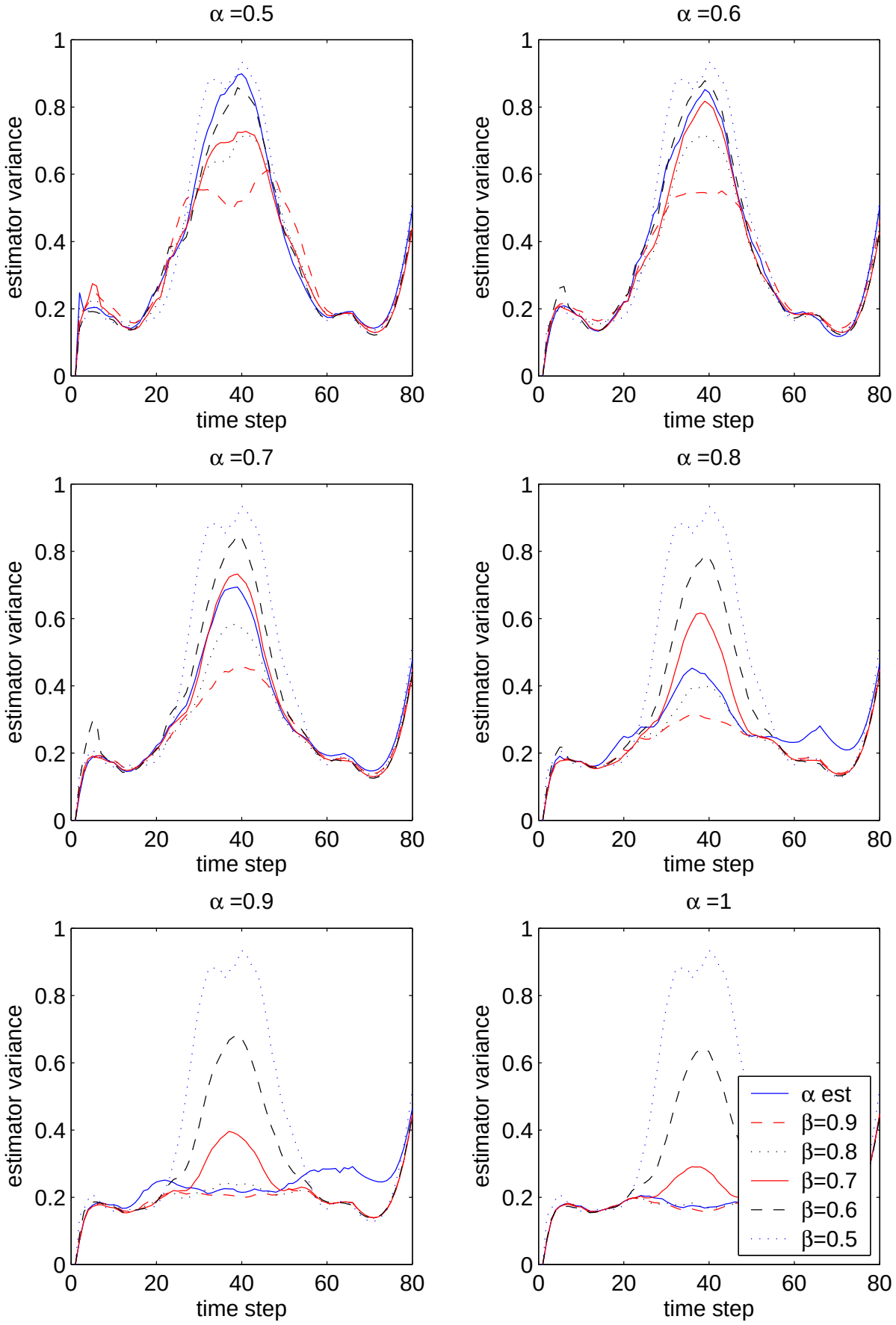


Figure 4.12: variance of the estimator $\hat{r}_t^1$ with C mismatched, turning scenario

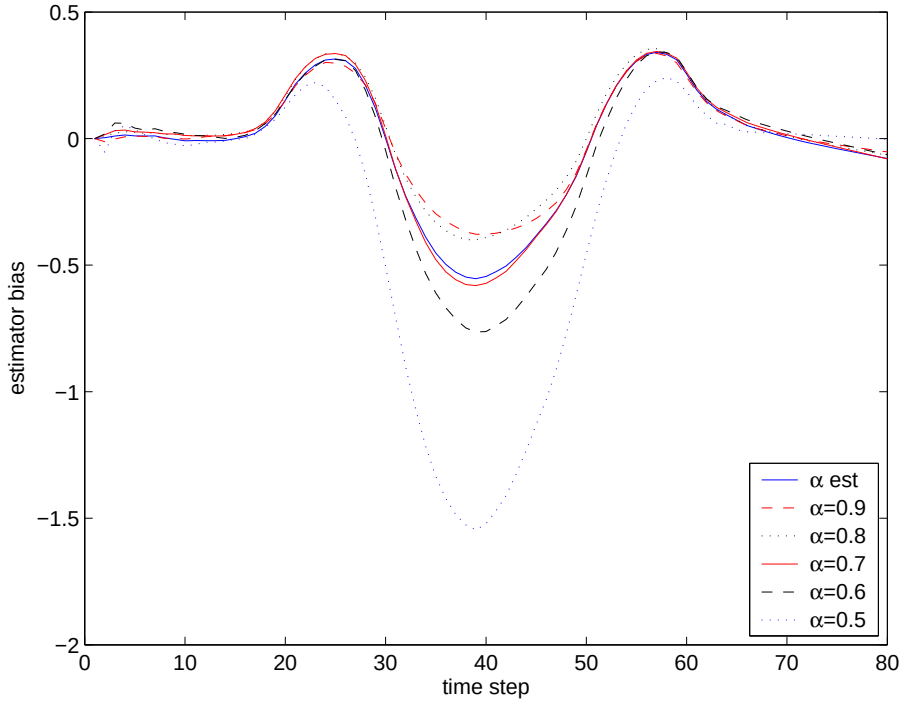


Figure 4.13: Bias of the estimator $\hat{r}_t^1$ when $\beta = 0.7$ is assumed, turning scenario

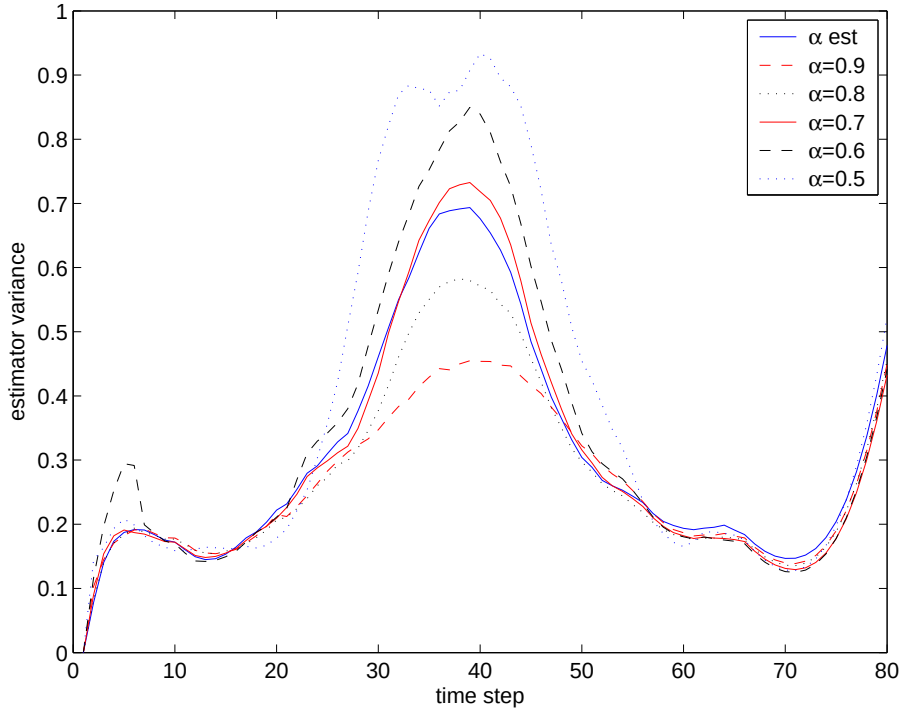


Figure 4.14: Variance of the estimator $\hat{r}_t^1$ when $\beta = 0.7$ is assumed, turning scenario

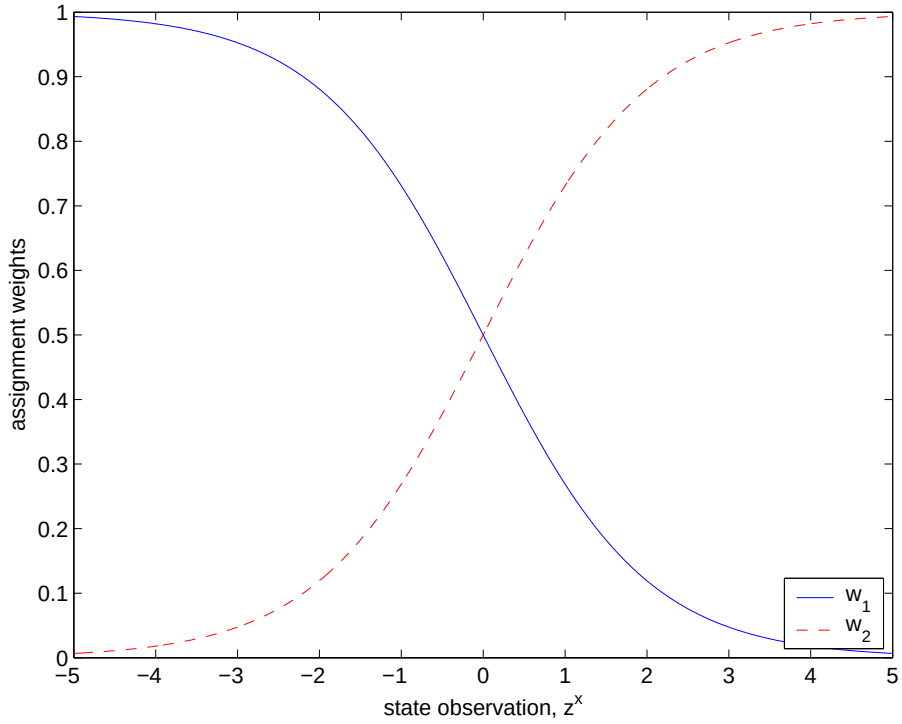


Figure 4.15: Assignment weights for PMHT

nonlinear distance measure, and the weights for the tracks will follow the curves shown in figure 4.15. The weight for track 1, w_1, is plotted as a solid line and the weight for track 2, w_2, as a dashed line. Track 1 is preferred over track 2 when the measurement is closer to it and visa versa.

Suppose there is now a classification measurement associated with the state observation, and that the PMHT-c assumes that the probability that it is correct is β. There are two pairs of curves, one for the case that the classification indicates target 1, i.e. $\boldsymbol{z}^k = 1$, and one for the case that it indicates target 2, i.e. $\boldsymbol{z}^k = 2$. Since the scenario chosen is symmetrical, these curves are simply reflections of each other. The weight functions obtained are plotted in figure 4.16 for $\beta = 0.7$ and figure 4.17 for $\beta = 0.9$. By assuming a higher value of β, the crossover point of the w_1 and w_2 curves is moved away from the geometric midpoint of the two tracks. The effect of the classification measurement is to shift the point where both tracks are equally likely to have caused the measurement away from the track indicated by the classification measurement.

The reason that the PMHT-c gives improved performance over the PMHT is that these weight functions now depend on the true source of the measurement. The weights shown in figure 4.15 depend only on the value of the state observation - there is no way of knowing which target caused the observation, so the weight curves are independent of the true measurement source, k. This is not the case for the curves shown in figures 4.16 and 4.17 because they also depend on the classification measurement $\boldsymbol{z}^k$ which is dependent on the true source of the measurement, k. So, with the PMHT-c, there will be different weight characteristics depending on the probability that each target will cause the class measurement. If $k = 1$ and the true confusion matrix parameter is α then the weight will have an expected value given by

$$\bar{w}_1 = \alpha w_1\left(\boldsymbol{z}^k = 1\right) + (1-\alpha) w_1\left(\boldsymbol{z}^k = 2\right). \tag{4.64}$$

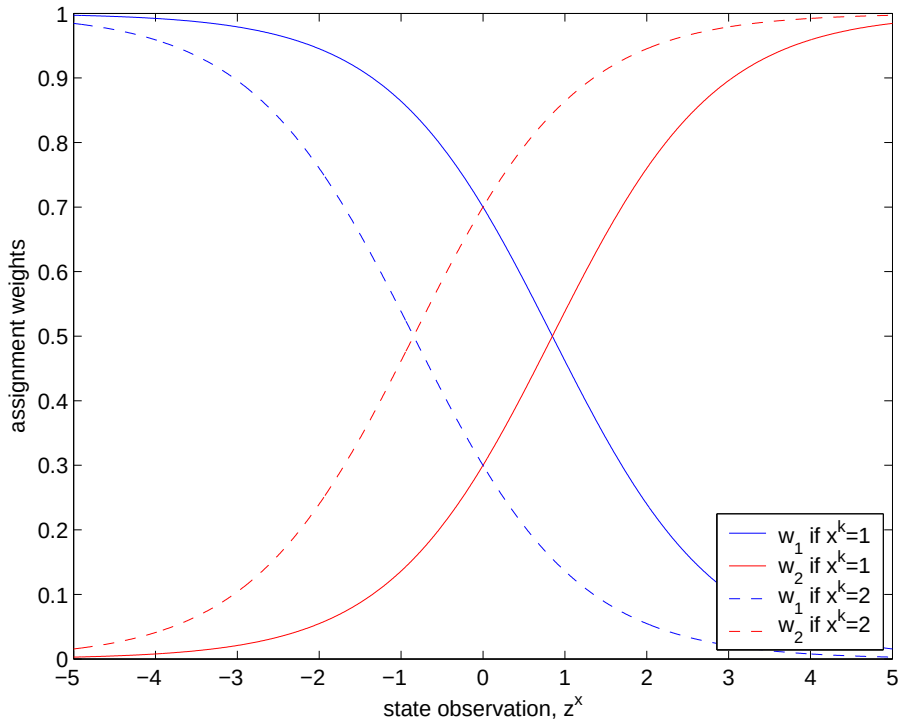


Figure 4.16: Assignment weights for PMHT-c when $\beta = 0.7$

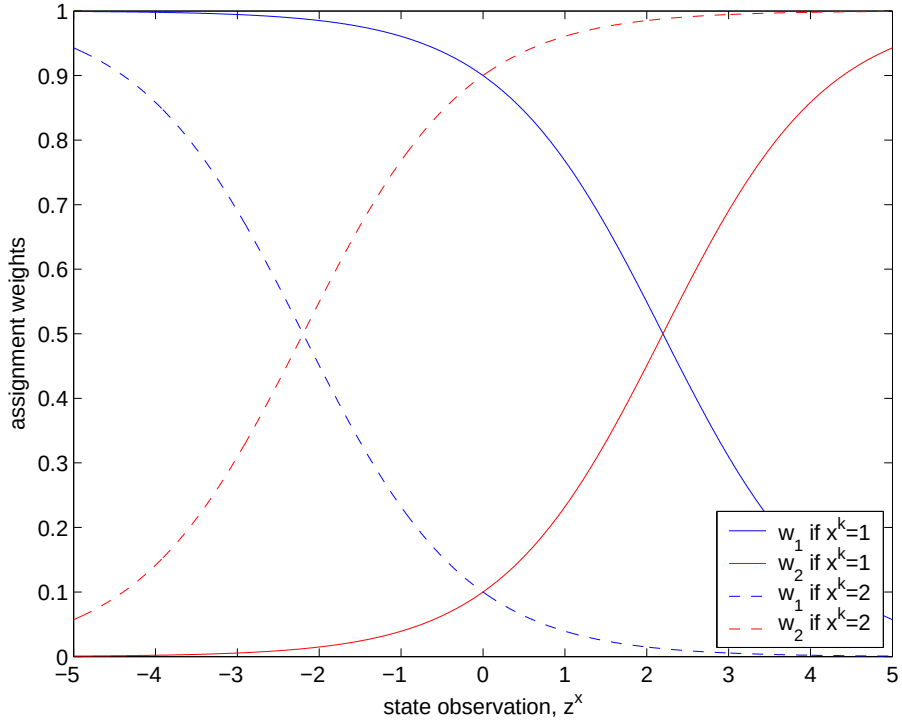


Figure 4.17: Assignment weights for PMHT-c when $\beta = 0.9$

This expected weight is dependent on k. Figure 4.18 plots the expected weights for track 1 when $\alpha = 0.7$ and $\alpha = 0.9$ and when the PMHT-c assumes $\beta = 0.7$ and $\beta = 0.9$.

When the classification measurements are more accurate ($\alpha = 0.9$), the weight curves extend farther away from the track before dropping down. This indicates that more measurements will be given the correct weight. Recall that these curves are produced for the case that $k = 1$. In fact, perfect classifications would give a curve that had $\bar{w}_1 = 1$ for all possible state observations. When the PMHT-c assumes that the measurements are more accurate than they really are, the expected weight is reduced close to the track since some measurements will be incorrectly classified and penalised by the PMHT-c. However, the expected weight is increased farther away. In effect, the weight given to nearby measurements is reduced slightly to increase the weight given to more distant measurements.

One way to quantify this *reaching* phenomenon is to plot the difference between the expected weights for track 1 when $k = 1$ and $k = 2$. The bigger this difference, the more the classification measurements are providing the filter with insight into the true k. Figure 4.20 shows this difference for $\alpha = 0.7$ and $\alpha = 0.9$ where the PMHT-c assumes $\beta = 0.7$ and $\beta = 0.9$. From figure 4.20 it is clear that there is a significant increase in the discrimination of the weights when a higher value of β is assumed.

The peak of the curves in figure 4.20 occurs at the spatial midpoint of the two tracks where

$$\begin{aligned}
\bar{w}_1(k=1) - \bar{w}_1(k=2) &= \Big\{ P\left(\boldsymbol{z}^k = 1 | k=1\right) w_1\left(\boldsymbol{z}^k = 1\right) + \\
&\qquad P\left(\boldsymbol{z}^k = 2 | k=1\right) w_1\left(\boldsymbol{z}^k = 2\right) \Big\} - \\
&\quad \Big\{ P\left(\boldsymbol{z}^k = 1 | k=2\right) w_1\left(\boldsymbol{z}^k = 1\right) + \\
&\qquad P\left(\boldsymbol{z}^k = 2 | k=2\right) w_1\left(\boldsymbol{z}^k = 2\right) \Big\} \\
&= \{\alpha\beta + (1-\alpha)(1-\beta)\} - \{(1-\alpha)\beta + \alpha(1-\beta)\} \\
&= (2\alpha - 1)(2\beta - 1) - 1. \qquad (4.65)
\end{aligned}$$

It can be seen from (4.65) that choosing a higher value of β will improve the discrimination. This might suggest that it is best to use β as high as possible, however, the probability of the target producing a measurement becomes very low as the distance from the target becomes large. Choosing β so that the weights remain high for these measurements is dangerous since it will mean that measurements due to other targets that are misclassified will be given high weights and the track will be lost. This is seen in the performance obtained when $\beta = 1$.

4.6 Value of Estimating the Confusion Matrix

Estimating the confusion matrix appears to give worse performance than assuming an incorrect confusion matrix. On any particular realisation, the adaptive algorithm will choose a particular estimate $\hat{\alpha}$ from the distribution of the estimator. If this value is high, the results show that the filter will give good performance. However, sometimes it will be 0.5 or lower. Even if the true associations were known, some realisations of the classification measurements would give estimates with low values. In this case, we have the performance of the first row of tables 4.3 and 4.4 or worse. In effect, the performance of the adaptive algorithm is the convolution of the estimator distribution with the column

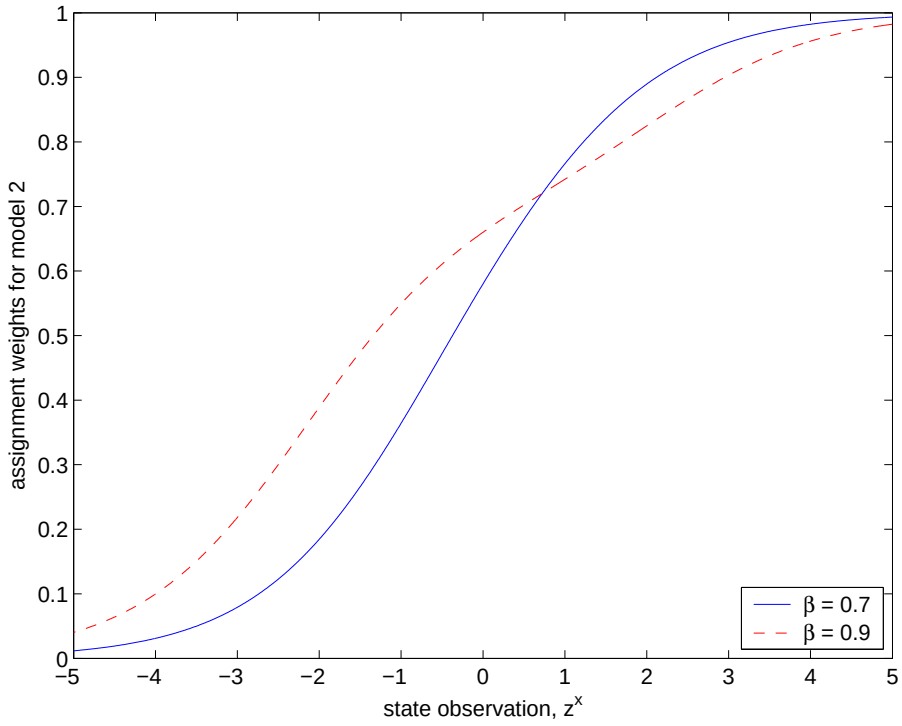


Figure 4.18: Assignment weights for PMHT-c when $\alpha = 0.7$

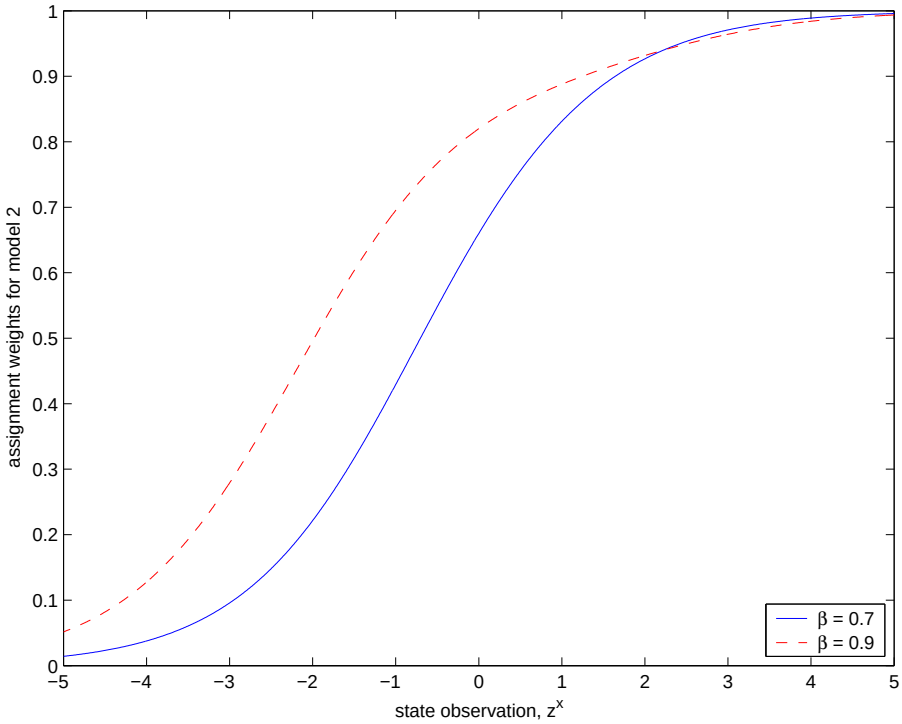


Figure 4.19: Assignment weights for PMHT-c when $\alpha = 0.9$

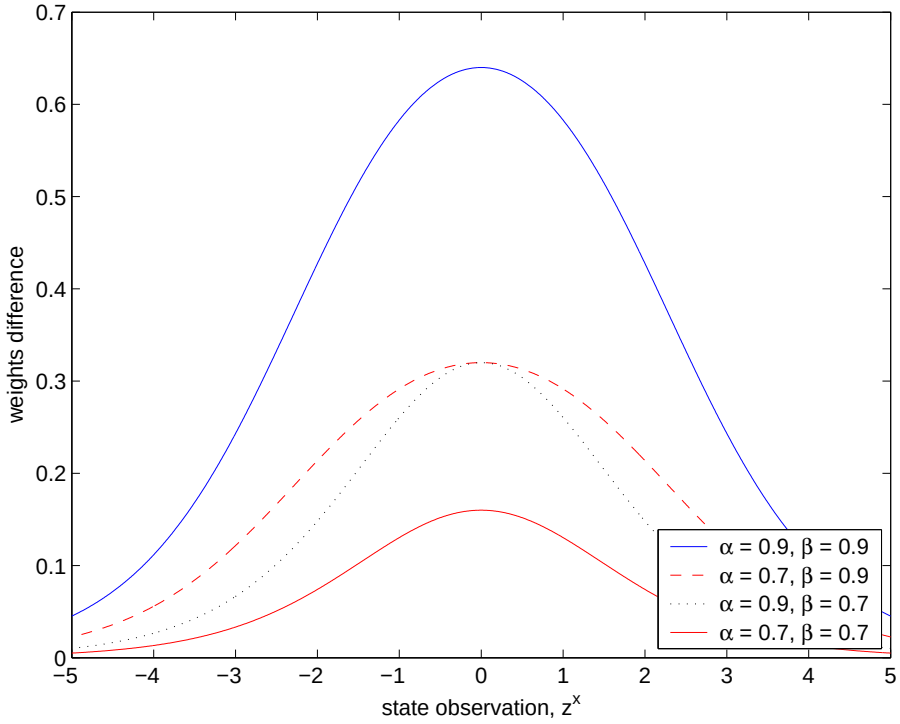


Figure 4.20: Assignment weights for PMHT

of table 4.4 or 4.3 corresponding to the true scenario. The result is that the overall performance is reduced by an amount dependent on the probability of the estimate, $\hat{\alpha}$, being 0.5 or lower.

For the scenario presented, there is no value to be gained by estimating the confusion matrix, unless learning it is the goal itself. It is unclear whether different scenarios might give more favourable results for this part of the algorithm.

4.7 Summary

In many applications the tracking algorithm has access to additional data besides the state observations which are usually assumed in the algorithm formulation. This data may be discrete and take the form of classifications, provided by either alternative processing of the sensor data, or by additional sensors. This chapter derived a PMHT algorithm for incorporating classification information by treating the data as observations of the assignment variable itself. This algorithm is referred to as the PMHT with classification measurements, or PMHT-c.

The PMHT-c was derived under both known (or assumed) classifier statistics, and unknown classifier statistics. When the classifier confusion matrix is unknown, the PMHT-c can estimate it. The PMHT-c was shown to simplify to the standard PMHT when the classification measurements were known to be uninformative, and to revert to hard association when the classification measurements are known to be perfect.

Simulations of two simple scenarios were used to demonstrate that the use of classification information can improve estimation accuracy, and also enhance the ability of the tracker to follow difficult manoeuvring trajectories. It was found that the difference in performance between a known and unknown confusion matrix was small.

Similar simulations were also used to examine the change in performance when the PMHT-c assumed a confusion matrix which was incorrect. Surprisingly, it was shown that the performance actually improved in many cases when an overly optimistic confusion matrix was assumed. This occurs because the PMHT-c with an optimistic confusion matrix gives more emphasis to the classification measurements. Provided that they are correct most of the time, this implies that the algorithm emphasises the correct assignment decision, and performance is improved.

Chapter 5

A Dynamic Model for Measurement Assignment - the Probabilistic Multi-Hypothesis Tracker with Hysteresis

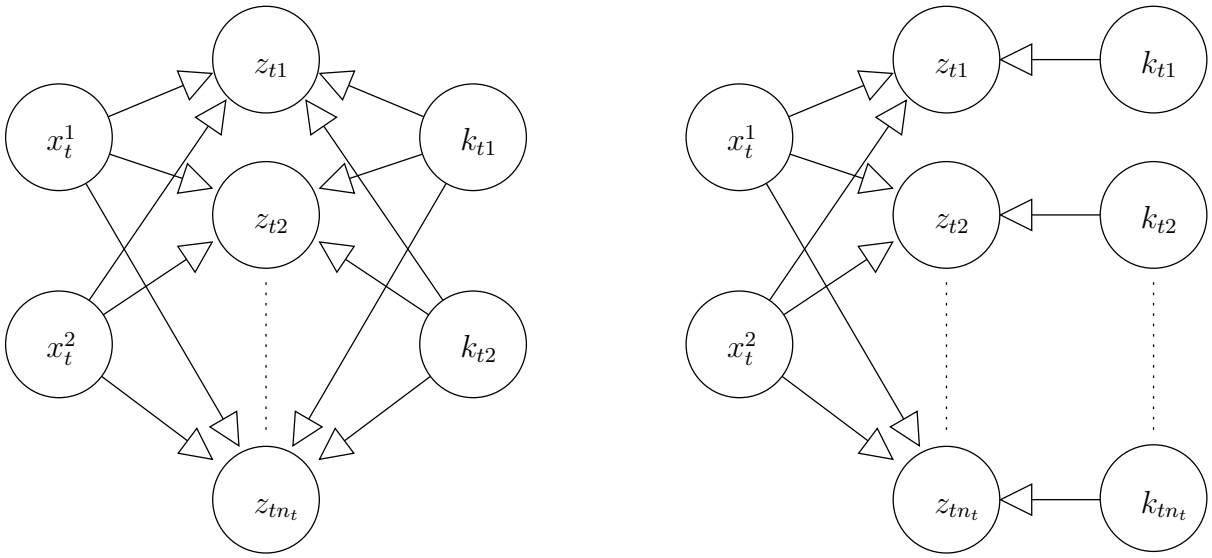
A key feature of the PMHT algorithm is the unique way in which it models the assignment of measurements to the dynamic state models used to represent targets and clutter. The PMHT assumes that the true source of each measurement is a random variable. Each measurement has an assignment random variable which is independent of the assignment of other measurements, given that the probability mass is known. When the assignment probability mass is unknown, it can be estimated by the PMHT.

Under the standard PMHT formulation [SL95], the probability mass of the assignment variable is assumed to be either time independent, or fixed over the batch. This is rather limiting, and it may be useful to adopt a model where the assignment probability evolves according to a probabilistic process.

This chapter presents a new extension of the PMHT which incorporates a state model for the assignment prior and hence allows for its dynamic evolution. That is, a Bayesian hyperparameter for the assignment process is introduced, and this hyperparameter is treated as a random process. This model is shown to be a generalisation of the standard PMHT assignment model. Two alternative approaches are identified for solving the generalised assignment model, and algorithms derived for each approach.

5.1 The Standard PMHT Measurement Assignment Model

The main difference between the PMHT algorithm and other tracking approaches is the assumed measurement model from which the algorithm arises. Under the usual measurement model, there is an assumed prior processing stage that ensures that every target present produces, at most, one measurement. If the target is distributed over several sensor bins, a quasi sufficient statistic is assumed to be achieved through locating the peak of this distributed response, or alternatively its centroid, or some other summarising point. The result of this assumption is that there is, at most, one measurement belonging to each track. This makes the track to observation association process dependent because the assignment of one observation may alter the possible assignment options for the next.



(a) standard measurement model (b) PMHT measurement model

Figure 5.1: Measurement Model BINs

The PMHT uses a different model, inspired by viewing the measurements as a collection of independent observations of a mixture. The PMHT assumes that the true assignment of each measurements is an independent random variable with a possible unknown prior probability mass function. The result of this assumption is that the track to observation association is independent for each different observation. This independence is the cause of the PMHT's linear complexity in the problem size, and allows the PMHT to perform batch processing, which is otherwise unfeasible due to the prohibitive growth in computational requirements under the standard model.

The difference between the assumed measurement process for the standard tracking paradigm and the PMHT is highlighted by the Bayesian Inference Networks (BINs) shown in figure 5.1. Each random variable is represented by a circle in the BIN and the arrows linking the circles show the conditional dependence of the variables. In the figure, there are two target models, and n_t measurements. Under the standard tracking measurement model, there is an assignment variable for each target model, and all measurements are dependent on it. For the PMHT model, there is different assignment for each measurement. Under the standard model, all measurements must be used to jointly estimate each assignment. Under the PMHT model, each measurement is used independently to estimate the single assignment variable associated with it.

As introduced in section 3.2.1, the prior probability mass of the assignment taking a value m, $P(k_{tr} = m)$, is denoted by π_t^m. The π_t^m values are constrained to sum to unity over m (by the law of total probability). Under the standard PMHT, the π_t^m are either known, or obey restrictive assumptions: they must be constant over fixed intervals and independent between intervals. These assumptions are usually simplified to make the priors constant over the batch, or independent from scan to scan. These restrictive assumptions will now be relaxed, and the probability mass will be assumed instead to obey a state model, with the state evolving randomly with time, i.e. the π_t^m are assumed to be a random process. The resulting algorithm is referred to as the PMHT with hysteresis, since the state model for assignments produces a temporal dependence in the assignment prior probability.

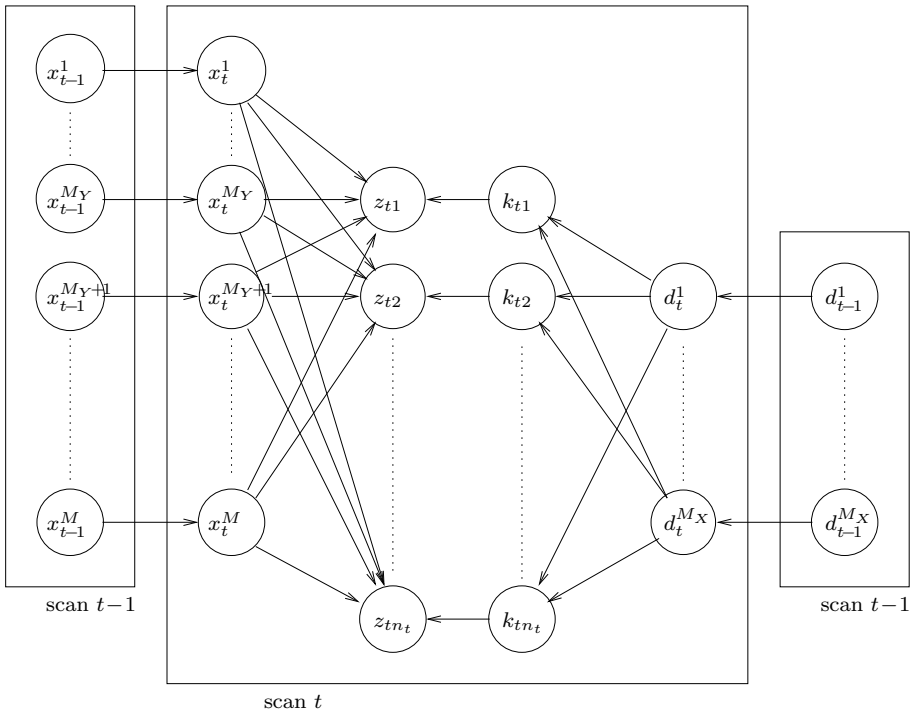


Figure 5.2: BIN for PMHT with hysteresis

5.2 Assignment State Model

The remainder of this chapter will deal with a problem definition expanded from that in chapter 3. The standard PMHT observer was reviewed in chapter 3, and consists of the model states, and the assignments. Here the observer is extended to also contain a further random variable, that acts as a state variable for the assignments. It will be assumed that a subset of the models have time dependent assignment probabilities. The others are independent, conditioned on the dependent priors (since normalisation links all of the priors). Those models that have time dependent assignment probability each have an associated assignment state. The assignment state for model m at scan t is d_t^m, and it defines the assignment prior probability for that model, i.e. d_t^m is a hyperparameter. d_t^m is a scalar. It will be seen to be convenient to choose the assignment state to be discrete (in order to achieve a problem solution). However, this is not a fundamental property of the hysteresis model and this choice will be deferred until the algorithm solution is obtained.

Define the per scan set $\boldsymbol{D}_t \equiv \left\{d_t^1 \dots d_t^{M_X}\right\}$ and the batch set $\boldsymbol{D} \equiv \{\boldsymbol{D}_0 \dots \boldsymbol{D}_T\}$. The assignment state is assumed to be a first order Markov random process with known evolution probability density (or mass) function $\Delta_t^m\left(d_t^m | d_{t-1}^m\right)$. The prior distribution for the assignment state of each model is also assumed known and is denoted as $\Delta_0^m\left(d_0^m\right)$.

Figure 5.2 shows a one scan slice of the Bayesian Inference Network (BIN) for the PMHT with hysteresis. The BIN demonstrates how the assignment state variables provide a dynamic model for the probability mass of K. This dynamic model is the key contribution of this chapter.

Assume that the number of measurements due to targets is relatively low compared with the number of measurements due to clutter. This implies that the mixing proportions associated with target models, π_t^m for $m = M_Y + 1 \dots M$, are low compared with those for the clutter models, π_t^m for $m = 1 \dots M_Y$. While this assumption is valid for many

tracking systems, it may not necessarily be the case.

The assignment state is the underlying random process that determines the π_t^m. It will be assumed that the dependence of $\mathbf{\Pi}_t$ upon $\boldsymbol{D}_t$ takes the form:

$$\pi_t^m\left(\boldsymbol{D}_t\right) = \begin{cases} \phi_t^{m-M_Y}\left(d_t^{m-M_Y}\right) & m > M_Y, \\ \sigma_t^m\left[1 - \sum_{s=1}^{M_X} \phi_t^s\left(d_t^s\right)\right] & 0 < m \leq M_Y, \end{cases} \tag{5.1}$$

where the functions $\phi_t^s\left(d_t^s\right)$ are assumed to be known, and the parameters σ_t^m are the relative mixing proportions of the clutter models, which are assumed to be unknown. The σ_t^m represent the prior probability that a measurement is due to clutter model m given that it is due to clutter, and are constrained by

$$0 \leq \sigma_t^m \leq 1, \tag{5.2}$$

$$\sum_{m=1}^{M_Y} \sigma_t^m = 1. \tag{5.3}$$

Define the sets $\mathbf{\Sigma}_t \equiv \left\{\sigma_t^1 \ldots \sigma_t^{M_Y}\right\}$ and $\mathbf{\Sigma} \equiv \left\{\mathbf{\Sigma}_1 \ldots \mathbf{\Sigma}_T\right\}$.

The functions $\phi_t^s\left(d_t^s\right)$ represent probabilities and so must be positive semi-definite functions, namely they may not give negative values for any d_t^s. (5.1) ensures that the π_t^m sum to unity for any values of $\phi_t^s\left(d_t^s\right)$, provided that the constraint (5.3) holds. However, if some of the $\phi_t^s\left(d_t^s\right)$ take values that are too large, then some of the π_t^m will not correspond to legal probabilities. For example, if the sum of the $\phi_t^s\left(d_t^s\right)$ is greater than unity, the clutter priors will be negative. This is why it is necessary to assume that the number of measurements due to targets is low. This assumption then corresponds to constraining the maximum values of each of the $\phi_t^s\left(d_t^s\right)$ to be sufficiently low to ensure that the π_t^m are positive.

It is possible to use a different definition for the dependence of the priors on the $\boldsymbol{D}$, rather than (5.1). A general form, such as assuming $\pi_t^m = \phi_t^m\left(\boldsymbol{D}_t\right)$ could be used. The form in (5.1) was chosen because it concentrates the joint dependence of $\mathbf{\Pi}$ into only the clutter models. The target π_t^m values are thus independent of each other. This may be desirable if partial EM is used to reduce the complexity. This choice does not alter the algorithm derivation, and a modified $\boldsymbol{D}$ dependence could easily be used.

The assignment state parameter will be assumed to occupy a finite discrete space. Without loss of generality, this space is assumed to contain M_D elements, with the assignment state taking an integer value between 0 and $M_D - 1$. The decision to use a discrete assignment state parameter, rather than a continuous one, is driven by a desire to achieve at a soluble problem. For continuous random variables, the only optimal estimators known are those due to Kalman [Kal60] and Benes [Ben81]. Each of these addresses a special case of the state evolution and observation equations. Both require a linear observation with Gaussian noise. In this problem, the functions $\Delta_t^m\left(d_t^m | d_{t-1}^m\right)$ and $\phi_t^m\left(d_t^m\right)$ are the evolution and observation functions respectively. To achieve the positive definite constraints, and an upper bound, means that these functions cannot adhere to the Kalman or Benes formulations. Thus, no continuous functions will give rise to an optimal solution (barring the discovery of another continuous optimal filter, which would be a major contribution of itself).

The particular forms of the functions $\phi_t^m\left(d_t^m\right)$ and $\Delta_t^m\left(d_t^m | d_{t-1}^m\right)$ are chosen according to the specific problem to be solved. The PMHT algorithms derived in this chapter are

applicable for any functions, when d_t^m is discrete. This is because discrete problems are solved in a direct numerical manner by algorithms such as the Viterbi algorithm, or the Hidden Markov Model Smoother, without the problematic analytic integration required for optimal solution of continuous problems.

5.2.1 A Note on Terminology

The hysteresis model is somewhat similar to a model used originally with the PDAF for track initiation. For historical reasons, this model is variously referred to as *target observability*, *track existence*, *null track*, *target perceivability*, and *target visibility* (e.g. [CDA86, MES94, LL01a, Col99]). These various names all apply to the same approach (with minor variations on the theme) and are collectively described as visibility. The purpose of visibility is to provide an integrated model specifically designed for making track decisions, such as those for automated initiation and termination of tracks.

In the PDAF context, visibility is a binary attribute of targets that controls whether or not they are able to be detected. A *visible* target provides a signal at the sensor which may (or may not) be detected, whereas an *invisible* target cannot be detected. An example of an invisible target, is one occluded from the sensor due to physical constraints, such as blind doppler zones, or the physical extent of the sensor region. Targets may become invisible due to a manoeuvre (for example an aircraft might land) or simply by leaving the footprint of the sensor. Tracks started on clutter measurements may be considered to be due to postulated fictitious invisible targets. Hence the model incorporates both detectability and existence.

The visibility model is not restricted to the PDAF, but is rather an alternative state model for targets. This model has been used in various other filters, such as the Gaussian Sum Filter [RA00], the Viterbi Algorithm [PL97], and with Random Sets [CVW02].

The framework of the PMHT with hysteresis can be viewed as a generalised version of visibility. The simplest case of the hysteresis model is when the assignment state variable is chosen to be binary. In this case, the PMHT with hysteresis simplifies to the PMHT with visibility when the appropriate dependence of $\mathbf{\Pi}$ on $\boldsymbol{D}$ is chosen. This model will be discussed in more detail in chapter 6 when the PMHT with hysteresis is applied to track initiation.

Whereas visibility is a binary random variable with a fixed effect on the detectability of a target, the assignment state model used for the PMHT is a general Markov process with an unspecified dependence on the prior assignment probability mass, $\mathbf{\Pi}$. The assignment state acts as a state of a general dynamics model for the assignment prior. Of course, particular models must be assumed to arrive at an implementable algorithm, but the PMHT formulation is not restricted to these forms.

In contrast to PDAF implementations of target visibility, the interpretation of the hysteresis model with PMHT is well defined and not contentious. In the PDAF community, the physical interpretation of visibility has been a source of some consternation among researchers, with various claims made about which interpretation offers the most complete theoretic description.

The term *visibility* is not used to describe the PMHT with Hysteresis because the extended assignment model is much more general than the problem of track initiation and termination addressed by the visibility model. The PMHT with Hysteresis is a fundamental generalisation of the standard PMHT, of which automated track decision making is merely an application.

5.2.2 PMHT with Hysteresis

The PMHT with hysteresis is derived in substantially the same way as the standard PMHT. The derivation of the PMHT has been given in more detail in chapter 4 where the PMHT-c algorithm was derived. Therefore, the derivation of the PMHT with hysteresis is merely outlined here, although detail is given where it deviates from the derivation in chapter 4. The symbols used in this derivation were defined in chapter 3.

There are two approaches that can be used to develop an algorithm based on this model. Firstly, the assignment state can be treated as further missing data in an EM context. This means that the auxiliary function will be the expectation over the assignments, $\mathbf{K}$, and the assignment states, $\boldsymbol{D}$. In this case, the algorithm will be seen to calculate the probabilities of the d_t^m, much as the standard PMHT calculates the probabilities of the assignments, k_{tr} (these are the weights, w_{mtr}).

The second method of solution is to estimate the assignment states. In this case, the auxiliary function becomes a function of the target states and the assignment states, dependent on their values from a previous iteration. Under this approach, the assignment state is treated in a similar way to the target state estimates. An initial assignment state sequence is assumed, and this sequence is iteratively refined based on the measurement assignment weights.

Each of the above approaches will now be explored, and PMHT algorithms derived. The algorithm resulting from treating the assignment states as missing data will be referred to by the notational shorthand PMHT-ym. Similarly, the algorithm resulting from estimation of the assignment states will be referred to as PMHT-ye.

5.3 Hysteresis as Missing Data, PMHT-ym

One way to derive an algorithm with the expanded measurement model incorporating hysteresis is to treat the assignment state variables as additional missing data. This approach is now considered. The algorithm derived through this approach is referred to as the PMHT-ym.

The missing data is now the assignments, and the assignment states. This means that the auxiliary function will be the conditional expectation over the assignment states, and the assignments. Namely,

$$Q\left(\mathbf{X}, \boldsymbol{\Sigma} | \mathbf{X}^{(i)}, \boldsymbol{\Sigma}^{(i)}\right) = \sum_{\boldsymbol{D}} \sum_{\mathbf{K}} \log P(\mathbf{X}, \boldsymbol{D}, \mathbf{K}, \mathbf{Z}) P(\boldsymbol{D}, \mathbf{K} | \mathbf{X}^{(i)}, \mathbf{Z}), \tag{5.4}$$

where the summation denoted $\sum_{\boldsymbol{D}}$ is the sum over all possible permutations of the assignment states, $\boldsymbol{D}$. Explicitly,

$$\sum_{\boldsymbol{D}} \{\cdot\} \equiv \sum_{d_0^1=0}^{M_D-1} \sum_{d_0^2=0}^{M_D-1} \cdots \sum_{d_0^{M_X}=0}^{M_D-1} \sum_{d_1^1=0}^{M_D-1} \cdots \sum_{d_t^{M_X}=0}^{M_D-1} \{\cdot\}. \tag{5.5}$$

The BIN in figure 5.2 illustrates the independence assumptions of the filter. These assumptions are that the state models are independent of each other and that the assignment states are independent of each other. Under these assumptions, the complete data likelihood is

$$P(\mathbf{X}, \boldsymbol{D}, \mathbf{K}, \mathbf{Z}) = P(\mathbf{X}) P(\boldsymbol{D}) P(\mathbf{K} | \boldsymbol{D}) P(\mathbf{Z} | \mathbf{K}, \mathbf{X}). \tag{5.6}$$

The assignments, k_{tr}, are conditionally independent given the assignment state vector for scan t, $\boldsymbol{D}_t$. Hence

$$P(\mathbf{K}|\boldsymbol{D}) = \prod_{t=1}^{T}\prod_{r=1}^{n_t} \pi_t^{k_{tr}}\left(\boldsymbol{D}_t\right), \tag{5.7}$$

where $\pi_t^{k_{tr}}\left(\boldsymbol{D}_t\right)$ is defined by (5.1).

The assignment states are independent of each other and are each first order Markov processes, so

$$P(\boldsymbol{D}) = \prod_{m=1}^{M_X}\left\{\Delta_0^m\left(d_0^m\right)\prod_{t=1}^{T}\Delta_t^m\left(d_t^m|d_{t-1}^m\right)\right\}. \tag{5.8}$$

The remaining terms in (5.6) are the same as the standard PMHT and are given by (4.11) and (4.14), namely

$$P(\mathbf{Z}|\mathbf{K},\mathbf{X}) = \prod_{t=1}^{T}\prod_{r=1}^{n_t} P\left(\boldsymbol{z}_{tr}|k_{tr},\mathbf{X}_t\right),$$

and

$$P(\mathbf{X}) = \prod_{m=1}^{M}\left\{\psi_0^m\left(\boldsymbol{x}_0^m\right)\prod_{t=1}^{T}\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)\right\}.$$

The conditional probability of the assignment states and the assignments in (5.4) can be expanded via Bayes rule as

$$P(\boldsymbol{D},\mathbf{K}|\mathbf{X}^{(i)},\mathbf{Z}) = P(\mathbf{K}|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z})P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}). \tag{5.9}$$

It is important to note that (5.9) is implicitly dependent on the clutter parameters $\boldsymbol{\Sigma}^{(i)}$ from the previous EM iteration. It is not a function of the unknown parameters $\boldsymbol{\Sigma}$.

Substituting (5.6) and (5.9) into (5.4) gives

$$\begin{aligned} Q\left(\mathbf{X},\boldsymbol{\Sigma}|\mathbf{X}^{(i)},\boldsymbol{\Sigma}^{(i)}\right) &= \log P(\mathbf{X}) + \sum_{\boldsymbol{D}}\log P(\boldsymbol{D})P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}) \\ &\quad + \sum_{\boldsymbol{D}}\sum_{\mathbf{K}}\log P(\mathbf{K}|\boldsymbol{D})P(\mathbf{K}|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z})P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}) \\ &\quad + \sum_{\boldsymbol{D}}\sum_{\mathbf{K}}\log P(\mathbf{Z}|\mathbf{X},\mathbf{K})P(\mathbf{K}|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z})P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}). \end{aligned} \tag{5.10}$$

The first term in (5.10) is the log likelihood of the state sequence. This term is the same as for the standard PMHT and is fully expanded in (4.14). The second term is constant and can be ignored. The third and fourth terms in (5.10) both involve a double sum over $\boldsymbol{D}$ and $\mathbf{K}$. Note that the dependence upon $\boldsymbol{\Sigma}$ is implicit in the probability masses of the assignments.

The $\log P(\mathbf{K}|\boldsymbol{D})$ term is the only part of (5.10) that is dependent on the clutter parameter, $\boldsymbol{\Sigma}$. Substituting (5.1) and (5.7), and making similar manipulations as in section 4.1.4, the $\log P(\mathbf{K}|\boldsymbol{D})$ term in (5.10) can be written as

$$
\begin{aligned}
&\sum_{\boldsymbol{D}}\sum_{\mathbf{K}} \log P(\mathbf{K}|\boldsymbol{D}) P(\mathbf{K}|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}) \\
&\quad = \sum_{\boldsymbol{D}}\sum_{\mathbf{K}} \left\{ \sum_{t=1}^{T}\sum_{r=1}^{n_t} \log P\left(k_{tr}|\boldsymbol{D}_t\right) \right\} P(\mathbf{K}|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}) \\
&\quad = \sum_{t=1}^{T} \left\{ \sum_{\boldsymbol{D}_t}\sum_{\mathbf{K}_t}\sum_{r=1}^{n_t} \log \pi_t^{k_{tr}}\left(\boldsymbol{D}_t\right) P(\mathbf{K}_t|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}_t|\mathbf{X}^{(i)},\mathbf{Z}) \right\} \\
&\quad = \sum_{t=1}^{T} \left\{ \sum_{\boldsymbol{D}_t}\sum_{r=1}^{n_t}\sum_{k_{tr}=1}^{M} \log \pi_t^{k_{tr}}\left(\boldsymbol{D}_t\right) P(\mathbf{K}_t|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}_t|\mathbf{X}^{(i)},\mathbf{Z}) \right\} \\
&\quad = \sum_{t=1}^{T}\sum_{\boldsymbol{D}_t}\sum_{m=1}^{M} \log \pi_t^{m}\left(\boldsymbol{D}_t\right) \sum_{r=1}^{n_t} P(k_{tr}=m|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}_t|\mathbf{X}^{(i)},\mathbf{Z}) \\
&\quad = \sum_{t=1}^{T}\sum_{\boldsymbol{D}_t} \left\{ \sum_{m=1}^{M_X} \log \phi_t^{m}\left(d_t^{m}\right) \sum_{r=1}^{n_t} w_{(m+M_Y)tr}(\boldsymbol{D}_t) + \right. \\
&\qquad \left. \sum_{s=1}^{M_Y} \left[\log \sigma_t^{s} + \log\left(1 - \sum_{m=1}^{M_X} \phi_t^{m}\left(d_t^{m}\right) \right) \right] \sum_{r=1}^{n_t} w_{str}(\boldsymbol{D}_t) \right\} P(\boldsymbol{D}_t|\mathbf{X}^{(i)},\mathbf{Z}), \quad (5.11)
\end{aligned}
$$

where $w_{mtr}(\boldsymbol{D}_t)$ is defined as

$$
w_{mtr}(\boldsymbol{D}_t) \equiv P(k_{tr}=m|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}). \quad (5.12)
$$

Note that $w_{mtr}(\boldsymbol{D}_t)$ is implicitly dependent on the iteration index, i, but this index is suppressed to somewhat simplify notation.

The terms involving $\phi_t^{m}\left(d_t^{m}\right)$ in (5.11) are constant, since they are summed over d_t^{m}. So, the only term in (5.11) that is significant is the one involving σ_t^{s}. Let

$$
Q_{t\pi} \equiv \sum_{s=1}^{M_Y} \left\{ \log \sigma_t^{s} \sum_{r=1}^{n_t} w_{str} \right\}, \quad (5.13)
$$

where

$$
\begin{aligned}
w_{mtr} &= \sum_{\boldsymbol{D}} w_{mtr}(\boldsymbol{D}_t) P(\boldsymbol{D}_t|\mathbf{X}^{(i)},\mathbf{Z}) \\
&= \sum_{\boldsymbol{D}} P(k_{tr}=m|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}_t|\mathbf{X}^{(i)},\mathbf{Z}). \quad (5.14)
\end{aligned}
$$

Then, (5.11) can be written as

$$
\sum_{\boldsymbol{D}}\sum_{\mathbf{K}} \log P(\mathbf{K}|\boldsymbol{D}) P(\mathbf{K}|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}) = \sum_{t=1}^{T} Q_{t\pi} + A_K, \quad (5.15)
$$

where A_K is an irrelevant constant.

The measurement term, $\log P(\mathbf{Z}|\mathbf{X},\mathbf{K})$, can be simplified using the same process as in section 4.1.4 and can be written as

$$
\begin{aligned}
&\sum_{\boldsymbol{D}}\sum_{\mathbf{K}} \log P(\mathbf{Z}|\mathbf{X},\mathbf{K}) P(\mathbf{K}|\boldsymbol{D},\mathbf{X}^{(i)},\mathbf{Z}) P(\boldsymbol{D}|\mathbf{X}^{(i)},\mathbf{Z}) \\
&\qquad = \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{r=1}^{n_t} \log \zeta_t^{m}\left(z_{tr}|x_t^{m}\right) w_{mtr}.
\end{aligned} \quad (5.16)
$$

The auxiliary function can now be written as

$$Q\left(\mathbf{X}, \boldsymbol{\Sigma} | \mathbf{X}^{(i)}, \boldsymbol{\Sigma}^{(i)}\right) = \sum_{m=1}^{M} Q_X^m + \sum_{t=1}^{T} Q_{t\pi} + Q_D, \tag{5.17}$$

where Q_D is constant, and the model term, Q_X^m, is given by

$$Q_X^m = \log \psi_0^m\left(x_0^m\right) + \sum_{t=1}^{T} \log \psi_t^m\left(x_t^m | x_{t-1}^m\right) + \sum_{r=1}^{n_t} \log \zeta_t^m\left(z_{tr} | x_t^m\right) w_{mtr}. \tag{5.18}$$

(5.18) is the same expression as is achieved for the standard PMHT except that the weights are obtained using (5.14). As is described in section 4.1.5, this equation can be optimised using a Kalman smoother when the processes are linear and the random elements are Gaussian.

The $Q_{t\pi}$ term in (5.17) is a function of the relative clutter probabilities, $\boldsymbol{\Sigma}$, and is essentially the same as the prior auxiliary function for the standard PMHT. This is to be maximised subject to the constraint (5.3), namely

$$\sum_{m=1}^{M_Y} \sigma_t^m = 1.$$

This is achieved by using the Lagrangian $L_{t\pi} = Q_{t\pi} + \lambda_t \left(1 - \sum_{k=1}^{M_Y} \sigma_t^k\right)$. Setting the derivative $\dfrac{dL_{t\pi}}{d\sigma_t^k} = 0$ gives

$$\sigma_t^k = \frac{1}{\lambda_t} \sum_{r=1}^{n_t} w_{tr}^k, \tag{5.19}$$

and reapplying the constraint gives the estimate for σ_t^k as

$$\sigma_t^k = \frac{\sum_{r=1}^{n_t} w_{ktr}}{\sum_{m=1}^{M_Y} \sum_{r=1}^{n_t} w_{mtr}}. \tag{5.20}$$

This is the same form as the solution for π_t^m in the standard PMHT except that the weights are implicitly dependent on the assignment state probabilities.

5.3.1 Assignment Weights

To complete the algorithm, an expression for the assignment weight, w_{mtr}, is required. The weight, as defined in (5.14), is comprised of two terms

$$w_{mtr} = \sum_{\boldsymbol{D}} P(k_{tr} = m | \boldsymbol{D}, \mathbf{X}^{(i)}, \mathbf{Z}) P(\boldsymbol{D}_t | \mathbf{X}^{(i)}, \mathbf{Z}).$$

The first term in (5.14) is the conditional probability of the assignments, and the second term is the posterior probability of the assignment states.

The conditional probability of the assignments is found via Bayes Rule

$$\begin{aligned} P\left(k_{tr}=m|\mathbf{X}^{(i)},\boldsymbol{D},\mathbf{Z}\right) &= \frac{P\left(k_{tr}=m,\mathbf{X}^{(i)},\boldsymbol{D},\mathbf{Z}\right)}{\sum_{s=1}^{M}P\left(k_{tr}=s,\mathbf{X}^{(i)},\boldsymbol{D},\mathbf{Z}\right)} \\ &= \frac{P\left(\mathbf{X}^{(i)},\boldsymbol{D},\mathbf{Z}\setminus\boldsymbol{z}_{tr}\right)\pi_t^m\left(\boldsymbol{D}_t\right)\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i)}\right)}{\sum_{s=1}^{M}P\left(\mathbf{X}^{(i)},\boldsymbol{D},\mathbf{Z}\setminus\boldsymbol{z}_{tr}\right)\pi_t^s\left(\boldsymbol{D}_t\right)\zeta_t^s\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{s(i)}\right)} \\ &= \frac{\pi_t^m\left(\boldsymbol{D}_t\right)\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i)}\right)}{\sum_{s=1}^{M}\pi_t^s\left(\boldsymbol{D}_t\right)\zeta_t^s\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{s(i)}\right)}, \end{aligned} \tag{5.21}$$

where $Z\setminus z_{tr}$ is the set of all measurements in the batch except the measurement z_{tr}. (5.21) is the same as the standard PMHT weight equation except that the prior distribution of the assignments, $\pi_t^m\left(\boldsymbol{D}_t\right)$ is dependent on the assignment state $\boldsymbol{D}_t$.

The posterior probability of the assignment state vector D_t given all of the batch measurements and the state estimates at the previous iteration, namely $P(\boldsymbol{D}_t|\mathbf{X}^{(i)},\mathbf{Z})$, can be determined by using the Hidden Markov Model (HMM) smoother [JR86].

Using Bayes Rule write

$$\begin{aligned} P\left(\boldsymbol{D}_t|\mathbf{X},\mathbf{Z}\right) &\propto P\left(\boldsymbol{D}_t,\mathbf{X},\mathbf{Z}\right) \\ &= P\left(\boldsymbol{D}_t,\mathcal{X}_1^t,\mathcal{X}_{t+1}^T,\mathcal{Z}_1^t,\mathcal{Z}_{t+1}^T\right) \\ &= P\left(\boldsymbol{D}_t,\mathcal{X}_1^t,\mathcal{Z}_1^t\right)P\left(\mathcal{X}_{t+1}^T,\mathcal{Z}_{t+1}^T|\boldsymbol{D}_t,\mathcal{X}_1^t,\mathcal{Z}_1^t\right) \\ &\equiv \alpha_t\left(\boldsymbol{D}_t\right)\beta_t\left(\boldsymbol{D}_t\right)\equiv\gamma_t\left(\boldsymbol{D}_t\right), \end{aligned} \tag{5.22}$$

where

$$\mathcal{X}_{t_1}^{t_2} \equiv \left\{X_{t_1}\ldots X_{t_2}\right\}, \tag{5.23}$$

$$\mathcal{Z}_{t_1}^{t_2} \equiv \left\{Z_{t_1}\ldots Z_{t_2}\right\}. \tag{5.24}$$

The required probabilities are the $\gamma_t(\boldsymbol{D}_t)$ and these shall be found be deriving recursive relations for $\alpha_t(\boldsymbol{D}_t)$ and $\beta_t(\boldsymbol{D}_t)$.

$$\begin{aligned} \alpha_t\left(\boldsymbol{D}_t\right) &= P\left(\boldsymbol{D}_t,\mathcal{X}_1^t,\mathcal{Z}_1^t\right) \\ &= \sum_{\boldsymbol{D}_{t-1}}P\left(\boldsymbol{D}_t,\boldsymbol{D}_{t-1},\mathcal{X}_1^t,\mathcal{Z}_1^t\right) \\ &= \sum_{\boldsymbol{D}_{t-1}}P\left(\boldsymbol{D}_t|\boldsymbol{D}_{t-1}\right)P\left(\mathbf{X}_t|\mathbf{X}_{t-1}\right)P\left(\mathbf{Z}_t|\boldsymbol{D}_t,\mathbf{X}_t\right)\alpha_{t-1}\left(\boldsymbol{D}_{t-1}\right) \\ &\propto \sum_{\boldsymbol{D}_{t-1}}P\left(\boldsymbol{D}_t|\boldsymbol{D}_{t-1}\right)P\left(\mathbf{Z}_t|\boldsymbol{D}_t,\mathbf{X}_t\right)\alpha_{t-1}\left(\boldsymbol{D}_{t-1}\right) \\ &= \sum_{\boldsymbol{D}_{t-1}}\left\{\prod_{m=1}^{M_X}\Delta_t^m\left(d_t^m|d_{t-1}^m\right)\right\}P\left(\mathbf{Z}_t|\boldsymbol{D}_t,\mathbf{X}_t\right)\alpha_{t-1}\left(\boldsymbol{D}_{t-1}\right), \end{aligned} \tag{5.25}$$

and $\alpha_t\left(\boldsymbol{D}_t\right)$ is normalised by dividing by $\sum_{\boldsymbol{D}_t}\alpha_t\left(\boldsymbol{D}_t\right)$.

$$\begin{aligned}
\beta_t(\boldsymbol{D}_t) &= P\left(\mathcal{X}_{t+1}^T, \mathcal{Z}_{t+1}^T | \boldsymbol{D}_t, \mathcal{X}_1^t, \mathcal{Z}_1^t\right) \\
&= \sum_{\boldsymbol{D}_{t+1}} P\left(\mathcal{X}_{t+1}^T, \mathcal{Z}_{t+1}^T, \boldsymbol{D}_{t+1} | \boldsymbol{D}_t, \mathcal{X}_1^t, \mathcal{Z}_1^t\right) \\
&= \sum_{\boldsymbol{D}_{t+1}} P\left(\boldsymbol{D}_{t+1}|\boldsymbol{D}_t\right) P\left(\mathbf{X}_{t+1}|\mathbf{X}_t\right) P\left(\mathbf{Z}_{t+1}|\boldsymbol{D}_{t+1}, \mathbf{X}_{t+1}\right) \beta_{t+1}\left(\boldsymbol{D}_{t+1}\right) \\
&\propto \sum_{\boldsymbol{D}_{t+1}} P\left(\boldsymbol{D}_{t+1}|\boldsymbol{D}_t\right) P\left(\mathbf{Z}_{t+1}|\boldsymbol{D}_{t+1}, \mathbf{X}_{t+1}\right) \beta_{t+1}\left(\boldsymbol{D}_{t+1}\right) \\
&= \sum_{\boldsymbol{D}_{t+1}} \left\{ \prod_{m=1}^{M_X} \Delta_{t+1}^m \left(d_{t+1}^m | d_t^m\right) \right\} P\left(\mathbf{Z}_{t+1}|\boldsymbol{D}_{t+1}, \mathbf{X}_{t+1}\right) \beta_{t+1}\left(\boldsymbol{D}_{t+1}\right), \qquad (5.26)
\end{aligned}$$

and $\beta_t(\boldsymbol{D}_t)$ is normalised by dividing by $\sum_{\boldsymbol{D}_t} \beta_t(\boldsymbol{D}_t)$.

The incomplete conditional data likelihood $P\left(\mathbf{Z}_t|\boldsymbol{D}_t, \mathbf{X}_t\right)$ can be written as

$$\begin{aligned}
P\left(\mathbf{Z}_t|\boldsymbol{D}_t, \mathbf{X}_t\right) &= \prod_{r=1}^{n_t} P(\boldsymbol{z}_{tr}|\boldsymbol{D}_t, \mathbf{X}_t) \\
&= \prod_{r=1}^{n_t} \left\{ \sum_{m=1}^{M} \pi_t^m\left(\boldsymbol{D}_t\right) \quad \zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}}\right) \right\}. \qquad (5.27)
\end{aligned}$$

5.3.2 Statement of PMHT-ym Algorithm

The PMHT-ym algorithm is mostly the same as the standard PMHT. Where it differs, is that the PMHT-ym has an additional step where the assignment state probabilities are calculated via the HMM smoother, and the assignment weights are dependent on these probabilities. The algorithm proceeds as follows:

1. Initialise the model state estimates $\mathbf{X}^{(0)}$.

2. Determine the posterior probability of assignment state using

$$P\left(\boldsymbol{D}_t|\mathbf{X}^{(i-1)}, \mathbf{Z}\right) = \frac{\alpha_t\left(\boldsymbol{D}_t\right)\beta_t\left(\boldsymbol{D}_t\right)}{\sum_{\boldsymbol{U}} \alpha_t\left(\boldsymbol{D}_t = \boldsymbol{U}\right)\beta_t\left(\boldsymbol{D}_t = \boldsymbol{U}\right)}.$$

   where $\alpha_t(\boldsymbol{D}_t)$ and $\beta_t(\boldsymbol{D}_t)$ are defined using (5.25) and (5.26). This probability is implicitly iteration dependent through the conditioning on the state $\mathbf{X}^{(i-1)}$.

3. Calculate the assignment weights for each measurement and model,

$$w_{mtr}^{(i)} = \sum_{\boldsymbol{D}_t} \frac{P\left(\boldsymbol{D}_t|\mathbf{X}^{(i-1)}, \mathbf{Z}\right) P\left(k_{tr} = m|\boldsymbol{D}_t\right) \zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i-1)}\right)}{\sum_{p=0}^{M} P\left(k_{tr} = p|\boldsymbol{D}_t\right) \zeta_t^p\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{p(i-1)}\right)}.$$

4. Update the state estimates using the Maximum Likelihood Estimator to give $\mathbf{X}^{(i)}$. This part of the algorithm is identical to the standard PMHT. As with the standard PMHT, a Kalman Smoother can be exploited to find the ML estimates under the case that the evolution pdf, $\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)$, and the observation pdf, $\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right)$, are linear Gaussian functions.

5. Repeat steps 2 ... 4 until convergence

5.4 Estimated Assignment State, PMHT-ye

The alternative to treating the assignment state variable as missing data is to estimate it. Under this approach, the assignment state plays a similar role in the algorithm to the model states: an initial assignment state sequence is assumed, and then the assignment weights are used to iteratively estimate the optimal assignment state sequence for each target track. The algorithm derived under this approach is referred to as the PMHT-ye.

5.4.1 Modified Auxiliary Function

The auxiliary function for the PMHT-ym is given by (5.4),

$$Q\left(\mathbf{X}, \boldsymbol{\Sigma} | \mathbf{X}^{(i)}, \boldsymbol{\Sigma}^{(i)}\right) = \sum_{\boldsymbol{D}} \sum_{\mathbf{K}} \log P(\mathbf{X}, \boldsymbol{D}, \mathbf{K}, \mathbf{Z}) P(\boldsymbol{D}, \mathbf{K} | \mathbf{X}^{(i)}, \mathbf{Z}).$$

Under the PMHT-ym approach, the assignment state sequence is missing data, and the auxiliary function is the expectation over the missing data. For this approach, the assignment state sequence is to be estimated, so the new auxiliary function is:

$$Q\left(\mathbf{X}, \boldsymbol{D}, \boldsymbol{\Sigma} | \mathbf{X}^{(i)}, \boldsymbol{D}^{(i)}, \boldsymbol{\Sigma}^{(i)}\right) = \sum_{\mathbf{K}} \log P(\mathbf{X}, \boldsymbol{D}, \mathbf{K}, \mathbf{Z}) P(\mathbf{K} | \mathbf{X}^{(i)}, \boldsymbol{D}^{(i)}, \mathbf{Z}). \tag{5.28}$$

The difference between (5.28) and (5.4) is the removal of the summation over the possible assignment state values and the modified term $P(\mathbf{K}|\mathbf{X}^{(i)}, \boldsymbol{D}^{(i)}, \mathbf{Z})$. This term has already been derived in section 5.3.1 and is given by (5.21)

$$\begin{aligned} P(\mathbf{K}|\mathbf{X}^{(i)}, \boldsymbol{D}^{(i)}, \mathbf{Z}) &= \prod_{t=1}^{n_t} \prod_{r=1}^{n_t} P\left(k_{tr} | \mathbf{X}^{(i)}, \boldsymbol{D}^{(i)}, \mathbf{Z}\right) \\ &= \prod_{t=1}^{n_t} \prod_{r=1}^{n_t} w_{mtr}\left(\boldsymbol{D}_t^{(i)}\right)\Big|_{m=k_{tr}} \\ &= \prod_{t=1}^{n_t} \prod_{r=1}^{n_t} \frac{\pi_t^m\left(\boldsymbol{D}_t^{(i)}\right) \zeta_t^m\left(\boldsymbol{z}_{tr} | \boldsymbol{x}_t^{m(i)}\right)\Big|_{m=k_{tr}}}{\sum_{s=1}^{M} \pi_t^s\left(\boldsymbol{D}_t^{(i)}\right) \zeta_t^s\left(\boldsymbol{z}_{tr} | \boldsymbol{x}_t^{s(i)}\right)}. \end{aligned}$$

Using the same method as in section 5.3, the auxiliary function can be written as

$$\begin{aligned} Q\left(\mathbf{X}, \boldsymbol{D}, \boldsymbol{\Sigma} | \mathbf{X}^{(i)}, \boldsymbol{D}^{(i)}, \boldsymbol{\Sigma}^{(i)}\right) &= \log P(\mathbf{X}) + \log P(\boldsymbol{D}) \\ &\quad + \sum_{\mathbf{K}} \log P(\mathbf{K}|\boldsymbol{D}) P(\mathbf{K}|\boldsymbol{D}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z}) \\ &\quad + \sum_{\mathbf{K}} \log P(\mathbf{Z}|\mathbf{X}, \mathbf{K}) P(\mathbf{K}|\boldsymbol{D}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z}). \end{aligned} \tag{5.29}$$

The term $\sum_{\mathbf{K}} \log P(\mathbf{K}|\boldsymbol{D}) P(\mathbf{K}|\boldsymbol{D}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z})$ is simplified using the same steps as in (5.11)

$$
\begin{aligned}
&\sum_{\mathbf{K}} \log P(\mathbf{K}|\boldsymbol{D}) P(\mathbf{K}|\boldsymbol{D}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z}) \\
&\quad = \sum_{\mathbf{K}} \left\{ \sum_{t=1}^{T} \sum_{r=1}^{n_t} \log P\left(k_{tr}|\boldsymbol{D}_t\right) \right\} P(\mathbf{K}|\boldsymbol{D}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z}) \\
&\quad = \sum_{t=1}^{T} \left\{ \sum_{\mathbf{K}_t} \sum_{r=1}^{n_t} \log P\left(k_{tr}|\boldsymbol{D}_t\right) P(\mathbf{K}_t|\boldsymbol{D}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z}) \right\} \\
&\quad = \sum_{t=1}^{T} \sum_{m=1}^{M} \log P\left(k_{tr} = m|\boldsymbol{D}_t\right) \sum_{r=1}^{n_t} P(k_{tr} = m|\boldsymbol{D}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z}) \\
&\quad = \sum_{t=1}^{T} \left\{ \sum_{m=1}^{M_X} \log \phi_t^m\left(d_t^m\right) \sum_{r=1}^{n_t} w_{(m+M_Y)tr}(\boldsymbol{D}_t^{(i)}) + \right. \\
&\qquad \left. \sum_{s=1}^{M_Y} \left[\log \sigma_t^s + \log \left(1 - \sum_{m=1}^{M_X} \phi_t^m\left(d_t^m\right) \right) \right] \sum_{r=1}^{n_t} w_{str}(\boldsymbol{D}_t^{(i)}) \right\} \\
&\quad = \sum_{t=1}^{T} Q_{t\pi} + \sum_{m=1}^{M_X} \sum_{t=1}^{T} \log \phi_t^m\left(d_t^m\right) \bar{w}_{mt} + \sum_{t=1}^{T} \log \left(1 - \sum_{m=1}^{M_X} \phi_t^m\left(d_t^m\right) \right) \bar{w}_{0t}, \qquad (5.30)
\end{aligned}
$$

where

$$
\bar{w}_{mt} \equiv \sum_{r=1}^{n_t} w_{(m+M_Y)tr}(\boldsymbol{D}_t^{(i)}), \qquad (5.31)
$$

$$
\bar{w}_{0t} \equiv \sum_{s=1}^{M_Y} \sum_{r=1}^{n_t} w_{str}(\boldsymbol{D}_t^{(i)}). \qquad (5.32)
$$

The clutter term, $Q_{t\pi}$, is identical to the term for the PMHT-ym, given in (5.13) except that the weights w_{mtr} are replaced by conditional weights $w_{mtr}\left(\boldsymbol{D}_t^{(i)}\right)$

$$
Q_{t\pi} \equiv \sum_{s=1}^{M_Y} \left\{ \log \sigma_t^s \sum_{r=1}^{n_t} w_{str}\left(\boldsymbol{D}_t^{(i)}\right) \right\}. \qquad (5.33)
$$

Substituting (5.30) into (5.29), the auxiliary function can be written as

$$
Q\left(\mathbf{X}, \boldsymbol{\Sigma}|\mathbf{X}^{(i)}, \boldsymbol{\Sigma}^{(i)}\right) = \sum_{m=1}^{M} Q_X^m + \sum_{t=1}^{T} Q_{t\pi} + Q_D, \qquad (5.34)
$$

where, unlike (5.17), the Q_D term is no longer constant and is the objective function for optimising the assignment state sequence estimate. This Q_D term is given by

$$
\begin{aligned}
Q_D &= \log P(\boldsymbol{D}) + \sum_{m=1}^{M_X} \sum_{t=1}^{T} \log \phi_t^m\left(d_t^m\right) \bar{w}_{mt} + \sum_{t=1}^{T} \log \left(1 - \sum_{m=1}^{M_X} \phi_t^m\left(d_t^m\right) \right) \bar{w}_{0t} \\
&= \sum_{m=1}^{M_X} \left\{ \log \Delta_0^m\left(d_0^m\right) + \sum_{t=1}^{T} \left[\log \Delta_t^m\left(d_t^m|d_{t-1}^m\right) + \log \phi_t^m\left(d_t^m\right) \bar{w}_{mt} \right] \right\} \\
&\qquad + \sum_{t=1}^{T} \log \left(1 - \sum_{m=1}^{M_X} \phi_t^m\left(d_t^m\right) \right) \bar{w}_{0t}. \qquad (5.35)
\end{aligned}
$$

The model term, Q_X^m, is identical to that for the PMHT-ym given in (5.18) except that the weights w_{mtr} are replaced by conditional weights $w_{mtr}\left(\boldsymbol{D}_t^{(i)}\right)$

$$Q_X^m = \log \psi_0^m\left(x_0^m\right) + \sum_{t=1}^{T} \log \psi_t^m\left(x_t^m | x_{t-1}^m\right) + \sum_{r=1}^{n_t} \log \zeta_t^m\left(z_{tr} | x_t^m\right) w_{mtr}\left(\boldsymbol{D}_t^{(i)}\right). \tag{5.36}$$

Like (5.18), (5.36) is the same expression as is achieved for the standard PMHT except with a different definition for the weights. As is described in section 4.1.5, this equation can be optimised using a Kalman smoother when the processes are linear and the random elements are Gaussian.

The solution for the σ_t^k estimate is found by optimising $Q_{t\pi}$. Since the σ_t^k are constrained to sum to unity, this is done via a Lagrangian, as shown in section 5.3. The estimator for σ_t^k is then

$$\sigma_t^k = \frac{\sum_{r=1}^{n_t} w_{ktr}\left(\boldsymbol{D}_t^{(i)}\right)}{\sum_{m=1}^{M_Y}\sum_{r=1}^{n_t} w_{mtr}\left(\boldsymbol{D}_t^{(i)}\right)}. \tag{5.37}$$

5.4.2 Assignment State Sequence Estimate

All that remains to complete the algorithm is to derive a recursion for the estimated assignment state sequence by optimising (5.35)

$$\begin{aligned} Q_D \;=\; & \sum_{m=1}^{M_X}\left\{\log \Delta_0^m\left(d_0^m\right) + \sum_{t=1}^{T}\left[\log \Delta_t^m\left(d_t^m | d_{t-1}^m\right) + \log \phi_t^m\left(d_t^m\right)\bar{w}_{mt}\right]\right\} \\ & + \sum_{t=1}^{T} \log\left(1 - \sum_{m=1}^{M_X} \phi_t^m\left(d_t^m\right)\right)\bar{w}_{0t}. \end{aligned}$$

The difficulty is that the final term couples together the assignment states of all the target models. The result is that the optimal estimator for the assignment state sequence will have to be a joint estimator for all the target models. This should not be particularly surprising, since the PMHT-ym algorithm resulted in a joint probability calculation.

The optimal sequence for $\boldsymbol{D}$ can be found using the Viterbi algorithm [Vit67, FJ73]. Since (5.35) is coupled between the target models via the clutter term, the assignment states must be jointly estimated. This means that the Viterbi algorithm must run over a state space with cardinality $(M_D)^{M_X}$. For completeness, a statement of the Viterbi algorithm applied to this problem is now presented. This algorithm is repeated for each target model (or the models could be estimated in parallel).

5.4.2.1 Viterbi Algorithm for Assignment State Sequence Estimation

Define nodes, n_t^j, corresponding to the vth possible assignment state value at scan t. The nodes n_0^v represent the possible initial assignment state values. The number of nodes for each t (i.e. the range of values for j) is the cardinality of the joint assignment state domain, $(M_D)^{M_X}$. There are $(M_D)^{M_X}$ nodes at each scan, and T scans, giving a total of $(M_D)^{M_X T}$ possible assignment state sequences. Enumerating these is possible through brute force, however it is unnecessary because the aim to find the best path, not rank all

paths. Thus, some sequences can be discarded before they are fully enumerated, once it is apparent that they cannot be the best path.

The log likelihood of a particular sequence can be accumulated along the path due to the Markov nature of the assignment state process - given d_t^m, d_{t+1}^m is independent of all other d_u^s and its likelihood is known. Further, for a particular node, it is possible to construct the likelihood of all partial sequences after that node, independent of the partial sequence prior to the node. This means that if the best sequence passes through node n_t^j, then it must consist of the best partial sequence leading from scan 1 to n_t^j and the best partial sequence from n_t^j to scan T.

The Viterbi algorithm then operates by retaining only the best partial sequence leading to each node for each scan. When the last scan is considered, then the best sequence is chosen from the best sequences leading to each final node. Effectively, each possible path is considered, but only $(M_D)^{M_X}$ sequences need to be remembered instead of $(M_D)^{M_X T}$.

For each node, define a cost, q_t^j, and a path p_t^j. The path for node n_t^j represents the most likely assignment state at $t-1$ given that the assignment state at t is n_t^j. The cost is the log likelihood of the most likely assignment state sequence leading to node n_t^j.

The Viterbi algorithm proceeds as follows:

1. For each j, calculate the initial node costs given by

$$q_0^j = \sum_{m=1}^{M_X} \log \Delta_0^m \left(d_0^m \equiv n_0^j \right). \tag{5.38}$$

2. Starting with $t = 1$, and for each j determine the best path p_t^j given by

$$p_t^j = \arg\max \sum_{m=1}^{M_X} \left[q_{t-1}^k + \log \Delta_t^m \left(d_t^m \equiv n_t^j | d_{t-1}^m \equiv n_{t-1}^k \right) \right]. \tag{5.39}$$

3. Calculate the node cost for each node at t given by

$$\begin{aligned} q_t^j &= \sum_{m=1}^{M_X} \left[q_{t-1}^k + \log \Delta_t^m \left(d_t^m \equiv n_t^j | d_{t-1}^m \equiv n_{t-1}^k \right) \right] \Big|_{k=p_t^j} \\ &\quad + \sum_{m=1}^{M_X} \bar{w}_{mt} \log \phi_t^m \left(d_t^m \equiv n_t^j \right) + \bar{w}_{0t} \log \left(1 - \sum_{m=1}^{M_X} \phi_t^m \left(d_t^m \equiv n_t^j \right) \right). \end{aligned} \tag{5.40}$$

4. Repeat steps 2 and 3 for $t = 2 \ldots T$.

5. Determine the most likely final assignment state by finding the node with the highest q_T^j

$$\hat{\boldsymbol{D}}_T \equiv \arg\max \left[q_T^j \right]. \tag{5.41}$$

6. Determine the earlier assignment state values by following the path backwards from $t = T$ to $t = 1$

$$\hat{\boldsymbol{D}}_{t-1} = p_t^j \Big|_{k \equiv \hat{\boldsymbol{D}}_t^m}. \tag{5.42}$$

5.4.3 Statement of PMHT-ye Algorithm

The PMHT-ye algorithm is mostly the same as the standard PMHT. Where it differs, is that the PMHT-ye has an additional step where the assignment state sequence is determined via the Viterbi algorithm, and the assignment weights are dependent on this sequence.

The algorithm proceeds as follows:

1. Initialise the model state estimates, $\mathbf{X}^{(0)}$, and the assignment state estimates, $\boldsymbol{D}^{(0)}$.
2. Calculate the assignment weights for each measurement and model,

$$w_{mtr}^{(i)} = \frac{P\left(k_{tr} = m|\boldsymbol{D}_t^{(i-1)}\right)\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i-1)}\right)}{\sum_{p=1}^{M} P\left(k_{tr} = p|\boldsymbol{D}_t^{(i-1)}\right)\zeta_t^p\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{p(i-1)}\right)}.$$

3. Update the state estimates using the Maximum Likelihood Estimator to give $\mathbf{X}^{(i)}$. This part of the algorithm is identical to the standard PMHT. As with the standard PMHT, a Kalman Smoother can be exploited to find the ML estimates under the case that the evolution pdf, $\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)$, and the observation pdf, $\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right)$, are linear Gaussian functions.
4. Using the Viterbi algorithm, update the assignment state estimates to give $\boldsymbol{D}^{(i)}$.
5. Repeat steps 2 ... 4 until convergence

5.4.4 Approximate PMHT-ye with Reduced Complexity

In the derivation of the PMHT-ye, it is fairly clear that the necessity to jointly estimate the assignment state variables arises from the coupling through the clutter term in (5.35). If this term can be decoupled, then it would be possible to independently optimise the different models. This is appealing because the Viterbi algorithm must consider all possible transitions in the state space, of which there are D^2, where D is the number of states. Performing M_X parallel optimisations over a M_D sized space is far preferable to performing one optimisation over a $(M_D)^{M_X}$ sized space.

It is not possible to achieve this decoupling. However, an approximate decoupling can be achieved by using the Taylor series expansion of the logarithm

$$\log(1+z) = z - \frac{1}{2}z^2 + \frac{1}{3}z^3 - \ldots \qquad |z| \leq 1 \text{ and } z \neq -1, \tag{5.43}$$

which for small z, admits the approximation

$$\log(1+z) \approx z \qquad |z| \ll 1. \tag{5.44}$$

Notice that it has already been assumed in section 5.2.2 that the $\phi_t^m\left(d_t^m\right)$ are sufficiently small to preserve normalisation. This requirement is quite loose. Provided that the other values are very small, $\phi_t^m\left(d_t^m\right) = 0.9$ would still be acceptable. If this requirement is tightened such that

$$\sum_{m=1}^{M_X} \phi_t^m\left(d_t^m\right) \ll 1, \tag{5.45}$$

then (5.35) can be approximated by

$$\begin{aligned} Q_D &\approx \sum_{m=1}^{M_X} \left\{ \log \Delta_0^m \left(d_0^m\right) + \sum_{t=1}^{T} \left[\log \Delta_t^m \left(d_t^m | d_{t-1}^m\right) + \log \phi_t^m \left(d_t^m\right) \bar{w}_{mt} \right] \right\} \\ &\quad + \sum_{t=1}^{T} \left(- \sum_{m=1}^{M_X} \phi_t^m \left(d_t^m\right) \right) \bar{w}_{0t} \qquad (5.46) \\ &= \sum_{m=1}^{M_X} \left\{ \log \Delta_0^m \left(d_0^m\right) + \sum_{t=1}^{T} \left[\log \Delta_t^m \left(d_t^m | d_{t-1}^m\right) + \log \phi_t^m \left(d_t^m\right) \bar{w}_{mt} - \phi_t^m \left(d_t^m\right) \bar{w}_{0t} \right] \right\} \\ &= \sum_{m=1}^{M_X} Q_D^m. \qquad (5.47) \end{aligned}$$

The advantage of this approximation is that each of the functions Q_D^m is now independent of all of the other models and the assignment state sequence for model m can be estimated independently of the other models. This means that the algorithm has linear complexity in the number of target models.

Let $\tilde{\phi}_t^m \left(d_t^m\right)$ be defined such that

$$\log \tilde{\phi}_t^m \left(d_t^m\right) \equiv \log \phi_t^m \left(d_t^m\right) \bar{w}_{mt} - \phi_t^m \left(d_t^m\right) \bar{w}_{0t}, \qquad (5.48)$$

$$\tilde{\phi}_t^m \left(d_t^m\right) \equiv \frac{\phi_t^m \left(d_t^m\right)^{\bar{w}_{mt}}}{\exp\left[\phi_t^m \left(d_t^m\right) \bar{w}_{0t}\right]}. \qquad (5.49)$$

Then, Q_D^m can be written as

$$Q_D^m = \log \Delta_0^m \left(d_0^m\right) + \sum_{t=1}^{T} \left[\log \Delta_t^m \left(d_t^m | d_{t-1}^m\right) + \log \tilde{\phi}_t^m \left(d_t^m\right) \right]. \qquad (5.50)$$

The function $\tilde{\phi}_t^m \left(d_t^m\right)$ is analogous to a measurement likelihood. If it were possible to choose $\phi_t^m \left(d_t^m\right)$ such that $\tilde{\phi}_t^m \left(d_t^m\right)$ is a Gaussian function, then the assignment state sequence could be estimated with a Kalman smoother. However, the form of (5.49) makes it impossible to find such a function for $\phi_t^m \left(d_t^m\right)$. In fact, it appears that there is no function $\phi_t^m \left(d_t^m\right)$ that will result in a closed form solution[1]. However, (5.50) can be optimised since d_t^m is a discrete random variable. As with the proper PMHT-ye, the assignment state sequence is now estimated with the Viterbi algorithm. Due to the separation achieved through the approximation, each model is now optimised with an independent Viterbi smoother.

5.4.5 Special Cases

In the previous chapter, the PMHT-c was seen to simplify to the standard PMHT when the classification measurements were known (or assumed) to be uninformative. Similarly, the PMHT-ye simplifies to the standard PMHT under special conditions. This kind of behaviour is not considered for the PMHT-ym, because it treats the prior in a different way to the standard PMHT, which estimates it.

The transition probability, $\Delta_t^m \left(d_t^m | d_{t-1}^m\right)$, determines the main difference between the standard PMHT and the PMHT-ym. The standard PMHT assumes the assignment

[1] It is the author's belief that no closed form solution exists. No attempt has been made to prove this, although it can be shown that there is no function $\phi_t^m \left(d_t^m\right)$ that results in a Gaussian $\tilde{\phi}_t^m \left(d_t^m\right)$.

prior is constant, or independent with time. These two cases can be simply emulated by choosing $\Delta_t^m\left(d_t^m|d_{t-1}^m\right) = \delta\left(d_t^m - d_{t-1}^m\right)$ (where $\delta(\cdot)$ is the Dirac delta function) or $\Delta_t^m\left(d_t^m|d_{t-1}^m\right) = \frac{1}{M_D}$ respectively. In the first case, the Viterbi algorithm always selects the node corresponding to the same assignment state at the previous time, since other transitions are illegal. This means that the M_D state sequences at the end of the batch all correspond to different possible constant states, and the ML constant value is chosen. In the second case, the most likely previous state for each node is simply the ML state at that time, since all transitions are equally likely. The final sequence estimate will be the set of ML states for each point in time, determined independently. However, for both cases, the estimated prior value is constrained to take one of M_D values, since d_t^m is a discrete variable. The standard PMHT does not impose these conditions on the prior. Thus the PMHT-ym only simplifies to the PMHT in the limit as $M_D \to \infty$, i.e. when d_t^m becomes continuous. Further, the $\phi_t^m\left(d_t^m\right)$ must be such that the prior π_t^m is capable of taking all values in $[0, 1]$, i.e. it must have support over $[0, 1]$. Under these conditions, the PMHT-ym simplifies to the standard PMHT.

5.5 Comparison of PMHT-ym and PMHT-ye

The PMHT-ym and PMHT-ye are both algorithms similar to the standard PMHT. This is to be expected since they address substantially the same problem. One salient area in which they differ in computational complexity. The standard PMHT requires computations that scale linearly with the problem size. This is not the case for the PMHT-ym and PMHT-ye. Since these algorithms both operate jointly over the target space for some part of their work, these parts of the algorithm scale exponentially with the number of targets; each must consider $(M_D)^{2M_X}$ transitions. This is because of the assumed measurement model. However, other parts of the algorithm scale linearly, notably the state estimation.

Figure 5.3 shows flow diagrams for the two algorithms. These flow diagrams highlight some of the differences between them. The key differences are:

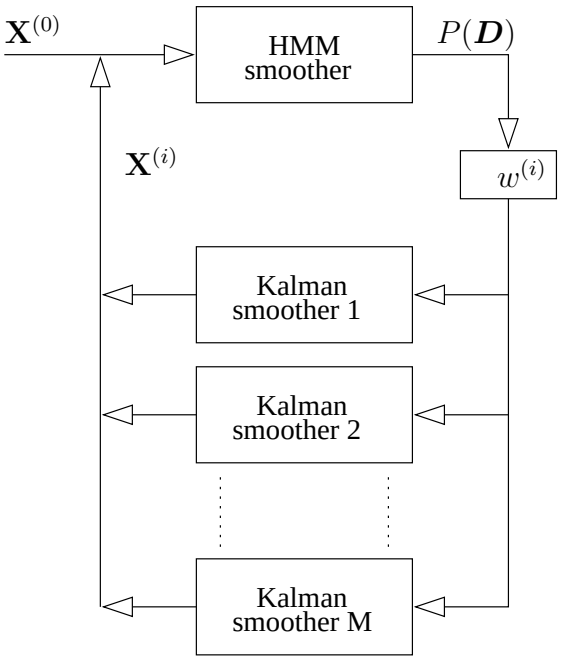
- The PMHT-ye depends on initialisation of the assignment state variables. This means that local convergence may be an issue.

- The PMHT-ye constrains the assignment state to take one of a fixed set of values. If fine discrimination in π_t^m is required, a large state space must be used (i.e. large M_D) and the complexity of the algorithm grows quickly with M_D. In contrast, the PMHT-ym effectively averages over the different discrete values and can be expected to achieve finer resolution for low M_D.

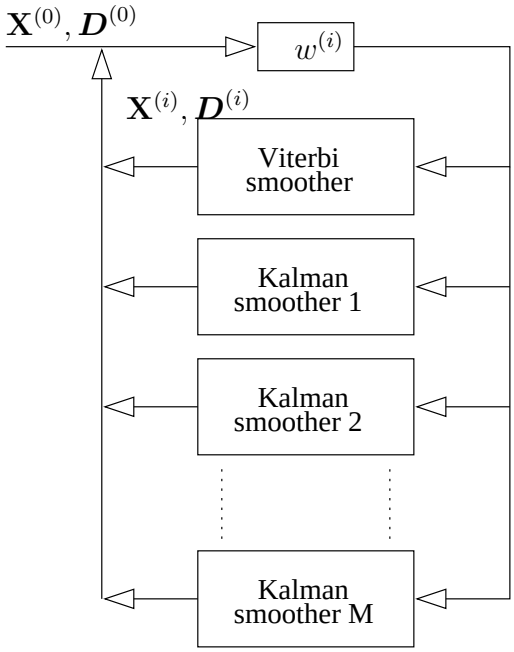
- The PMHT-ym provides the probability of the assignment state at each point in time, which gives a measure of the uncertainty in the estimated prior. The PMHT-ye provides only an estimated sequence of values.

- The PMHT-ym probability is a single scan quantity. This means that the PMHT-ym can estimate sequences that are illegal (where the corresponding $\Delta_t^m\left(d_t^m|d_{t-1}^m\right)$ is zero).

- The PMHT-ye estimates a sequence, so incorrect estimates in one scan produce a bias in the next, compounding errors.



(a) PMHT-ym



(b) PMHT-ye

Figure 5.3: Flow diagrams for PMHTs with hysteresis

5.6 Unknown Assignment State Parameters

The derivations of both the PMHT-ym and PMHT-ye assume that the parameters of the assignment state model are known. Namely, $\Delta_0^m\left(d_0^m\right)$, $\Delta_t^m\left(d_t^m|d_{t-1}^m\right)$, and $\phi_t^m\left(d_t^m\right)$ are all known. The validity of this assumption will depend on the application. If the assignment state model is adopted simply as a means of smoothing the estimated π_t^m then these parameters are design variables, chosen to ensure the desired behaviour of the estimated π_t^m. However, if it is believed that the true underlying prior truly follows a Markov chain, then it is not appropriate to choose somewhat arbitrary values. Under this circumstance it may be desirable to estimate the true assignment state parameters.

Firstly, consider the family of functions $\phi_t^m\left(d_t^m\right)$. This function is analogous to the measurement function in a standard HMM. However, it does not control the measurements directly, but influences the probability of selecting different models. Effectively, this function acts as a discretisation of the π_t^m function. Thus, the designer can always choose the $\phi_t^m\left(d_t^m\right)$.

In contrast, the $\Delta_t^m\left(d_t^m|d_{t-1}^m\right)$ represent the true underlying dynamic behaviour of the π_t^m. This function can be incorporated in the same way as the unknown confusion matrix in chapter 4. In a similar way, the optimal estimate for each matrix element is given by

$$\Delta_t^m\left(d_t^m = i|d_{t-1}^m = j\right) = \frac{\gamma_t\left(d_t^m = i, d_{t-1}^m = j\right)}{\gamma_{t-1}\left(d_{t-1}^m = j\right)}, \tag{5.51}$$

where the probability $\gamma_t\left(d_t^m = i, d_{t-1}^m = j\right)$ is determined using a Hidden Markov Model smoother with forwards and backwards recursions similar to those used for $\gamma_t\left(d_t^m\right)$ as given in (5.25) and (5.26). The above estimate for Δ suffers from a scarcity of data, and it would probably be prudent to assume that the transition function is stationary. This

leads to temporal averaging and the estimated Δ is then given by

$$\Delta_t^m\left(d_t^m = i | d_{t-1}^m = j\right) = \frac{\sum_{t=1}^{T} \gamma_t\left(d_t^m = i, d_{t-1}^m = j\right)}{\sum_{t=1}^{T} \gamma_{t-1}\left(d_{t-1}^m = j\right)}. \tag{5.52}$$

The main application of the assignment state model considered in this thesis is as a means for smoothing the π_t^m estimate. Thus, it will be assumed that the assignment state parameters are known, since they are design choices based on the degree of smoothing desired.

5.7 Simulated Example

To demonstrate the use of the PMHT-ym and PMHT-ye algorithms, a simple simulated example is now considered. The simulation consists of a two model scenario with static states, but varying mixing proportions. Multiple scalar measurements are detected at each scan, and these are the only information provided to the tracker.

Each model has a two element state, defining the mean and variance of the (Gaussian) measurement density function for that model. Thus the state vector is

$$\boldsymbol{x}_t^m \equiv \boldsymbol{x}^m = \left[x^m[\mu], x^m[\sigma^2]\right]^{\mathsf{T}}, \tag{5.53}$$

and the measurement pdf is

$$\zeta^m\left(z_{tr}\right) = \left(2\pi x^m[\sigma^2]\right)^{-\frac{1}{2}} \exp\left\{-\frac{\left(z_{tr} - x^m[\mu]\right)^2}{2x^m[\sigma^2]}\right\}. \tag{5.54}$$

Both models are chosen to have a variance of 2.25, i.e. $x^1[\sigma^2] = x^2[\sigma^2] = 2.25$. The means are chosen as $x^1[\mu] = 1$ and $x^2[\mu] = -1$. A batch length of $T = 100$ is used with $n_t = 5$ measurements per scan.

The mixing proportions of the two components vary sinusoidally such that

$$\pi_t^1 = 0.5 - 0.3cos\left(\frac{2\pi f t}{T}\right). \tag{5.55}$$

Thus the π_t^m vary between 0.2 and 0.8, and there are f cycles of the sinusoid over the batch. Two different frequencies are considered: $f = 1$, and $f = 4$. Figure 5.4 shows the measurements from an example trial where $f = 1$ is used. It can be seen that there are more measurements at high values when the first model dominates, in the middle of the batch.

The mean and variance of the state estimate and the prior estimate are measured over 1000 Monte Carlo trials. The PMHT-ym does not produce a prior estimate, but rather a probability mass for the prior. The mean of this probability mass is used as a prior estimate for the purpose of this comparison.

The performance of the PMHT-ym and PMHT-ye is compared with the standard PMHT which assumes that the prior probabilities are time independent.

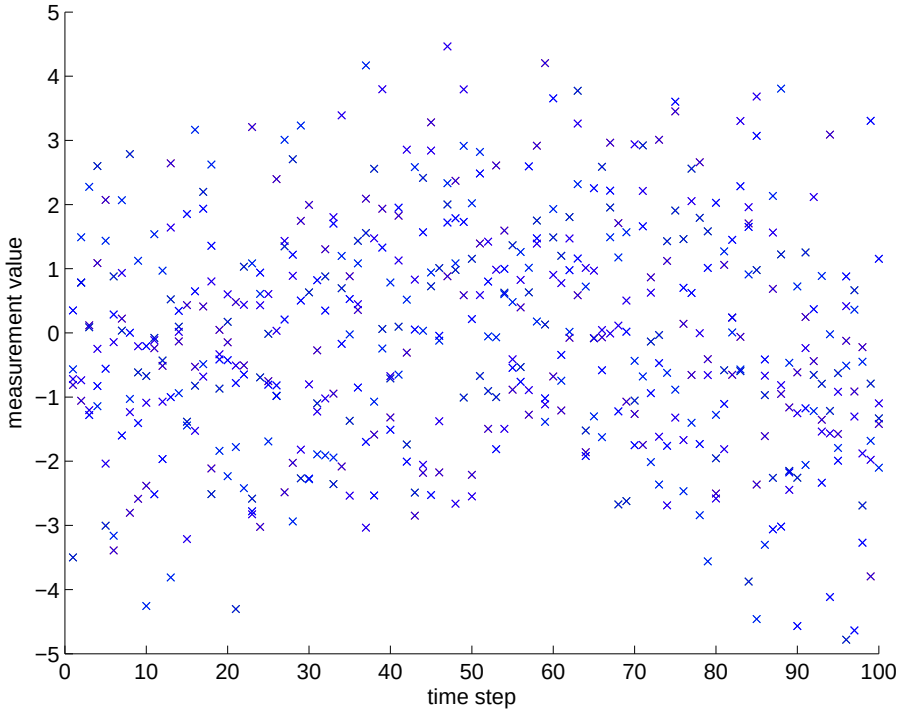


Figure 5.4: Example trial with $f = 1$

5.7.1 Assumed Assignment State Model

Both the PMHT-ym and PMHT-ye use the same assignment state model for this example. The model has $M_D = 11$ and

$$\phi_t^m\left(d_t^m\right) = \frac{d_t^m}{10}, \qquad 0 \leq d_t^m \leq M_D - 1. \tag{5.56}$$

Thus the π_t^m are modelled as one of $0, 0.1, \ldots 0.9, 1.0$.

The transition matrix is assumed to be tri-diagonal with the diagonal elements at 0.5 and the elements one step off-diagonal at 0.25. This means that the model assumes that the prior changes only slowly, and does not allow a step of more than 0.1 from one scan to the next. Using a strongly diagonal transition matrix of this form imposes a low-pass filter effect on the estimated π_t^m. This has the desired effect of smoothing out the estimate. Choosing a transition matrix with strong off-diagonal terms could be used to impose a high-pass filter on the estimated π_t^m. As a limiting case, a matrix with zero along the diagonal implies that the prior never remains constant.

5.7.2 State Estimation Performance

The bias and variance of the state estimate are estimated from the sample mean and variance[2]. These values are given in tables 5.1 and 5.2 respectively.

All of the algorithms have a bias in the mean which pushes the two state estimates apart: the bias in model 1 (which has a true mean of 1) is always positive, and the bias in model 2 (which has a true mean of -1) is always negative. Measurements from one tail of model 1 are assigned to model 2, and this skews the estimated parameters away from

[2]Chapter 4 gives more information about the sample bias and variance in section 4.4.2

	$x^1[\mu]$	$x^1[\sigma^2]$	$x^2[\mu]$	$x^2[\sigma^2]$
$f = 1$				
standard PMHT	0.25	-0.54	-0.24	-0.54
PMHT-ym	0.040	-0.10	-0.061	-0.13
PMHT-ye	0.15	-0.33	-0.16	-0.35
$f = 4$				
standard PMHT	0.25	-0.55	-0.24	-0.55
PMHT-ym	0.097	-0.22	-0.11	-0.22
PMHT-ye	0.15	-0.35	-0.17	-0.36

Table 5.1: State estimation bias

	$x^1[\mu]$	$x^1[\sigma^2]$	$x^2[\mu]$	$x^2[\sigma^2]$
$f = 1$				
standard PMHT	0.019	0.039	0.017	0.040
PMHT-ym	0.030	0.10	0.026	0.10
PMHT-ye	0.012	0.061	0.011	0.060
$f = 4$				
standard PMHT	0.019	0.038	0.018	0.038
PMHT-ym	0.034	0.090	0.031	0.086
PMHT-ye	0.012	0.060	0.012	0.060

Table 5.2: State estimation variance

model 2. Measurements from the tail of model 2 assigned to model 1 also adds to this effect.

In a similar way, all algorithms consistently underestimate the variance of the two models. This occurs for the same reason as above: outliers from one model are assigned to the other, decreasing the variance of the set of measurement assigned to the model.

It is also worth noting that the biasses in table 5.1 are very significant when compared with the true state values. The standard PMHT shows approximately a twenty five percent bias, the PMHT-ye a fifteen percent bias, and the PMHT-ym five percent.

In both examples, the PMHT-ym gives the lowest state estimation bias. The bias of the PMHT-ym is significantly less than the standard PMHT. The bias of the PMHT-ym increases when the variation in the prior becomes more rapid (i.e. $f = 4$). The PMHT-ye gives performance between the two.

In contrast to the bias performance, the PMHT-ym gives the worst variance, as shown in table 5.2. The PMHT-ym variance is generally more than twice the standard variance. The PMHT-ye has variance performance better than PMHT for the mean, but worse for the variance. The variance does not significantly change with the variability of the prior. The variance values are also significant compared with the true state values. The standard deviation in the mean estimate is fourteen percent for the standard PMHT, seventeen percent for the PMHT-ym, and eleven percent for the PMHT-ye.

The bias and variance can be combined to form the model state estimation root mean

	$x^1[\mu]$	$x^1[\sigma^2]$	$x^2[\mu]$	$x^2[\sigma^2]$
$f = 1$				
standard PMHT	0.29	0.58	0.27	0.58
PMHT-ym	0.18	0.33	0.17	0.34
PMHT-ye	0.19	0.41	0.19	0.43
$f = 4$				
standard PMHT	0.29	0.58	0.28	0.58
PMHT-ym	0.21	0.37	0.21	0.37
PMHT-ye	0.19	0.42	0.20	0.44

Table 5.3: State estimation RMS error

square (RMS) error, represented by ϵ. The estimated RMS error is given by

$$\hat{\epsilon} = \sqrt{\hat{\sigma^2} + \left(\hat{B}\right)^2}. \tag{5.57}$$

Combining tables 5.1 and 5.2 to form RMS estimates gives table 5.3. In terms of RMS error, both the PMHT-ym and PMHT-ye perform better than the standard PMHT for all parameters, for both frequency values, f. Both the PMHT-ym and the PMHT-ye have similar errors for the mean parameter, $\boldsymbol{x}^m[\mu]$, but the PMHT-ym gives significantly better performance for the variance parameter, $x^m[\sigma^2]$. This is a reflection of the bias having more of a contribution than the variance to the RMS result.

Overall, PMHT-ym gives the best model state estimation performance, closely followed by PMHT-ye.

5.7.3 Prior Estimation Performance

The bias and variance of the estimated π_t^m values is now examined for each of the filters. As mentioned previously, the PMHT-ym does not produce an estimate of the prior, rather it gives the probability mass of it. For this comparison, an estimate is formed by taking the mean of this mass. Note that this means that the PMHT and PMHT-ym estimates are continuous valued quantities whereas the PMHT-ye will always provide an estimate based on the model, which restricts π_t^m to $0, 0.1, \ldots 1.0$.

5.7.3.1 Low Frequency Prior Variation

Figure 5.5 shows an example of the estimated priors for each of the algorithms. The figure shows the prior probability for model 1 (the probability for model 2 is the compliment of this value). The PMHT estimate is shown as a solid line, the PMHT-ym as a dashed line, and the PMHT-ye as a dotted line. The dash-dot line shows the true value of the prior. The discretised nature of the PMHT-ye estimate is immediately obvious from the staircase shape of the corresponding curve. The size of the jumps could be trivially reduced by increasing the dimension of the assignment state model (i.e. M_D). The variation in the PMHT estimate is substantial and the estimated π_t^m that it produces is erratic.

Figure 5.6 shows the probability mass function estimated by the PMHT-ym.

The results in figures 5.5 and 5.6 show the performance from a single random trial. The bias and variance of the estimators are estimated using the sample estimators from

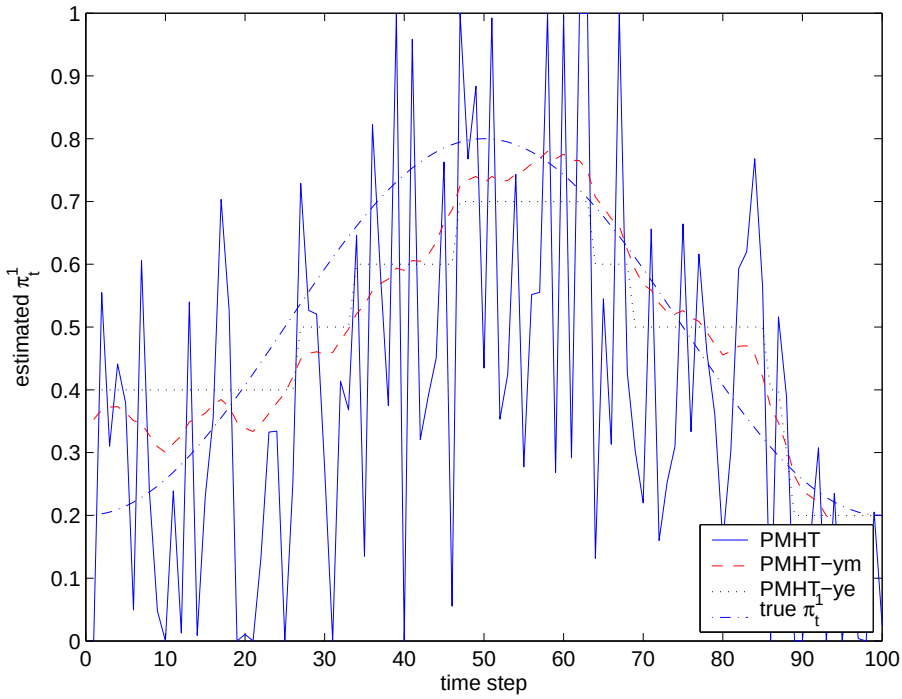


Figure 5.5: Single trial prior estimation, $f = 1$

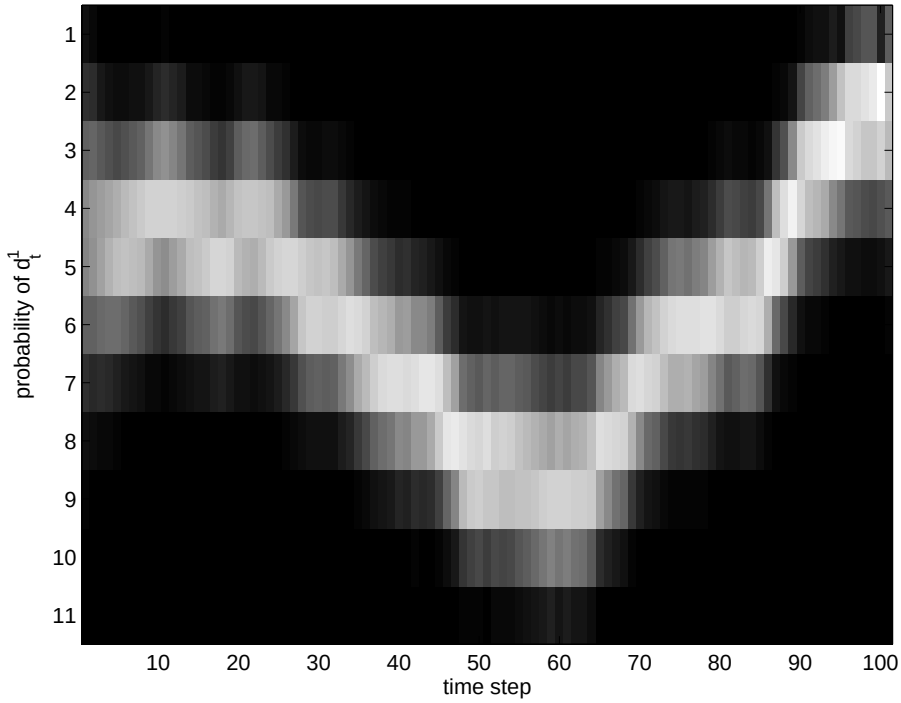


Figure 5.6: PMHT-ym prior probability mass

1000 Monte Carlo realisations. The bias of each of the filters is shown in figure 5.7 and the variance in figure 5.8.

The bias of the PMHT-ym is marginally better than the standard PMHT, and that of the PMHT-ye is marginally worse. There is no significant difference between the three algorithms in bias. However, the reduction in variance by using the assignment state model is dramatic. Both the PMHT-ym and PMHT-ye reduce the variance by an order of magnitude over the standard PMHT. This result is clear from the single realisation shown in figure 5.5. The high variance of the PMHT is clearly seen in the erratic variability noted earlier. The high variance of the standard PMHT is also seen by the variability in the bias curve in figure 5.7. This curve is averaged over 1000 trials, and the variance of the average is still apparent in the figure.

The variance of the PMHT estimate can be clearly seen to peak when the true prior is 0.5 and to have minima when it reaches the most one-sided points ($\pi_t^1 = 0.2$ and $\pi_t^1 = 0.8$). This is because the filter is estimating a binomial process, and the variance of that estimate is expected to follow a $\pi_t^1 \ (1 - \pi_t^1)$ curve analogous to that derived in section 4.4.2.

One reason for the dramatic reduction in variance is that the dynamics of the assignment state model allows the filter to effectively use measurements from adjacent scans to estimate the prior for any particular scan. This is made quite obvious from the expression maximised by the HMM smoother, i.e. $P\left(\boldsymbol{D}_t|\mathbf{X},\mathbf{Z}\right)$. This is the probability of assignment state at scan t given measurements over the whole batch. The effect of measurements from different scans is reduced the farther that scan is in time, depending on the diffuseness of the transition matrix, $\Delta_t^m\left(d_t^m|d_{t-1}^m\right)$. The standard PMHT uses measurements only from one scan, since it assumes that the prior at other scans is independent.

5.7.3.2 Higher Frequency Prior Variation

The same bias and variance curves are generated for a higher frequency variation in π_t^m, with $f = 4$. The estimates from a single trial are shown in figure 5.9. In this case both the PMHT-ym and PMHT-ye appear to underestimate the true variation in the prior. The two curves clearly vary at the same frequency as the true prior, but the departure from $\pi_t^m = 0.5$ is reduced. This is actually a manifestation of the low-pass filtering property of the transition matrix chosen for the assignment state model. The variation in this example is at a rate that shows significant attenuation by the filter, but not high enough to be suppressed fully. This is also seen in the bias curve in figure 5.10. However, in figure 5.11, the variance of the PMHT-ym and PMHT-ye estimates remains an order of magnitude below that of the standard PMHT.

5.8 Summary

This chapter introduced a dynamic model for the assignment prior probability. This model is referred to as hysteresis. The hysteresis model uses an independent discrete state variable for each target model, so the assignment process is modelled with a M_X dimensional randomly evolving hyperparameter. Two alternative approaches for solving the generalised problem posed by the introduction of this model were considered: firstly, the state value was treated as missing data in an EM sense, and secondly, the state was estimated.

The PMHT algorithm derived by treating the assignment state as missing data is referred to as the PMHT-ym. The PMHT-ym determines the probability of the assign-

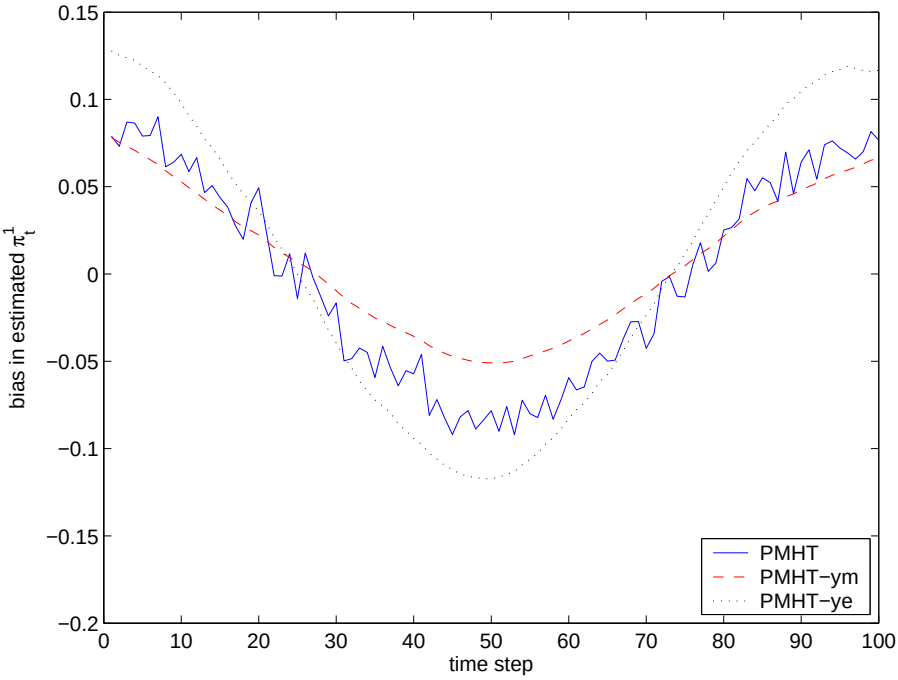


Figure 5.7: Bias of estimator $\hat{\pi}_t^1$, $f = 1$

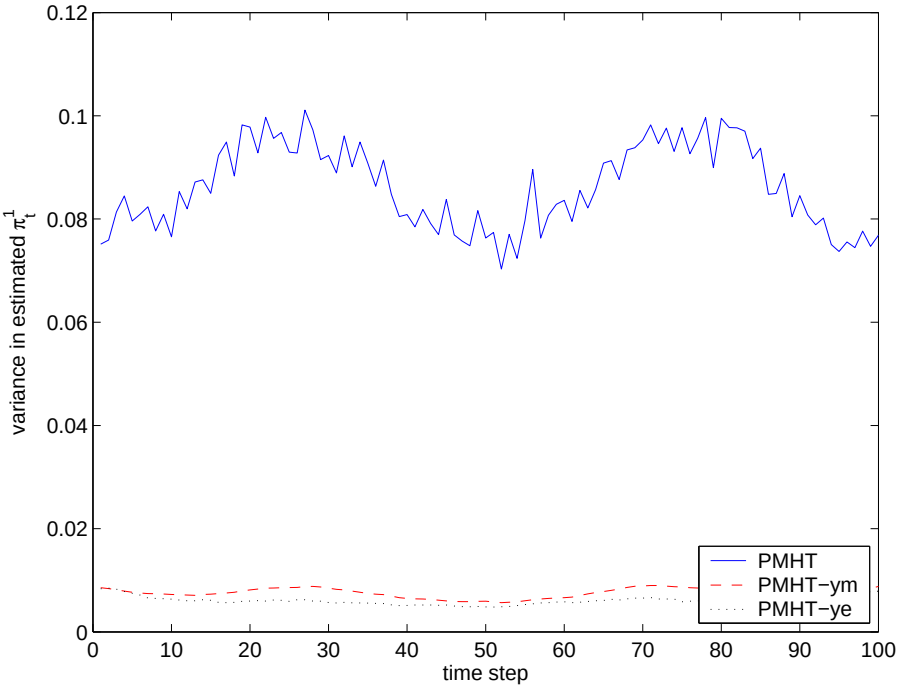


Figure 5.8: Variance of estimator $\hat{\pi}_t^1$, $f = 1$

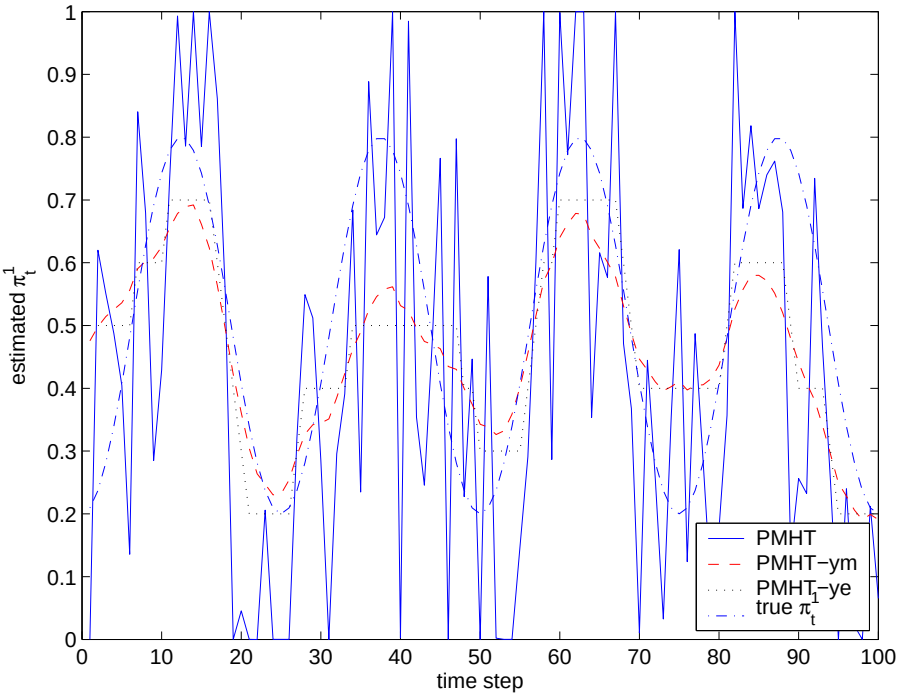


Figure 5.9: Single trial prior estimation, $f = 4$

ment state, in much the same way as the standard PMHT determines the probability of the assignments. This probability can be calculated using the Hidden Markov Model Smoother.

The PMHT algorithm derived by estimating the assignment state values is referred to as the PMHT-ye. The PMHT-ye treats the assignment state in the same way as the model states: an initial value is assumed, and then the algorithm iteratively updates the assignment state estimate based on the measurements. The refined assignment state sequence can be determined using the Viterbi algorithm.

The PMHT-ym and PMHT-ye were run on a simple simulated example to demonstrate the effect each has on the estimated assignment prior. The algorithms were run on mixtures with low and high frequency variation in their mixing proportions. It was found that by assuming a certain degree of smoothness in the hysteresis model, a low pass filtering effect was achieved. The PMHT-ym and PMHT-ye both apply a restriction on the rate of change of the prior, based on the assumed hysteresis model, which suppresses higher frequency variations. Both algorithms dramatically reduce the variance over the standard PMHT which assumes that the assignment prior is time independent. This reduction in variance is achieved because data from adjacent scans contributes to the prior estimate through the assumed dynamic model.

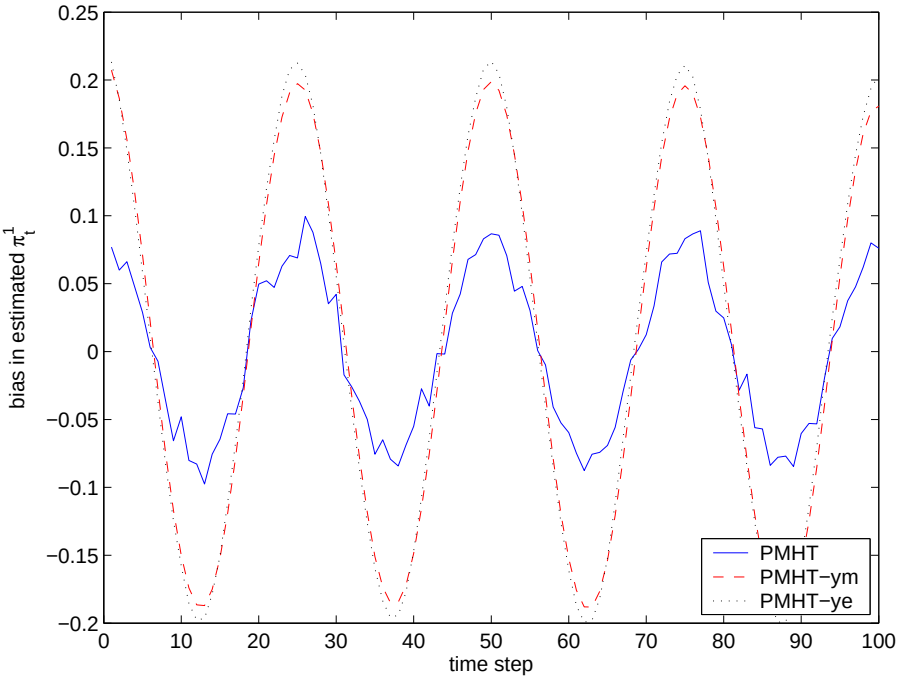


Figure 5.10: Bias of estimator $\hat{\pi}_t^1$, $f = 4$

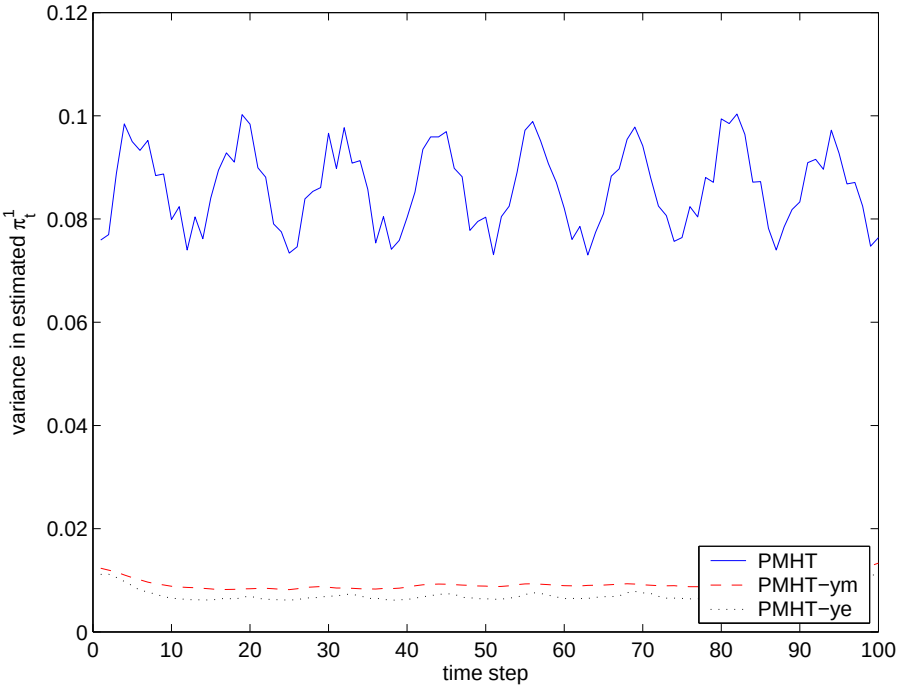


Figure 5.11: Variance of estimator $\hat{\pi}_t^1$, $f = 4$

Chapter 6

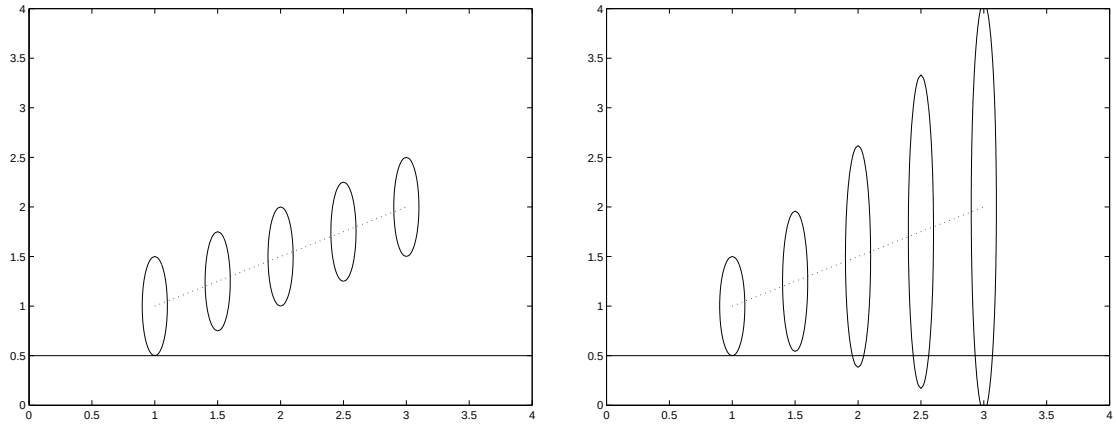
Track Initiation and Initialisation with the Probabilistic Multi-Hypothesis Tracker

The original Probabilistic Multi-Hypothesis Tracker derived by Streit and Luginbuhl makes the assumption that the number of models is fixed and known. Further, it assumes that the tracker has available a prior distribution for the state of each of these models. In a realistic tracking situation, targets enter and leave the field of view of the sensor for various reasons. This means that the number of models that the tracker should use (namely the number of targets in the scene) is an unknown and dynamic quantity. Since new targets appear inside the sensor field of view at unknown locations, no prior distribution of their state is known, so a prior must be assumed based on sensor data.

In order to use the PMHT algorithm in an online implementation, it is necessary to develop a method of initiating new tracks and terminating old tracks as new data arrives. This function may be performed manually by the sensor operator, however this may be time consuming and manual track management is only used if automatic management is unavailable or unreliable . It is desirable for the tracking algorithm to be able to automatically perform the tasks of initiation and termination without operator intervention.

In addition to track initiation, the tracker must also be able to perform self initialisation. Namely the tracker should choose the initial state estimate automatically. The PMHT is a numerical optimisation approach that iteratively converges to the Maximum Likelihood state estimate. However, it converges only to local maxima, and the initialisation of the algorithm is therefore critical. Since the algorithm is required to initiate new tracks, it must provide a method for initialising the state estimates of these tracks. The problem of poor initialisation is similar to the situation where a target manoeuvres and the previously accurate state estimate becomes biassed. Thus, incorporation of an automatic initialisation scheme is anticipated to also improve tracking of manoeuvering targets.

This chapter presents methods for addressing track initiation and initialisation with the PMHT. Firstly, an initialisation technique is introduced which is based on the homothetic measurement model of [RWS95a]. This method uses the innovation covariance matrix as a second measurement model. Since the innovation covariance matrix represents the current uncertainty in the predicted measurement location, this method enables the PMHT to use a broad secondary measurement model when initialisation is poor, and to narrow this secondary model when good initialisation is obtained.



(a) measurement covariance (b) innovation covariance

Figure 6.1: Growth of the innovation covariance under poor initialisation

A general framework for track initiation and termination is then presented. Two alternative approaches using the standard PMHT are described. These approaches make use of existing features of the PMHT algorithm. A third approach is the application of the Hysteresis model presented in the previous chapter. Under this approach, the assignment state itself can be used as a statistic for significance testing of proposed new target tracks.

The different methods for track initiation with the PMHT are compared via simulation studies and a preferred method is identified.

6.1 Innovation Homothetic Measurement Model for Initialisation

One of the problems with the PMHT algorithm is that it requires good initialisation of the target state estimate to avoid convergence to a local maximum. In many practical situations, it is not possible to obtain an accurate initial estimate of the target states (otherwise tracking would not be required), so a scheme to reduce the sensitivity to initialisation is required. One cause of this sensitivity is that the PMHT associates measurements using the measurement covariance. This is because the algorithm performs assignment assuming that the current estimate is the true target state. This may cause target measurements to be given a low weight if the initial state estimate is poor. Most other tracking algorithms (for example PDAF) associate measurements using the innovation covariance, which gives the scatter of measurements conditioned on the current state estimate. When the current estimate is poor, the innovation covariance matrix allows measurements more distant to have high association probabilities, which allows the algorithm to correct the estimate. Figure 6.1 demonstrates a simplistic example of how the innovation covariance matrix can grow when initialisation is poor and allows algorithms such as PDAF to recover from this condition. In the figure, a solid line represents the true target position, and the dotted line the initialised trajectory. Elipses are drawn to show the size of the covariance matrix used for assignment and hence give a qualitative representation of the weights target measurements might be given.

One way to improve global convergence of the PMHT is to use inflated covariance values for the measurement process. This has the effect of smearing the likelihood surface and local maxima may become hidden by the spread global maximum. After convergence

with the inflated measurement covariance, the covariance is reduced somewhat and the algorithm run again. This process is repeated until the final iteration where the true measurement covariance matrix is used. This method requires a choice of how much to inflate the measurement covariance matrix and how quickly to approach the true value. Since the batch of data is processed several times, use of covariance inflation incurs a significant computation cost.

To reduce the extra computational load required by covariance inflation, the degree of inflation can be made adaptive. One way of adapting the inflated covariance is to use the innovation covariance associated with the current state estimate. This allows higher covariance inflation for tracks with greater uncertainty.

The covariance inflation approach can also be automated by using the homothetic measurement model proposed in [RWS95a]. Under the homothetic measurement model, the measurement process for each target is itself a mixture of Gaussians. Usually the mixture is composed of Gaussians that share a common mean, although this is not required. The homothetic measurement model can be used as a method of covariance inflation by making the measurement process a mixture of a Gaussian with covariance defined by the assumed "true" measurement covariance and a second Gaussian with covariance given by the innovation covariance of the current state estimate. When the track estimate is first initialised, it is assigned a high degree of uncertainty and the innovation covariance is high. As the EM iterations are performed, the smoothed state estimate gains higher accuracy and the innovation covariance is reduced. When the smoothed state estimates are sufficiently close to the observed measurements, the algorithm associates the measurements with the tighter density that uses the "true" measurement variance.

Figure 6.2 demonstrates how this approach allows the PMHT to recover from poor initialisation through a mechanism similar to the PDAF shown in figure 6.1. The figure shows a sequence of plots. In each plot, false detections are shown with the symbol 'x' and valid detections '+'. The estimated trajectory for that iteration of the PMHT is drawn with a solid line, and elipses are drawn to show the innovation covariance. As the state estimate improves, the elipses shrink, indicating the the innovation covariance matrix corresponds to a tighter measurement scatter.

The standard homothetic measurement model for PMHT uses measurement covariance matrices which are scalar multiples of the true measurement covariance. This results in state estimation with the Kalman Smoother where the measurement covariance is replaced by a scalar multiple of the true measurement variance. The assignment weights determine the scaling factor. The smoother uses a synthetic measurement which is a weighted sum of the sensor observations. As presented in chapter 3, these are given by

$$\tilde{\boldsymbol{z}}_t^m = \left\{ \sum_{r=1}^{n_t} \sum_{p=1}^{P} \frac{w_{mptr}}{\kappa^{mp}} \right\}^{-1} \sum_{r=1}^{n_t} \left(\sum_{p=1}^{P} \frac{w_{mptr}}{\kappa^{mp}} \right) \boldsymbol{z}_{tr}, \tag{6.1}$$

and

$$\tilde{\mathsf{R}}_t^m = \left\{ \sum_{r=1}^{n_t} \sum_{p=1}^{P} \frac{w_{mptr}}{\kappa^{mp}} \right\}^{-1} \mathsf{R}_t^m, \tag{6.2}$$

where κ^{mp} is the scalar multiplier for the measurement variance associated with homothetic model p and w_{mptr} is the assignment weight for model p of track m.

The results in (6.1) and (6.2) do not hold when the second model is the innovation matrix because the innovation covariance is not necessarily a scalar multiple of the

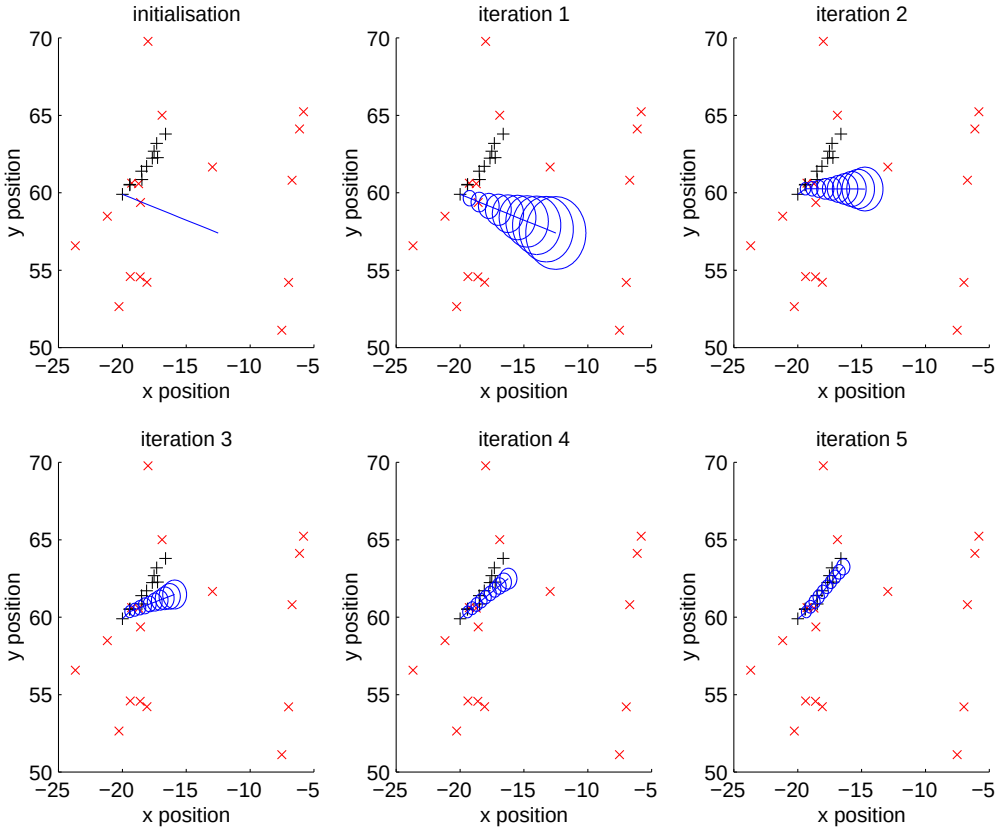


Figure 6.2: Correction for poor initialisation with innovation homothetic model

measurement variance. However, the state estimate can still be solved using a Kalman Smoother. The smoother uses a synthetic measurement and covariance defined by

$$\tilde{\boldsymbol{z}}_t^m = \tilde{\mathsf{R}}_t^m \left\{ \sum_{r=1}^{n_t} \left[w_{m1tr} \left(\mathsf{R}_t^m\right)^{-1} + w_{m2tr} \left(\mathsf{S}_t^m\right)^{-1} \right] \boldsymbol{z}_{tr} \right\}, \tag{6.3}$$

and

$$\left(\tilde{\mathsf{R}}_t^m\right)^{-1} = \sum_{r=1}^{n_t} w_{m1tr} \left(\mathsf{R}_t^m\right)^{-1} + \sum_{r=1}^{n_t} w_{m2tr} \left(\mathsf{S}_t^m\right)^{-1}, \tag{6.4}$$

where S_t^m is the innovation covariance matrix for model m at scan t.

The expressions (6.3) and (6.4) are more generalised versions of those in (6.1) and (6.2) respectively. They simplify to the latter expressions if the matrix S_t^m can be written as $\kappa\mathsf{R}_t^m$, with κ constant. The derivation of (6.3) and (6.4) is presented, along with a more detailed discussion of this generalised homothetic model in Appendix A.

6.2 Initiation Methodology

The process of creating new tracks when new targets are detected is called track initiation, or track formation. There are numerous techniques for solving the track initiation problem, but each of them can be formulated under a common framework of candidate

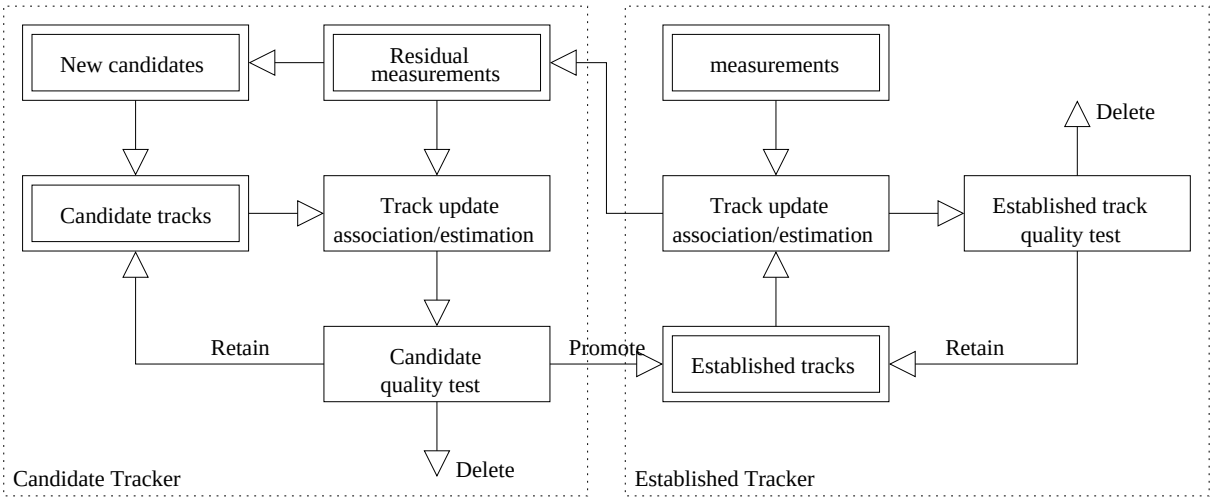


Figure 6.3: General initiation flow diagram

tracking. Using the radar measurements, candidate tracks are initialised at possible target locations. Candidate tracks might be started on all measurements, or on a subset, according to a selection rule. For example, one might choose measurements not assigned to an existing target model which are sufficiently close to a measurement from the previous scan. However formed, the candidate tracks are allowed to assign measurements and are updated. After a number of scans, the candidates are tested using a decision rule. Based on this test, candidates may be discarded or promoted to established tracks. The decision to promote or discard may be deferred, and the candidate retained.

Figure 6.3 shows a typical scheme for tracking with automatic initiation. In figure 6.3 established tracks, which were initiated at some earlier time, get the first pickings of the sensor measurements. These tracks remove measurements that have a high assignment probability and the remaining measurements (the residual measurements) are used to form candidate tracks. The candidate tracks perform association and state estimation using the residual measurements. This prevents the candidate tracker from running a candidate track on the same target as is represented by an established track.

A possible fault with the scheme shown in figure 6.3 is that the association of measurements and the updating of state estimates should perhaps be done jointly for all tracks rather than in the sequential manner shown. A scheme performing this joint association and update is shown in figure 6.4. The danger with the sequential approach is that an established track will be assigned measurements from more than one target and it will prevent new tracks forming on the other targets. This is likely to occur with the joint scheme anyway since the new candidates must be formed from the residual measurements, otherwise multiple tracks will be formed on every target.

A danger with the joint scheme is that candidate tracks will degrade the established tracks. Suppose there is a track following a target that performs a heading change. The established track may take a few scans to adjust heading to the new trajectory. In the meantime, a candidate track has been started on the new heading and is very close to the measurements since they were used to initialise it. The candidate track is assigned the measurements in preference to the established track since the established track is farther from the correct state estimate. The consequence is that the established track will be terminated and the new candidate track will be promoted. This is undesirable performance which is easily prevented by using the sequential approach.

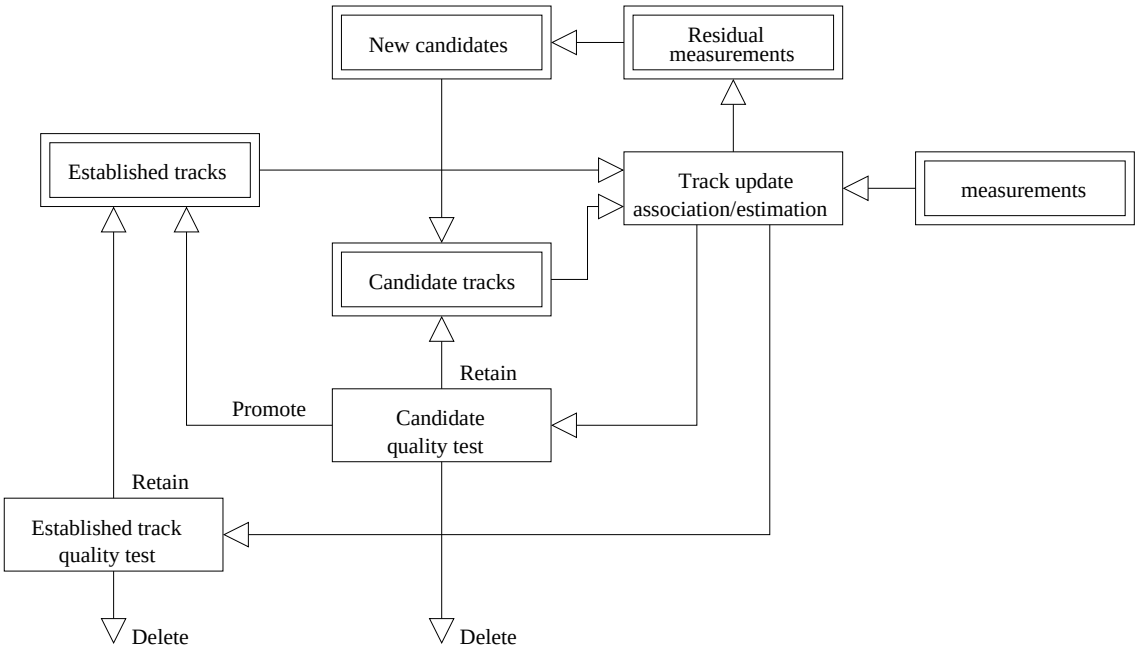


Figure 6.4: Alternative initiation flow diagram

A key component of either track initiation approach is the decision criteria for promoting candidate tracks. Given a particular algorithm for association and state estimation, the obvious question is: what decision criterion for candidate promotion gives the best performance? To start to answer this question, track initiation performance must be objectively defined. The design of the promotion test must balance two conflicting requirements: low probability of promoting a track that is false, and high probability of promoting a track that is valid. The first requires a high test threshold, whereas the second requires a low test threshold. In the following discussion, assume that the candidate promotion test is not allowed to defer its decision, instead the candidate tracker is given a short batch of data and required to make hard decisions about the candidate tracks at the end of the batch.

6.3 Candidate Tests using the Standard PMHT

The simplest way to perform track initiation is to use information already provided by the PMHT algorithm to test candidate tracks. Two such approaches are now introduced.

6.3.1 Sum of Weights Quality Statistic

A common test for promoting candidate tracks is referred to as an M of N rule. Under an M of N rule, candidates are promoted if they receive M validated measurements in N scans. This rule is also used to select potential measurements with which to create candidates. In this case, a candidate is formed if there are M measurements within a gate volume in N scans. Rather than simply counting the number of measurements that are within an arbitrary distance of the track, a more intuitively appealing approach is to test the sum of the association probabilities. This has the advantage of not promoting a track simply because it is in a high clutter environment. Also it gives a greater choice of

promotion threshold, since the test statistic is no longer integer valued. This leads to a simple ad hoc quality statistic given by the sum of the association weights:

$$q_w^m = \sum_{t=1}^{T} \sum_{r=1}^{n_t} w_{mtr}. \tag{6.5}$$

This statistic can be interpreted as the estimated number of measurements caused by model m. When normalised by the total number of measurements in the batch, q_w^m is also the maximum likelihood estimator for the prior probability of a measurement being caused by model m if this prior is time invariant.

This simple quality statistic will be referred to as the *sum of weights* quality statistic (for obvious reasons) and is used as a benchmark for other proposed initiation schemes. The sum of weights test could be implemented with no alteration to the PMHT algorithm and carries very little overheads.

6.3.2 Cost Function Increment

Another commonly used quality test for candidate tracks is the cumulative log likelihood of the assigned measurement sequence [Sit64]. The log likelihood can be determined by accumulating the squared innovations when the statistics are assumed to be Gaussian, i.e.

$$q^m = -\frac{1}{T} \sum_{t=1}^{T} \sum_{r=1}^{n_t} w_{mtr} \left(\mathsf{H}_t^m \mathsf{F}_t^m \boldsymbol{x}_{t-1}^m - \boldsymbol{z}_{tr}\right)^{\mathsf{T}} \left(\mathsf{S}_t^m\right)^{-1} \left(\mathsf{H}_t^m \mathsf{F}_t^m \boldsymbol{x}_{t-1}^m - \boldsymbol{z}_{tr}\right). \tag{6.6}$$

Intuitively, false tracks are expected to be assigned measurements that match the target dynamic model poorly and have a large scatter about the predicted measurements. Such tracks will have a high accumulated innovation and this can be used to reject them. This statistic gives a measure of how well the measurements assigned to a track fit the assumed target model.

An analogue of this approach for the PMHT would be to use the component of the auxiliary function corresponding to that model as the quality statistic, namely

$$\begin{aligned} q^m &= Q_X^m \\ &= \log \psi_0^l\left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T} \log \psi_t^m\left(\boldsymbol{x}_t^m | \boldsymbol{x}_{t-1}^m\right) + \sum_{t=1}^{T} \sum_{r=1}^{n_t} \log \zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^m\right) w_{mtr}. \end{aligned} \tag{6.7}$$

Note that an iteration index is not included on the weights because the quality statistic is only calculated on convergence of the algorithm.

There are two main problems with this approach. Firstly, a track with very few measurements assigned to it will clearly not have very high measurement scatter and so be treated as a good candidate track. Such tracks are not good candidates; the lack of assigned measurements is a strong indicator that the track is only assigned clutter measurements. To deal with this case, the log likelihood test must be augmented with a logical test to require the candidate to assign a certain number of measurements.

The second problem is that the test is centred on the track itself. The goal of adding a new track is not to add a new model which assigns measurements. Rather it is to add a model which accounts for measurements which seem unlikely under the current mixture model. To this end, the test statistic should encompass the overall improvement in the data description by the inclusion of a new model. This point is best illustrated

with an example. Suppose there is a sequence of valid target measurements. A target model on these measurements will give a good quality statistic. Further suppose that a target model assigns most of these measurements, but that a candidate track is initialised on them anyway. The candidate will be given significant weights for the measurements, and the weights for the pre-existing track will be reduced. Depending on the promotion threshold, this new candidate may be promoted even though it adds nothing to the overall description of the data.

Both of the above problems can be addressed by using the total system likelihood as the quality statistic, rather than the individual model likelihood. Under this approach, the quality assigned to a new candidate is the improvement in the overall likelihood obtained by adding the candidate. This improvement is given by

$$q_Q^c = Q\left(\mathbf{X}, \mathbf{X}^c\right) - Q\left(\mathbf{X}\right), \tag{6.8}$$

where X^c is the candidate state sequence and X are the states of the other models.

The addition of the candidate model is guaranteed to increase the auxiliary function, and the amount of this increase provides a measure of how much the candidate model adds to the overall data description. This approach was used to estimate the number of components and the parameters of a Gaussian mixture with an unknown number of components by Vlasis et al. in [VLK00, VL00].

6.3.2.1 Comparison with Model Order Estimation

The track initiation problem is really one of model order estimation. The aim is to find the best set of dynamic Gaussian models to describe the sensor data. The solution requires both the determination of the states of each component and an estimation of how many components provides the best fit to the data. This latter part is model order estimation.

It is not possible to use standard optimisation approaches for model order estimation. Optimisation criteria such as Maximum Likelihood or Least Squared Error will always prefer a model of higher order. Ultimately, such methods will find the optimal model order to be any number at least as large as the number of data points. Once the number of parameters is more than the number of data points, adding further complexity produces no change in the optimising objective function. To alleviate this shortcoming, model order estimation approaches incorporate a penalty term which reduces the objective function when a higher order model is used. This penalised objective function is then of the form

$$\tilde{Q}(\mathbf{X}) = \log L\left(\mathbf{X}, \mathbf{Z}\right) - f(d(\mathbf{X}), N), \tag{6.9}$$

where d is the total number of parameters (the sum of the lengths of all the state vectors) and N is the total number of data points, namely $N = \sum_t n_t$. L is the maximum value of the likelihood of the data under model X, i.e. the likelihood when X is set to the ML estimate. The model order is then that integer d which maximises the function $\tilde{Q}$.

The *A Information Criterion* (AIC) was developed by Akaike [Aka74] by minimising the Kullback-Leibler distance between the true data model and the estimated model. The derivation of the AIC relies on asymptotic arguments. The AIC is given by

$$\tilde{Q}_{\mathsf{AIC}} = \log L\left(\mathbf{X}, \mathbf{Z}\right) - 2d(\mathbf{X}). \tag{6.10}$$

A problem with the AIC is that the penalty term is independent of the size of the data and so the model order will be overestimated when N is large. Both Schwartz [Sch78] and Rissanen [Ris78] are critical of the arguments used to derive the AIC.

An alternative method, the *Bayes Information Criterion* (BIC) was presented by Schwartz [Sch78]. This approach is derived using asymptotic arguments about a Bayesian model order technique. The BIC is given by

$$\tilde{Q}_{\mathsf{BIC}} = \log L\left(\mathbf{X}, \mathbf{Z}\right) - \frac{\log N}{2} d(\mathbf{X}). \tag{6.11}$$

The Minimum Description Length (MDL) of Rissanen [Ris78] was derived by finding the smallest number of bits to represent an ARMA data sequence. This criterion is the same as the BIC except that it includes an additional term which takes into account model complexity (not all d parameter models are the same). However, this term is independent of the data size and becomes insignificant for large data sets. The MDL was also derived using asymptotic arguments.

These standard model order estimation techniques each have a penalty term that is linear in the model order. So, the AIC, BIC and MDL (for large data sets) can be written as

$$\tilde{Q} = \log L\left(\mathbf{X}, \mathbf{Z}\right) - \epsilon\ d(\mathbf{X}), \tag{6.12}$$

where ϵ varies depending on the selection criterion.

This criterion is equivalent to choosing a model of order d over a model of order $d-1$ when the log likelihood under model d is at least ϵ more than the log likelihood under model $d-1$:

$$\begin{aligned} \tilde{Q}(d) &> \tilde{Q}(d-1) \\ \log L\left(\mathbf{X}(d), \mathbf{Z}\right) - \epsilon d &> \log L\left(\mathbf{X}(d-1), \mathbf{Z}\right) - \epsilon(d-1) \\ \log L\left(\mathbf{X}(d), \mathbf{Z}\right) - \log L\left(\mathbf{X}(d-1), \mathbf{Z}\right) &> \epsilon. \end{aligned} \tag{6.13}$$

The model order estimation above does not involve a data association problem. When data association is introduced, the likelihood can be replaced by its expectation over the assignment hypotheses. It follows that (6.13) can be written as

$$Q\left(\mathbf{X}, \mathbf{X}^c\right) - Q\left(\mathbf{X}\right) > C\epsilon, \tag{6.14}$$

where C is the order of the candidate model. Thus, approaches such as AIC and MDL are equivalent to the incremental cost test as described above, but where a particular decision threshold is chosen. In general, Neyman-Pearson type detector laws may be desirable, where the threshold is chosen to specify a particular false alarm rate, rather than an optimal decision. Thus it is desirable in some instances to choose the decision threshold.

6.4 PMHT with Hysteresis for Initiation

An alternative to using outputs of the standard PMHT as a candidate test is to develop a modified algorithm that builds in a track quality measure by design. The sum of the assignment weights for model m is a scaled version of the standard PMHT estimate for the assignment prior, π_t^m. The prior is an intuitively appealing measure of candidate quality since it specifies the contribution of each model to the measurement set. The previous chapter demonstrated that the PMHT with Hysteresis can give superior prior estimation performance over the standard PMHT. This is because it provides the PMHT framework with a method for modelling the temporal behaviour of the assignment process. One fault with the weights sum approach is that it gives equal importance to a candidate which has

two assigned measurements in one scan, and a candidate which assigns one measurement per scan in two consecutive scans. It is evident that the latter is more likely to be a valid track. When the Hysteresis model is used, the Markov chain of the assignment process can be designed so that the assignment state itself is an in-built quality measure of the candidate.

When a candidate track is valid, then it is detected with probability Pd and if it is detected, then it forms one measurement. Thus, the true π_t^m of a valid candidate is $\frac{Pd}{n_t}$. If the candidate is false, then the true π_t^m is zero. An obvious quality test for candidates is thus the estimated π_t^m value. This is the weights sum approach already discussed. However, the estimated π_t^m has a high variance, as demonstrated in the previous chapter. This then motivates the use of the PMHT-y model for smoothing the π_t^m estimate, and was actually the problem which inspired its development.

6.4.1 Initiation with PMHT-ym

There are two types of candidate tracks: valid and false. Since this analysis assumes a fixed probability of detection Pd, which is the same for all targets, then there are two true values of π_t^m, as stated above. An intuitive decision is thus to use an assignment state space with two possible state values corresponding to false and valid tracks, i.e. $M_D = 2$. This is the simplest possible model, and has the benefit of minimising the computational requirements of the algorithm. Under this model,

$$\begin{aligned} \phi_t^m(0) &= 0, \\ \phi_t^m(1) &= \frac{Pd}{n_t}. \end{aligned} \tag{6.15}$$

This special case, with a binary assignment state model, is the same as the visibility model used for initiation with the PDAF and other filters. Section 5.2.1 discusses the relationship between the visibility approach and the hysteresis assignment model. For this special case, the assignment state for model m, d_t^m, will be referred to as the visibility of model m and the PMHT-ym using this model is referred to as the PMHT with visibility, or PMHT-v.

The PMHT-v calculates the posterior probability mass $P\left(d_t^m = 1\right)$, which is the probability that the candidate has a prior consistent with a valid track. This then provides a natural quality statistic. Since the visibility variable can change over the batch, the candidate test statistic is

$$q_v^m = \frac{1}{T} \sum_{t=1}^{T} P\left(d_t^m = 1 | \mathbf{X}, \mathbf{Z}\right). \tag{6.16}$$

All that remains is the selection of the transition probability matrix and the prior probability mass. Since the model is binary ($M_D = 2$) these are fully specified by three parameters: $\Delta_0^m\left(d_0^m\right)$ has two elements, one of which is constrained by normalisation, and $\Delta_t^m\left(d_t^m | d_{t-1}^m\right)$ has four elements, two of which are constrained by normalisation. These parameters are design parameters which can be chosen either by subjective belief, or by optimising particular criteria.

In [LL01b] a set of heuristic rules is optimised in order to select parameters of the *perceivability* model. This model is visibility by another name (in the PDAF context of [CDA86]), and [LL01b] is the only work to address the issue of parameter selection. Other authors either present quantities with no explanation, or do not provide these numbers

due to propriety issues. The parameters derived in [LL01b] are

$$\begin{aligned} P\left(d_0^m = 1\right) &= 0.5, \\ P\left(d_t^m = 1 | d_{t-1}^m = 1\right) &= 0.988, \\ P\left(d_t^m = 1 | d_{t-1}^m = 0\right) &= 0.0. \end{aligned}$$

The method used for initiation performance analysis used in this thesis is the initiation Receiver Operating Characteristic curve (described in detail in section 6.5.1). So, it is sensible to optimise the parameters for the curve. This confirmed the above parameters, with the exception of the prior probability, which was instead chosen as

$$P\left(d_0^m = 1\right) = 0.1.$$

6.4.2 Initiation with PMHT-ye

When the PMHT-ye is used, the algorithm estimates the assignment state sequence, and this sequence itself can be used as the quality measure. As with the PMHT-ym, the batch information is summarised by taking the average of the temporal values.

It is not appropriate to use the binary visibility model used for the PMHT-v, because the estimated visibility sequence will be a realisation of the visibility Markov chain. If this chain is binary, then the estimated visibility sequence will essentially correspond to either 0% or 100% confidence in the candidate track. This does not allow the user to choose a desirable false track promotion probability, except perhaps by tuning the Markov chain transition probabilities. The algorithm is forced to make a hard decision. It is preferable to have an algorithm that provides for some uncertainty in the quality of candidates. For this reason, there should be more than two values in the state space, i.e. $M_D > 2$.

The functions $\phi_t^m\left(d_t^m\right)$ are chosen to be

$$\phi_t^m\left(d_t^m\right) = \frac{d_t^m}{n_t \quad (M_D - 1)}. \tag{6.17}$$

Thus, higher values of d_t^m correspond to models which contribute more measurements to the mixture and are hence more likely to correspond to valid targets. M_D is now equivalent to a quantisation level of $\frac{1}{M_D - 1}$ in the $\phi_t^m\left(d_t^m\right)$ used for the PMHT-v in the previous section.

Heuristic specifications on the transition matrix are provided by intuition: the matrix should be dominantly diagonal with the probability of switching between extreme ends of the assignment state space being very low. When the main diagonal of the transition matrix is strongly dominant, then the estimated state sequence will smoothly transition between values.

The transition matrix has $M_D(M_D - 1)$ free elements to optimise. To simplify the parameter optimisation, it was chosen to limit the freedom of the transition matrix by assuming the following form:

$$\Delta_0^m\left(d_0^m\right) = \frac{1}{M_D} \quad \forall m, \tag{6.18}$$

$$\Delta_t^m\left(d_t^m = j | d_{t-1}^m = i\right) = \bar{d}_i \exp\left[-\frac{(i-j)^2}{2\left(\bar{\sigma}_D M_D\right)^2}\right], \tag{6.19}$$

where $\bar{\sigma}_D$ is now the only design parameter, and this quantity controls the dominance of the diagonal elements. $\bar{d}_i$ is a normalising constant that ensures that all of the transitions

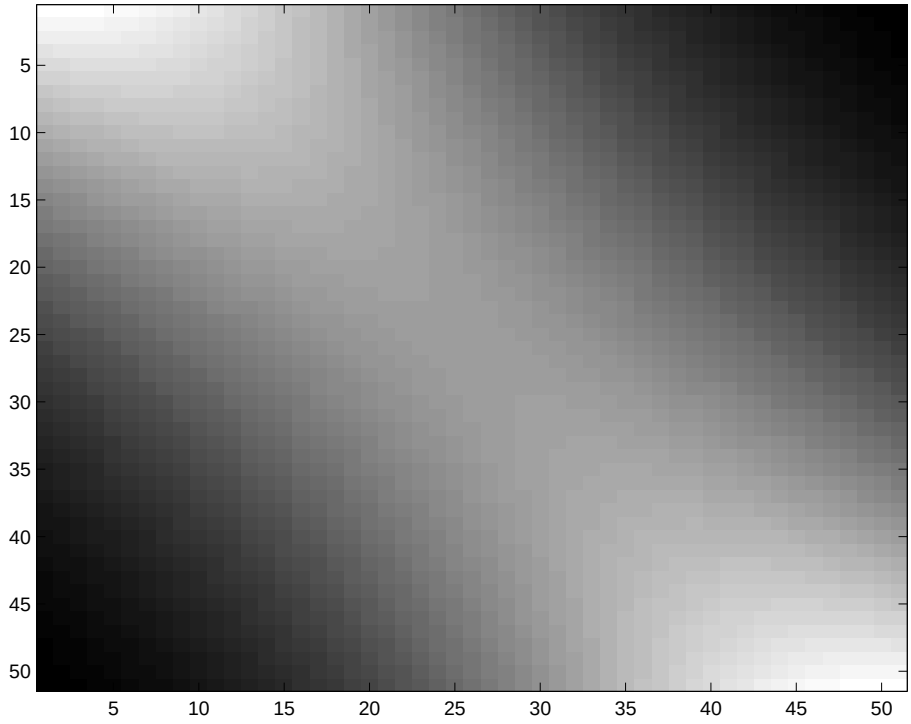


Figure 6.5: Transition matrix, $\Delta_t^m \left(d_t^m = j | d_{t-1}^m = i\right)$, for $M_D = 51$

from state i sum to unity probability and is given by

$$\bar{d}_i = \left\{ \sum_{j=1}^{M_D} \exp\left[-\frac{(i-j)^2}{2\left(\bar{\sigma}_D M_D\right)^2} \right] \right\}^{-1} = \left\{ \sum_{j=1-i}^{M_D-i} \exp\left[-\frac{j^2}{2\left(\bar{\sigma}_D M_D\right)^2} \right] \right\}^{-1}. \tag{6.20}$$

$\bar{d}_i$ is the inverse of the sum of the exponential function in (6.19) over a window that slides according to i. It will take its biggest value for $i = 1$ and $i = M_D$. This means that the diagonal terms in the transition matrix are biggest for the extreme visibility values. The matrix defined by (6.19) and (6.20) is shown in figure 6.5 for a particular value of $\bar{\sigma}_D$. The brightest areas show the greatest probabilities, and these areas are found near the top left and bottom right end of the matrix. This tends to encourage the model towards extreme values.

6.4.2.1 Parameter Selection for the PMHT-ye

The above hysteresis model for initiation with the PMHT-ye still has two parameters to select: the dimension of the assignment state space, M_D, and the dominance of the transition diagonal, controlled by $\bar{\sigma}_D$. These could be chosen by subjective belief, or preferably by optimisation of an objective criterion. As in section 6.4.1, the appropriate criterion to use is the initiation Receiver Operating Characteristic (ROC) curve, since the ROC will be used as the performance measure for track initiation. The ROC curve for track initiation is discussed in detail in section 6.5.1. The parameter M_D is chosen to be as small as possible, while maintaining high performance. The transition parameter, $\bar{\sigma}_D$ is chosen to give the best ROC curve.

The PMHT-ye is not included in the initiation analysis which is to follow. This is because it was found to give poor performance. When the algorithm was able to recover

from initialisation and follow the target trajectory, it easily discriminated the valid target track from false tracks due to clutter. However, on difficult realisations (usually when the target was not detected for several scans near the start of the batch) the PMHT-ye failed to recover from initialisation and diverged from the true trajectory. Essentially it performed excellently on the simple cases, but very poorly on the more difficult cases. For this reason, it is not considered further.

6.5 Performance Analysis Method

Several methods for track initiation with PMHT have now been presented. These methods will be compared in this chapter through simulation. However, the measures used in previous chapters (such as the mean squared estimation accuracy) are not appropriate for track initiation. Instead, the performance measure for track initiation should quantify how well the candidate test discriminates between valid and false candidate tracks. This will be done by using a Receiver Operating Characteristic curve.

6.5.1 The Receiver Operating Characteristic Curve

The track promotion test in the initiation process is a decision and can be viewed as a detector. The promotion test aims to detect valid target tracks and to reject false tracks that associate clutter measurements. One way of quantifying the performance of a detector is through the use of a Receiver Operating Characteristic curve. The ROC plots the probability of correctly detecting a desired signal as a function of the probability of incorrectly detecting a noise signal (a false alarm). Each point in the locus of the curve represents a particular detector setting (threshold). The ROC curve summarises the probability of detection and the probability of false alarm for all possible detector parameter choices and thus provides a measure of how well the test discriminates valid and false signals.

Figure 6.6 shows an example ROC curve for track initiation. Whereas the standard ROC curve plots probability of detection as a function of probability of false alarm, the tracker ROC plots the probability of promoting a valid track (a correct positive result) as a function of the probability of promoting a clutter track (a false positive). Here, a valid track is a track which has been initialised on a true target measurement(s) and is within tolerable estimation error of the true trajectory. A clutter track is one which is initialised on clutter and does not follow a true target trajectory. The ROC curves presented in this and following chapters, plot the probability of promoting a valid track on a linear scale and the probability of promoting a clutter track on a logarithmic scale. This is because the probability of promoting a clutter track is required to be very low. Commonly there are many more false detections than valid target detections and consequently the majority of the candidate tracks will be clutter tracks.

The ideal performance curve would sit in the top left corner of the plot. At this point, all valid tracks are promoted and all clutter tracks are discarded. At any particular probability of promoting a clutter track, a higher probability of promoting a valid track is desirable and so the curve which is higher is preferred. In figure 6.6, test A gives superior performance at high probability of promoting clutter tracks, but test B gives better performance for low probabilities. In such a case, neither test is universally preferred, and the design choice depends on the operating point at which the system will work.

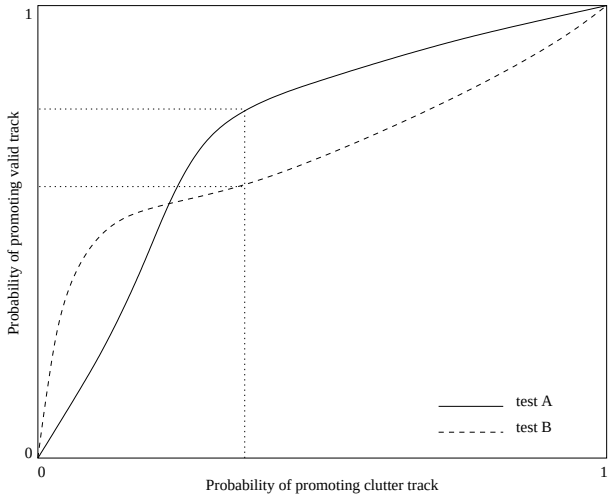


Figure 6.6: Example tracker ROC curve

Although the ROC curve has been used to quantify track initiation performance (e.g. [BSCL90, LHB96, BSF88]), it is not common. Rather, authors tend to choose to fix the false track rate (equivalent to choosing a particular probability of promoting a clutter track) and observe the valid track initiation performance. This may result in misleading conclusions. In figure 6.6 dotted lines show the performance obtained when a mediocre probability of promoting clutter tracks is specified. In this case, the curve corresponding to test A would give better results than the curve for test B, leading to the conclusion that test A is a superior test. Clearly this is not universally true, and in fact test B is preferable when very low probability of promoting clutter tracks is required (which it usually is). In order to show results representative of the algorithm under test and independent of the particular operating point chosen, the ROC curve is used as the primary ruler for measuring track initiation performance.

6.5.2 Other Performance Assessment Approaches

The ROC curve is a good method for assessing track initiation performance, but this is only one facet of the tracking algorithm. Aspects such as manoeuvre handling, estimation error, overshoot and false track duration are all important factors in overall tracker rating. In [CDD96], seventeen metrics for tracker performance are used to compare competing algorithms. This approach is more appropriate to gain an overall indication of tracking performance. However, he main focus of this thesis will be track initiation performance quantified via the ROC curve. Other studies have already considered the performance of PMHT under established track criterion, such as estimation accuracy (for example, [RWS95a, RWS99, RW01a, WRS98b]).

6.5.3 Generating ROC Curves From Simulated Data

In section 6.5.1 the track initiation ROC curve was introduced as a method for evaluating the performance of track initiation schemes. This curve is the locus of the probability of promoting a track, given it is associated with a valid target, as a function of the probability of promoting a track, given it is false. The problem with using the ROC curve is that these two promotion probabilities cannot be analytically derived for complicated algorithms

such as the PMHT and its variants. This means that the true ROC curve cannot be computed, and an approximated one must be used in its place. This approximated ROC curve is produced by estimating the promotion probabilities above. These estimates are obtained using Monte Carlo simulation.

The true ROC curve is composed of the promotion probabilities which are the cumulative distributions of the quality statistic

$$P(\text{promote}|\text{valid}; t) = \int_t^\infty p(q|\text{valid})\mathrm{d}q, \tag{6.21}$$

where t is the promotion threshold. Given a sample of N valid tracks, this probability can be approximated by

$$P(\text{promote}|\text{valid}; t) \approx \frac{1}{N}\sum_{n=1}^{N} I\left(q_n \geq t\right), \tag{6.22}$$

where $I(\cdot)$ is an indicator function taking the value unity when its argument is true and zero when it is false. This is equivalent to approximating the density $p(q|\text{valid})$ with a sum of dirac delta functions, each shifted to the location of one of the observations

$$p(q|\text{valid}) \approx \sum_{n=1}^{N} \delta\left(q - q_n\right). \tag{6.23}$$

Using the approximation in (6.22) may lead to a noisy estimated ROC curve. One way to smooth the estimated ROC is to use a smoother density estimate for the promotion probability. This may be done by using a kernel density approximation [Sil86]

$$p(q|\text{valid}) \approx \frac{1}{N}\sum_{n=1}^{N} \frac{1}{s}\eta\left(\frac{q - q_n}{s}\right), \tag{6.24}$$

where the kernel function, η, is a normalised, smooth function. If the support of the kernel function is fixed at (say) $[-1, 1]$, then the dilation parameter, s, controls the support of the contribution of each observation, and hence the smoothness of the pdf estimate.

Using this kernel estimator, the smoothed estimated cumulative density function is

$$P(\text{promote}|\text{valid}; t) \approx \frac{1}{N}\sum_{n=1}^{N} \int_t^\infty \frac{1}{s}\eta\left(\frac{q - q_n}{s}\right)\mathrm{d}q. \tag{6.25}$$

The ROC curves that are presented in this thesis have all been generated using a quadratic kernel function

$$\eta(q) = q^2 - 1, \tag{6.26}$$

with a dilation parameter, $s = 0.1$.

6.6 Simulated Track Initiation Performance

The performance of three of the PMHT track initiation approaches presented in this chapter is now examined through simulation. These approaches are

1. The standard PMHT algorithm using the sum of the candidate assignment weights as a promotion test statistic. This approach is referred to as the *weights sum* (see section 6.3.1).

2. The standard PMHT algorithm using the incremental improvement in the EM auxiliary function by introducing a candidate as a promotion test statistic. This approach is referred to as the *cost increment* (see section 6.3.2).

3. The PMHT-ym algorithm using visibility for the assignment state model. The candidate promotion test statistic is the average probability that the assignment state is that of a visible target. This approach is referred to as PMHT-v (see section 6.4.1).

A crucial factor in the production of the initiation ROC curves is a knowledge of the underlying truth. The statistics are conditioned on knowledge of whether each track is valid or false. In order to guarantee this knowledge, two test scenarios are used. To estimate the candidate statistics of false tracks, a scene containing no target is used. The statistics of valid tracks are estimated using a scene with a single target, and a valid target measurement is used to initialise the track. The valid candidate is discarded if the final state estimate deviates from the true target state by more than a prescribed amount. This approach avoids the possibility of coincident tracks, and false tracks being assigned valid target measurements.

The simulated target model is the two dimensional almost constant velocity model (see section 3.3 for more details on the almost constant velocity target model). This is a cartesian model, with target motion independent in the two coordinate axes.

The sensor provides a two dimensional measurement vector which contains an observation of the target position in each dimension, corrupted with independent noise. This is a linear Gaussian model. Different distributions of the clutter measurements are considered, although each consists of only a single model. Thus, $M_Y = 1$ and $\sigma_t^1 = 1$. The sensor detects measurements over a footprint arbitrarily labelled from -50 to 50 in the x direction and from 0 to 100 in the y direction. Measurements are collected over a batch of $T = 11$ scans.

When a target is present, it begins in the middle of the sensor footprint with an initial state given by $\boldsymbol{x}_0 = [0, 0.35, 50, 0.35]^{\mathsf{T}}$ (corresponding to a velocity vector of magnitude 0.5) . The process noise for the state evolution is set to $\mathsf{Q} = 0.001\mathbf{I}(2)$, where $\mathbf{I}(n)$ is the n dimensional identity matrix. This low process noise ensures that the target trajectory is approximately straight. When a target is present, it is detected with probability $Pd = 0.6$. Each target measurement is corrupted by Gaussian noise with covariance $\mathsf{R} = \mathbf{I}(2)$.

Tracks are initialised using a single measurement at zero velocity. This is equivalent to assuming that the mean of the initial target state distribution is

$$\bar{x}_0 = [z_{1r}[x], 0, z_{1r}[y], 0]^{\mathsf{T}}, \tag{6.27}$$

where $z_{1r}[x]$ and $z_{1r}[y]$ are the two components of the rth measurement at the first scan, i.e. $\boldsymbol{z}_{1r} \equiv [z_{1r}[x], z_{1r}[y]]^{\mathsf{T}}$.

The initial covariance assigned to the tracks (i.e. the assumed covariance of the initial state distribution) is

$$\mathsf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}. \tag{6.28}$$

The covariance of the position part of the initial state is the covariance of the measurement used to initialise it. The covariance of the velocity part of the initial state is chosen so that the true target velocity is several standard deviations from the initial estimate.

The track initialisation is deliberately chosen to be poor to demonstrate the robustness introduced via the innovation homothetic measurement model, and also to make it more difficult to discriminate between valid and false tracks. It is necessary to choose scenarios where discrimination is difficult because easier conditions will only show differences between the initiation approaches at extremely low false track promotion probabilities. Since the promotion probability is estimated by a smoothed ratio of simulated experiments, these low promotion probabilities can only be estimated by performing unrealistically numerous trials.

For the challenging scenarios chosen, 10000 Monte Carlo trials are used for each statistic to be tested. The cost increment statistic and the weights sum statistic are both derived from the standard PMHT algorithm. Thus, each plot in the following analysis is generated using 40000 random trials (10000 each of valid and false tracks for the standard PMHT and the PMHT-v).

6.6.1 Divergent Tracks

On some valid track trials, the state estimate will diverge from the true target trajectory. This may occur because the track is seduced by clutter, or it may be that the track is not able to recover from its poor initialisation (perhaps due to early missed detections). Such tracks are no longer valid. The candidate track is effectively false. Thus, they are not included in the analysis. To test for this condition, the distance between the track and the true target position is measured through the batch. If this distance exceeds a particular threshold, then the track is deemed divergent, and not included. This threshold is chosen to be 3 standard deviations of the measurement noise. This means a track is divergent if

$$\max_{t=1}^{T} (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t)^{\mathsf{T}} \mathsf{H}^{\mathsf{T}} \mathsf{R}^{-1} \mathsf{H} (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) > 9. \tag{6.29}$$

The number of divergent tracks is used independently as a measure of initialisation robustness, and is presented after the ROC curve analysis.

6.6.2 Uniform Clutter Distribution

The first case considered is the ubiquitous uniform clutter distribution. The performance of the three proposed initiation schemes is examined for two different rates of false detections. The first of these corresponds to 10 clutter measurements per scan. This is quite a low concentration of false measurements, and all of the schemes have a fairly easy time in discriminating between false and valid tracks. In this example there is little to distinguish the different approaches. The estimated ROC curve for each of the initiation schemes on this relatively low false detection rate is shown in figure 6.7. The weights sum and cost increment quality measures are shown as a dashed line and a dotted line respectively. The PMHT-v performance is shown as a solid line.

The second clutter false detection rate corresponds to 50 clutter measurements per scan. The estimated ROC curves for this relatively high concentration clutter are shown in figure 6.8. As is expected, the performance of all approaches is degraded from that obtained on the lower concentration clutter. Of the three approaches, the PMHT-v gives the best performance, although the difference between it and the weights sum approach is not significant. The cost increment approach gives significantly worse performance than the other two. This is a little surprising, since the cost increment approach is linked to established Model Order Estimation techniques, whereas the weights sum approach is

purely an intuitive, ad hoc method. This performance is expected, however, if the nature of the two statistics is more closely examined.

6.6.2.1 Relationship Between Cost Increment and Weights Sum for a Uniform Clutter Distribution

It may have been intuitively expected that the weights sum (being an ad hoc test) would give the worst initiation results. However, the cost function increment has worse performance. By manipulating the expression for the cost increment quality statistic, this result can be predicted.

In this experiment, there is only one target model, and one clutter model. If the target model is removed, then the cost function is

$$\begin{aligned} Q(\mathbf{X}) &= \sum_{t=1}^{T}\sum_{r=1}^{n_t} \log \zeta_t^1 \left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^1\right) \\ &= \sum_{t=1}^{T} n_t \log \zeta_t^1, \end{aligned} \tag{6.30}$$

since all of the measurements have the same probability when the clutter pdf is uniform.

With the target model, the cost function is

$$Q\left(\mathbf{X}, \mathbf{X}^c\right) = Q_X^2 + \sum_{t=1}^{T} Q_{t\pi} + \sum_{t=1}^{T}\sum_{r=1}^{n_t} w_{1tr} \log \zeta_t^1. \tag{6.31}$$

Thus

$$\begin{aligned} q_Q^c &= Q\left(\mathbf{X}, \mathbf{X}^c\right) - Q(\mathbf{X}) \\ &= Q_X^2 + \sum_{t=1}^{T} Q_{t\pi} + \sum_{t=1}^{T}\sum_{r=1}^{n_t} w_{1tr} \log \zeta_t^1 - \sum_{t=1}^{T} n_t \log \zeta_t^1 \\ &= Q_X^2 + \sum_{t=1}^{T} Q_{t\pi} + \sum_{t=1}^{T}\sum_{r=1}^{n_t} (1 - w_{2tr}) \log \zeta_t^1 - \sum_{t=1}^{T} n_t \log \zeta_t^1 \\ &= Q_X^2 + \sum_{t=1}^{T} Q_{t\pi} - \sum_{t=1}^{T}\sum_{r=1}^{T} w_{2tr} \log \zeta_t^1 \\ &= Q_X^2 + \sum_{t=1}^{T} Q_{t\pi} - \log \zeta_t^1 q_w^c, \end{aligned} \tag{6.32}$$

where Q_X^2 is the model cost for the candidate track.

The statistic clearly consists of a scaled version of the benchmark, q_w^c, plus two other terms. The $Q_{t\pi}$ term is relatively small and has secondary effect. The candidate term Q_X^2 is negative definite since it consists of the sum of the norms of the random errors in the model scaled by their corresponding covariances and by a factor of $-\frac{1}{2}$. The assignment weights also scale the measurement orientated error terms. The term proportional to q_w^c is positive definite since $\zeta_1 < 1$. Thus the Q_X^2 acts against the q_w^c term. When the track corresponds to a valid target, it is likely that many measurements will be assigned to it, and this makes the Q_X^2 term larger.

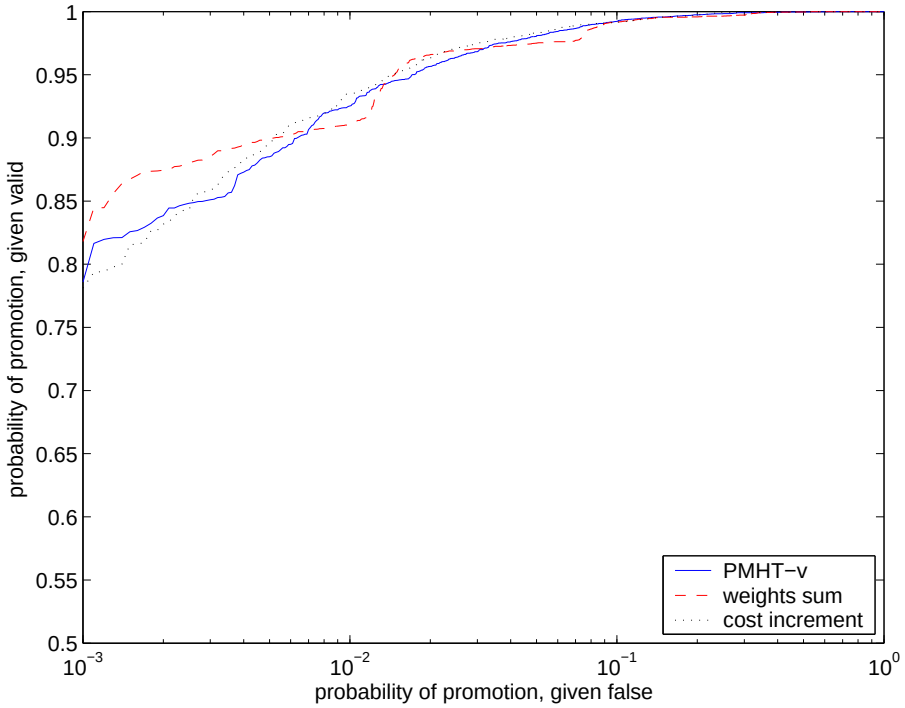


Figure 6.7: Uniform Clutter, low rate of false detections

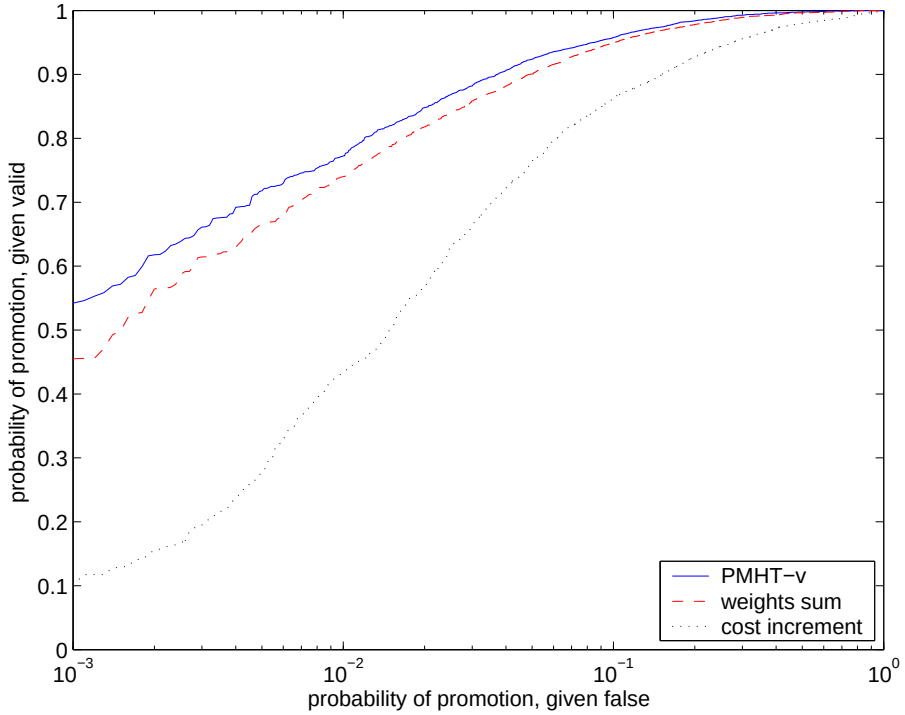


Figure 6.8: Uniform Clutter, high rate of false detections

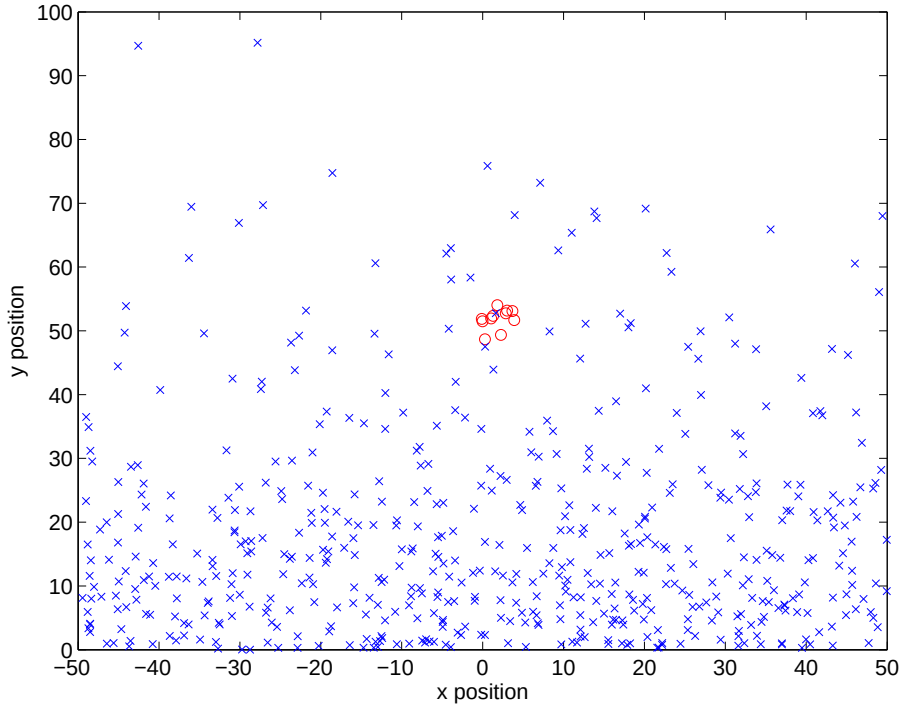


Figure 6.9: Example trial with target for chi-squared clutter

The cost increment statistic can be decomposed into two terms: the first term is the reduction in the clutter measurement cost due to assigning measurements to the candidate track. The second term penalises this cost improvement based on the discrepancy between the candidate track and the assumed model. When the clutter is uniformly distributed, then the first term is simply proportional to the sum of the candidate assignment weights. However, when a non-uniform clutter distribution is present, the first term acts to reduce the quality statistic of tracks in highly cluttered regions and increase the quality statistic of those in sparsely cluttered regions. Thus the cost increment approach is more suited to non-uniform clutter distributions.

6.6.3 Chi-Squared Clutter Distribution

The next clutter distribution considered is the chi-squared distribution with two degrees of freedom, i.e. an exponential distribution. This distribution is used in the y coordinate direction, and a uniform distribution is used in the x coordinate direction. The clutter distribution is thus given by

$$\zeta_t^1\left(\boldsymbol{z}_{tr}\right) = \frac{1}{2000}\exp\left\{-\frac{z_{tr}[y]}{20}\right\}. \tag{6.33}$$

As with the uniform distribution, two different rates of false detections are considered. Figure 6.9 shows an example trial. All of the measurements for the batch are shown in a spatial plot. The non-uniform distribution of clutter measurements in the vertical axis is clearly seen. The target is present in the trial, and the target detections can be seen in the middle of the plot. The target measurements are shown as circles, and false detections as crosses.

Figures 6.10 and 6.11 show the ROC curves generated for chi-squared distributed clutter. Again, the PMHT-v shows better performance than the weights sum. In this

case, the performance difference is more significant than in the uniform clutter example. In contrast to the uniform case, the cost increment approach now performs almost as well as the PMHT-v.

The weights sum approach performs particularly poorly for this example. Also, the ROC curve exhibits a staircase appearance. The staircase effect occurs because the PMHT weights tend to converge to extreme values: close to unity, or close to zero. This means that the weights sum statistic tends to have a pdf with peaks around integer values, for both valid and false tracks. The false track peaks are more broad because the false measurements have a higher scatter. Each point on the ROC curve corresponds to a possible promotion threshold setting. As this threshold increases, the resulting promotion probabilities follow the ROC curve from the top right corner (zero threshold where all tracks are promoted) to the bottom left corner (an extremely high threshold where all tracks are denied promotion). When this threshold changes from slightly less than a particular integer to slightly more, then a large number of tracks are no longer promoted, since there are peaks in the pdf at integer values. When many valid tracks are thus suppressed, this causes an almost vertical drop in the ROC curve. Conversely, when the threshold is varied between integer values, very few valid tracks are suppressed, but false tracks may be rejected because the false track peaks have more spread. This causes an almost horizontal segment in the curve. The result is a staircase appearance which is somewhat smoothed by the kernel pdf estimator (see section 6.5.3).

The overall poor performance of the weights sum occurs because the approach only considers the number of measurements which are assigned to a track, and does not take account for how well these measurements may have been described by the existing model without the new candidate. It is a track orientated approach, not a total system one. In contrast, the cost increment statistic is the increase in the log likelihood by the addition of the candidate. Thus if the candidate assigns measurements in a high clutter density region (in this case low values of $z_{tr}[y]$) the statistic will be small: those measurements already have a high likelihood under the hypothesis that they are due to clutter. If the candidate assigns measurements in a lower density region, then the statistic will be higher: those measurements had a low likelihood under the clutter hypothesis. Notice that the true target trajectory lies through a lower density part of the clutter distribution. If it were in the highest density part, the tracker would have no hope of following the target. Similarly, the PMHT-v uses a HMM smoother to estimate the probability of visibility. This smoother is driven by a gain term which is the conditional measurement likelihood under each visibility hypothesis, $P\left(\mathbf{Z}_t|\boldsymbol{D}_t, \mathbf{X}_t\right)$ (see (5.25) and (5.26)). When there is only one target, it can be seen that the HMM smoother is driven by the conditional likelihood ratio of the measurements with and without the candidate. If the candidate makes little change to the measurement likelihood (for reasons described above) then this ratio will be close to unity. If the likelihood is greatly increased, then the ratio is large and this drives the HMM smoother to give a high probability of the candidate track corresponding to a visible target.

The cost increment and PMHT-v approaches perform better than the weights sum because they both take account of the likelihood that the measurements assigned to the candidate track were due to clutter, whereas the weights sum effectively counts the number of measurements close to the candidate trajectory. The cost increment and the PMHT-v should thus be expected to always outperform the weights sum in non-uniform clutter densities. The PMHT-v gives better performance than the cost increment, because it also exploits the temporal history of the track. Candidates which assign measurements in many scans are given higher credence than those who assign a few outlier measurements.

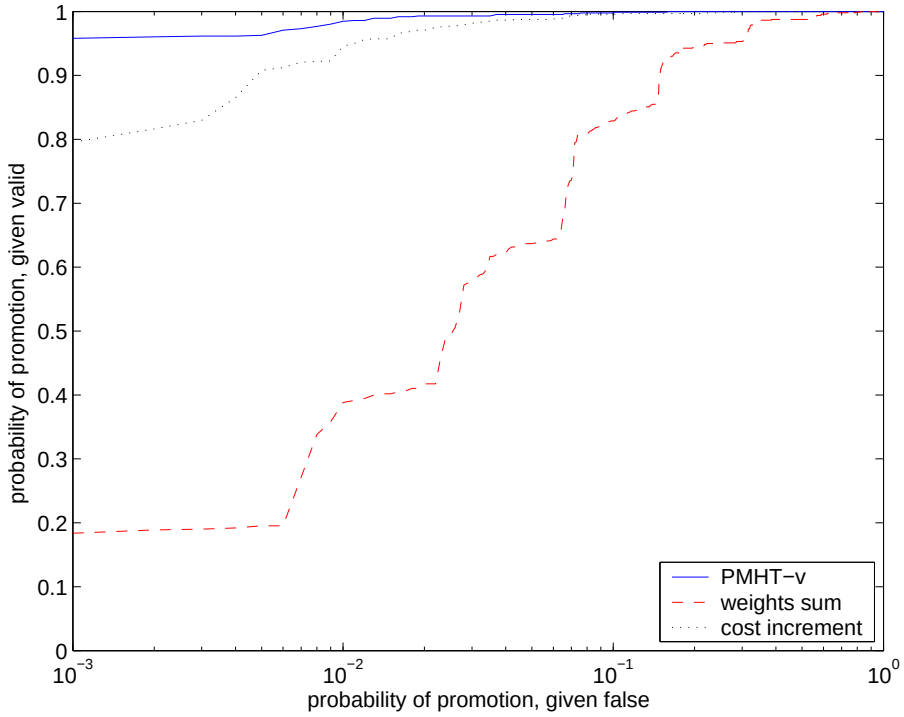


Figure 6.10: Chi-Squared Clutter, low rate of false detections

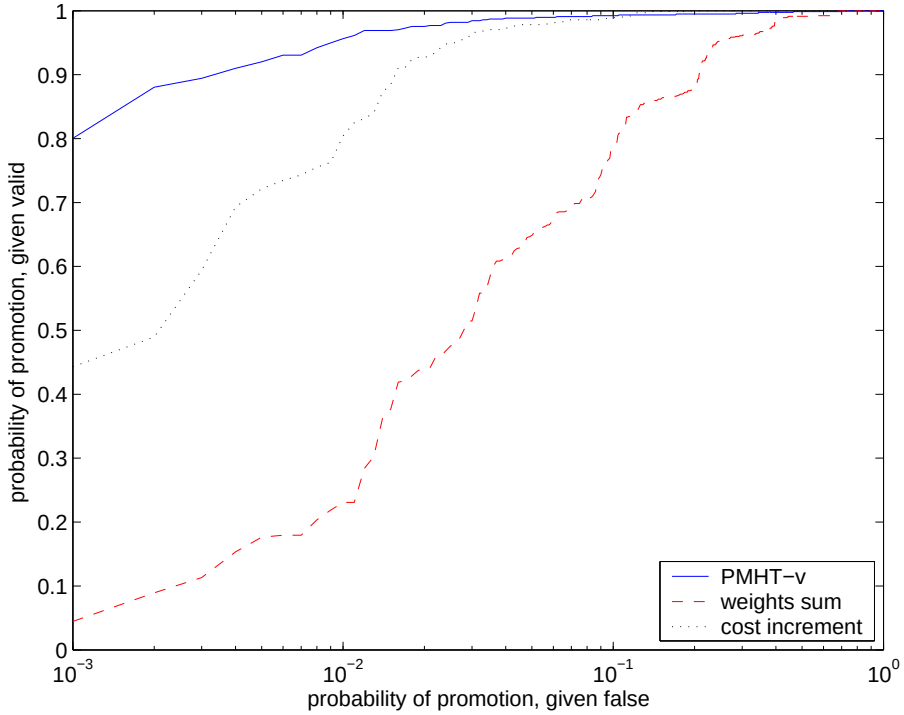


Figure 6.11: Chi-Squared Clutter, high rate of false detections

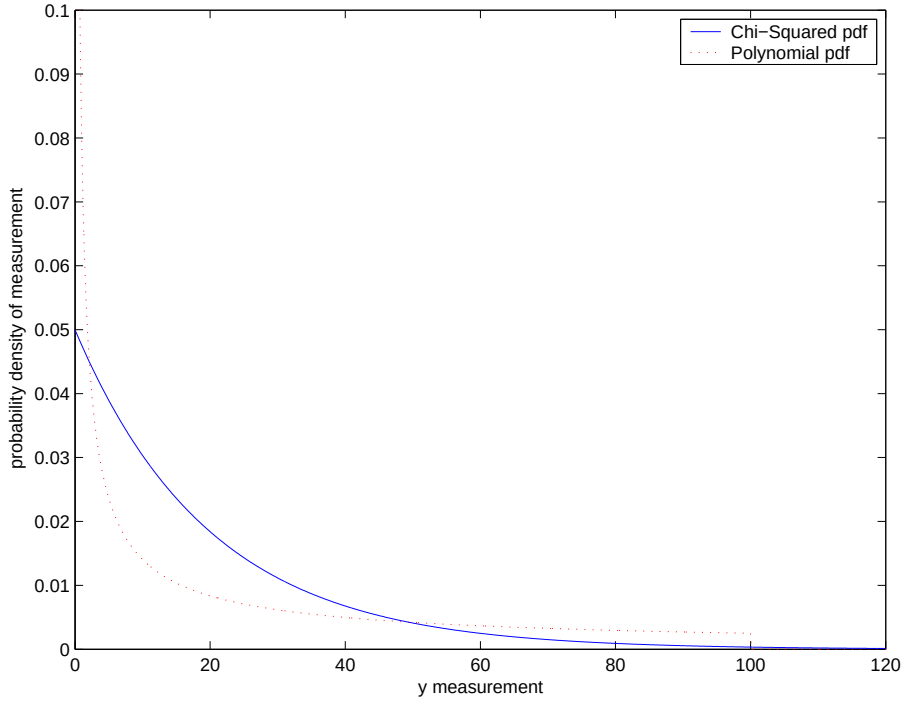


Figure 6.12: Non-uniform clutter densities

6.6.4 Polynomial Clutter Distribution

The chi-squared distribution in the previous section has a gradual decay. It also has an infinite support: it is possible (although unlikely) to observe measurements with arbitrarily high values. In this section, a polynomial distribution is used, which has a sharper peak at low values and a plateau at higher values. It also has compact support (the pdf is nonzero only within a closed interval) . This distribution is given by

$$\zeta_t^1(\boldsymbol{z}_{tr}) = \begin{cases} \frac{\sqrt{10}}{4000} \; z_{tr}[y]^{-0.75} & 0 < z_{tr}[y] < 100 \\ 0 & \text{otherwise.} \end{cases} \tag{6.34}$$

This distribution is long tailed, and also shows a large difference between the high and low density parts. It is generated using the simple relation

$$z_{tr}[y] = p^4, \tag{6.35}$$

where p is uniformly distributed on the interval $0 \ldots 1$.

Figure 6.12 demonstrates the difference between this distribution and the chi-squared distribution. The chi-squared (exponential) distribution is shown as a solid line and the polynomial distribution as a dashed line. The polynomial pdf can be seen to have a much higher peak at very low measurement values, but then the pdf drops quickly below the chi-squared pdf. For most of the measurement space, the polynomial pdf is almost constant, and high value measurements are more likely under this pdf than the chi-squared (up to the limit of the polynomial pdf support). This pdf is somewhat like a very low mean exponential pdf with a uniform pedestal.

The ROC curves for this clutter pdf with low and high false detection rates are shown in figures 6.13 and 6.14. As with the chi-squared case the weights sum gives the worst performance, although the difference is not as great. The weights sum also shows the

clutter type	uniform		chi-squared		polynomial	
false alarm rate	low	high	low	high	low	high
standard PMHT	597	3120	259	500	303	874
PMHT-v	619	2346	292	563	340	901

Table 6.1: Number of divergent trials

staircase appearance, particularly for low false detection rates. The cost increment gives somewhat better performance than the PMHT-v on the low false detection rate data, but they give approximately the same performance on the higher false detection rate data. This may be because the very high peak in the clutter pdf makes it likely that a false candidate will find a sequence of false detections at very low $z_{tr}[y]$ values. The PMHT-v would then enhance the statistic of such tracks because of the temporal history.

6.6.5 Initialisation Robustness

The empirical ROC curves are generated using only those valid tracks which remain within an error tolerance of the true target state. This ensures that tracks which have diverged from the correct trajectory do not adversely affect the initiation metric. However, the number of these divergent tracks is also of interest. If the initiation algorithm is able to perfectly discriminate valid tracks from clutter tracks, but diverges from the true trajectory most of the time, then it is much less effective than the ROC will suggest.

The number of divergent trials is therefore listed here to demonstrate that the different schemes have similar performance in this area, and that each only diverges in an acceptably small number of trials. Since the cost increment and weights sum approaches both use information provided by the standard PMHT, they have the same number of divergent trials.

Table 6.1 lists the number of divergent trials for each of the scenarios.

The uniformly distributed clutter has the highest number of divergent tracks. This is because the non-uniform clutter densities concentrate the false detections away from the simulated target. The higher density clutter in this region makes discrimination between valid and false tracks more difficult because it produces false tracks with more matching measurements. However, the total number of measurements is the same, so the density around the target is lower for the non-uniform clutter. Since most divergent tracks occur when the track is seduced by clutter detections, this lower density around the target leads to a lower rate of divergence.

In general, the rate of divergent tracks is approximately the same for the PMHT and the PMHT-v, with the PMHT-v showing a marginally higher rate. This happens because the parameter choice for the PMHT-v was to use a very low prior probability that the target is visible. In section 6.4.1, the prior probability of a target being visible was chosen to be $\Delta_0\left(d_0^m = 1\right) = 0.1$. This means that the PMHT-v may decide that some difficult trials, where early measurements are missed, do not contain a valid track. When the visibility probability is very low, then the track is not updated by measurements and the track is marked as divergent because it never recovers from initialisation. This effect can be reduced by choosing a higher initial probability, but that degrades the ROC performance. Since the increase in the rate of divergent tracks is only marginal, it was preferred to tolerate it, in preference for superior ROC performance.

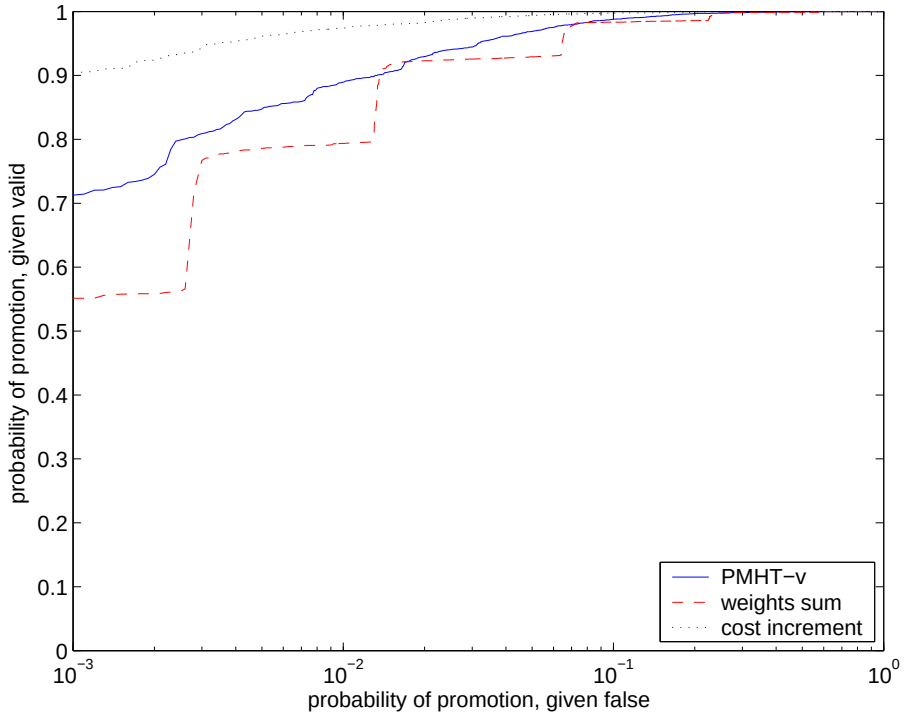


Figure 6.13: Polynomial Clutter, low rate of false detections

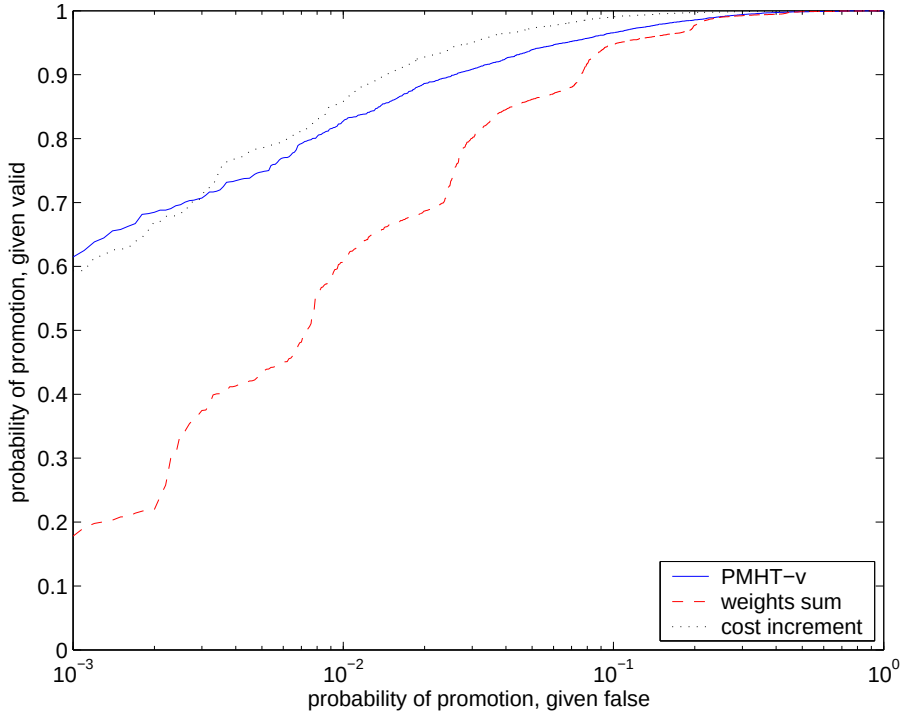


Figure 6.14: Polynomial Clutter, high rate of false detections

The exception to the above trend is the high false detection rate uniform clutter, where the PMHT-v performs much better than the PMHT. This is the only case where the two show a significant difference in performance. In this case, the PMHT-v shows better performance because it constrains the allowable values of π_t^m. If more than one measurement is close to the candidate track in one scan, then the standard PMHT will give an estimated $\hat{\pi}_t^m > \frac{1}{n_t}$. The PMHT-v limits the prior to $\frac{Pd}{n_t}$. Thus the converged assignment weights for the standard PMHT will be higher than those for the PMHT-v in such a case. Since the true measurement model used for the simulation can only produce one valid measurement per scan, the multiple measurement case only occurs due to false detections. These false detections can seduce the track away from the true path, and by limiting the value of π_t^m, the PMHT-v is made somewhat more resilient to this factor. This behaviour is not observed for any of the other data conditions because the non-uniform clutter pdfs concentrate the false detections in a different area to where the target is present. So, the high rate uniform clutter pdf is the case where the valid track is faced with the highest rate of false detections in its immediate vicinity.

6.6.6 Simulation Conclusions

Over the various clutter conditions simulated, the PMHT-v gave the best, or equal best performance on all but one example. In that case, it was slightly worse that the best performance. In non-uniformly distributed clutter, the weights sum approach did particularly poorly with performance becoming more degraded the less uniform the clutter pdf. The cost increment approach performed poorly in uniform clutter, and slightly worse than the PMHT-v in non-uniform clutter.

The PMHT-v algorithm, using the parameters selected to optimise ROC performance, was slightly less robust than the standard PMHT to poor initialisation when the target was undetected on several scans. However, it was better equipped to resist seduction in the presence of false detections.

Overall, the PMHT-v, which is a special case of the PMHT-ym algorithm, is found to be the preferred approach for track initiation out of those considered in this chapter.

6.7 Summary

The standard PMHT is limited by the assumption that the number of targets is fixed and known, and that prior information is available about the state of each. This chapter has introduced a method for making the PMHT robust to poor initialisation, due to inadequate knowledge of this prior. This method uses the homothetic measurement model, but employs a dynamic secondary measurement model specified by the innovation covariance matrix. This approach can be viewed as an automated covariance inflation method for initialisation.

This chapter has also presented alternative methods for allowing the PMHT to use a dynamic number of target models. These methods are based on over modelling the problem, and then rejecting superfluous models based on a model significance test. This approach is referred to as candidate based track initiation, and each candidate track is assigned a quality statistic by the tracker. Two different tests were propose for use with the standard PMHT. Firstly the candidate quality was quantified by using the sum of the assignment weights for that model. Secondly the candidate quality statistic was given by the improvement in the EM auxiliary function when it was introduced. The PMHT

with Hysteresis was also used to provide a candidate quality measure. The PMHT-ym was implemented using a binary assignment model, and the probability of the assignment state was used as the candidate quality. This algorithm is referred to as the PMHT-v because the binary assignment model is analogous to the visibility approach used for track initiation with other algorithms. The PMHT-ye was also used, but not tested extensively because it provided poor performance, even with a large state space.

The track initiation schemes proposed were tested on simulated data under a variety of clutter conditions, and their performance quantified by ROC curves for the candidate test. The PMHT-v (i.e. the binary PMHT-ym) method was found to consistently give good performance over all conditions and is considered the preferred approach based on these results.

Chapter 7

Applying the Probabilistic Multi-Hypothesis Tracker to Over The Horizon Radar

EARLIER chapters have introduced theoretical extensions of the PMHT algorithm. The performance of these extensions has been demonstrated through simulations. However, as with all things, *the proof of the pudding is in the eating* [dC05]. In this chapter, the PMHT extensions are applied to data recorded from the Jindalee Facility at Alice Springs (JFAS) Over the Horizon Radar (OTHR). The performance results obtained from recorded OTHR data are used to verify the simulated results of the previous chapters.

This chapter first reviews some basic principles of Over the Horizon Radar, with focus on some particular features of the JFAS radar pertinent to target tracking. Problems encountered when processing JFAS data are discussed and the PMHT solution to these problems presented. ROC curves for track initiation are then estimated using recorded JFAS data.

The JFAS OTHR is an Australian operational sensor. For this reason, the radar data itself, and the performance of the radar algorithms cannot be presented in an open publication. Instead, relative performance metrics comparing competing algorithms are used. Thus the performance measures, derived from radar data, presented in this chapter (in particular initiation ROC curves) deliberately do not show quantitative axis scales.

7.1 Over the Horizon Radar Fundamentals

Over the Horizon Radar is the name given to a class of active radar sensors that exploit electromagnetic refraction in the high frequency (HF) band. This refraction is used to curve the signal propagation path and sense beyond the line of sight limitation of a standard radar sensor. Figure 7.1 shows how a curved propagation path can increase the detection range of a radar.

Surface Wave radar is a form of OTHR that exploits refraction due to the sea. In Surface Wave radar, the radar signal hugs the sea surface because the media boundary between air and water curves the propagation path.

Skywave OTHR exploits a part of the atmosphere called the ionosphere. The ionosphere is a cloud of ionised particles formed by the solar radiation. It is composed of several regions of high ion density, referred to as layers. The elevation of these layers varies as does their density and spatial extents. Each layer provides a potential propaga-

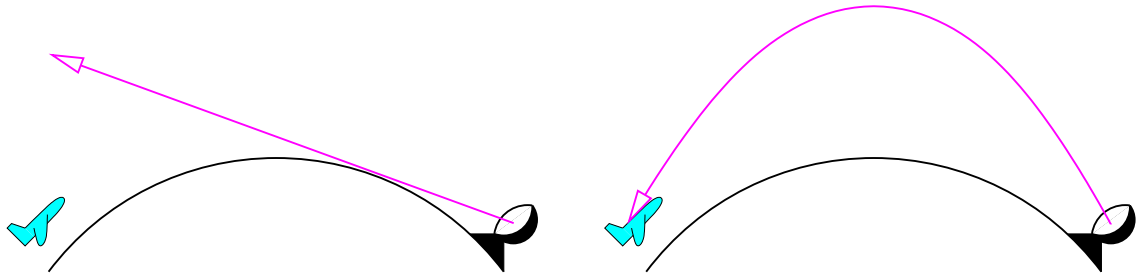


Figure 7.1: Extended radar range through electromagnetic refraction

tion path between the sensor and targets of interest. Since target echoes may return by a different layer from the one that transported the transmitted signal, skywave OTHR is a multipath sensing environment with the number of possible paths given by the square of the number of layers. There are other factors which also provide further propagation paths. For example, when an aircraft flies over water, a Lloyd's mirror effect is experienced, and a signal may propagate not only due to direct backscatter from the aircraft, but also from backscatter reflected from the water beneath. In this case, the length of the two paths will differ only slightly, however the phases of the two paths will be different, and may cause fluctuations in the received signal power.

A common condition for the JFAS radar is propagation support from two layers, called the E and F layers. Signals propagated via the E layer arrive more quickly than those propagated via the F layer, because of its lower altitude. Figure 7.2 shows how a two layer environment provides four return paths to the radar. In standard OTHR parlance, each of the different return propagation paths is referred to as a mode, and the modes are labelled to show the layer which supported the transmitted and reflected signals. The paths for which the return propagation is via a different path to the transmitted signal are called mixed modes. Mixed modes which involve the same two layers in reverse order (such as the FE mode and EF mode in figure 7.2) have the same length, but arrive at the receiver from different elevations. Since JFAS uses a linear receiver array, this causes the paths to be observed at different azimuths. The observed azimuth for a linear array is the cone angle between the array and the incident signal, which is a function of the bearing and elevation angles.

A result of sensing targets at long range is that there is very high propagation loss when backscattered signals eventually return to the receiver. In order to detect these signals, the transmitter power must be relatively high, and this requires a high duty cycle waveform. This restriction means that OTHR systems are built in a bistatic configuration with transmitter and receiver beyond line of sight to prevent interference. Since the separation of the transmitter and receiver is small compared with the observation range, the system can be assumed to be approximately monostatic.

The transmitter waveform is a frequency modulated signal with a carrier frequency ranging from a few megahertz to tens of megahertz. Lower frequencies are refracted more and are used to illuminate shorter ranges. Lower frequencies must also be used at night time when the ionosphere reduces in size without ionising radiation from sunlight. The radar uses a stepped scanning strategy, which means that wide area coverage is achieved by tiling the surveillance area with smaller regions. The sensor dwells on each region for a period, collecting data over a number of waveform repetitions to measure target range, bearing angle and Doppler frequency shift. The sensor then steps to a different region

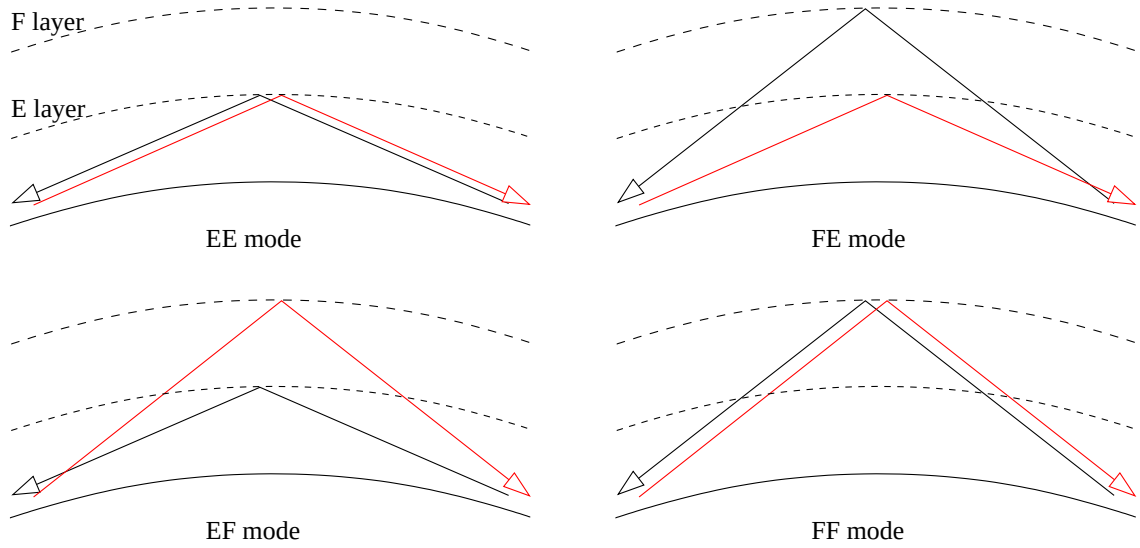


Figure 7.2: Typical multipath propagation

and later revisits each previous region based on a scheduling scheme. Regions may have different priorities and the revisit time (namely the time between consecutive looks for each region) is region dependent and variable.

7.1.1 Jindalee Facility at Alice Springs

The JFAS OTHR is a skywave radar built by the Defence Science and Technology Organisation (DSTO) to provide surveillance over the north western coast of Australia. The transmitter and receiver sites are located close to the city of Alice Springs in central Australia. This radar is operated by the Royal Australia Air Force (RAAF). Two similar radars at Laverton and Longreach have recently been commissioned. These sensors comprise the Jindalee OTHR Network (JORN), providing coverage to the entire northern and western coast of Australia.

Archived data from the JFAS OTHR will be used in this chapter to confirm the simulations presented in earlier sections.

A history of the development of the JFAS OTHR can be found in [Col00]. A technical overview of the signal processing algorithms used to process the radar data is given in [Lee87].

7.2 Specific Over the Horizon Radar Models

The particular target and clutter models used for this application are now presented. These models will be applied in all cases where experimental data is analysed.

7.2.1 Measurement Vector

The JFAS OTHR uses a sequential signal processing approach. The received signal from the antennae is passed through successive data processing algorithms culminating in the tracking algorithm. Figure 7.3 shows the typical data processing algorithms applied to the receiver data up to tracking. By the detection stage, the data has been converted

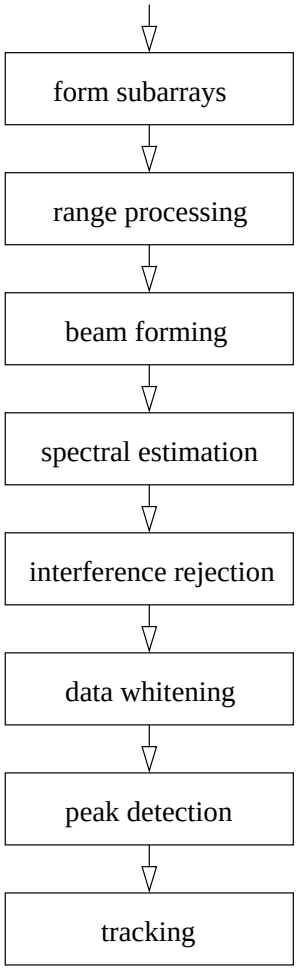


Figure 7.3: Typical OTHR data processing

into a map of the received amplitude in each azimuth, range, doppler bin. This map is referred to as the ARD. Local maxima are found in the ARD and these are interpolated using a 3 point quadratic fit in each of the ARD dimensions. This step is referred to as peak detection [DC99], or peak picking. The aim of peak detection is to produce a single measurement for distributed target returns which may have high amplitude output in many adjacent bins. The peak detector aims to localise the centroid of the target return with sub-bin accuracy.

In addition to detection, the peak detector also plays a part in clutter modelling. It segments the ARD into two areas, which define a classification surface, and are used to label each peak detection as either background or interference. This label identifies the data behaviour in the local vicinity of the bin where the detection is declared. If there is high amplitude in many adjacent bins, it is likely that this part of the ARD map is contaminated by an interference, and the interference label is given to the peak. Otherwise, the peak is labelled as background. These labels are used in the standard JFAS tracker for modelling the clutter probability density function [Col99]. Figure 7.4 shows an example of how the data whitener and peak detector operate on JFAS data. The figure shows firstly ARD data after doppler processing. The next image shows the data after the whitener, and the final image shows the peaks and the segmentation map. The peaks are shown as crosses and the grey shaded area in the third image shows the region designated as interference. In each image, there are five beams stacked on top of

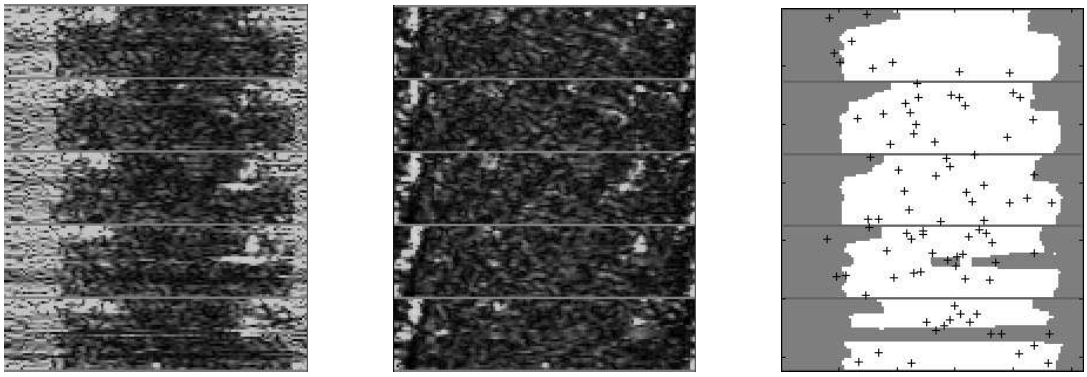


Figure 7.4: Data processing stages for the Jindalee OTHR

each other, and each beam is an image with range and doppler bins displayed vertically and horizontally respectively.

The output of the peak detector is the input to the tracker, and so there are two measurement types. Firstly, the peak detections themselves are four element vectors containing observed range, doppler, azimuth and amplitude. Thus,

$$\boldsymbol{z}_{tr}^{(x)} = [z_{tr}[r],\ z_{tr}[f_d],\ z_{tr}[\theta],\ z_{tr}[a]]^{\mathsf{T}}. \tag{7.1}$$

The second measurement is the discrete peak tags (interference or background). These classification tags can be viewed as a set of augmented measurements, giving

$$\boldsymbol{z}_{tr} = \begin{bmatrix} \boldsymbol{z}_{tr}^{(x)} \\ z_{tr}^{(k)} \end{bmatrix}, \tag{7.2}$$

where $z_{tr}^{(k)}$ is a binary variable denoting the class assigned to that peak.

7.2.2 Target State Representation

There are two different types of state models used to represent targets in the OTHR environment. These two types correspond to two different types of targets observed by the radar. The models are referred to as the *coupled model* and the *decoupled model* [Col99]. The difference between the two models is in the treatment of the observed Doppler frequency shift, $z_{tr}[f_d]$.

In the coupled model, the Doppler frequency shift is purely caused by the movement of the target. The shift is proportional to the radial velocity of the target, according to the standard Doppler shift equation

$$f_d = \frac{-2f_c\,\dot{r}}{c}, \tag{7.3}$$

where f_d is the doppler shift, f_c is the carrier frequency of the transmitted signal, $\dot{r}$ is the radial velocity of the scatterer, and c is the speed of light. The coupled model is used to represent standard air targets.

In the decoupled model, the observed Doppler shift is not purely due to the target motion, but also due to an additive frequency shift at the target, or in the propagation medium,

$$f_d = \frac{-2f_c\,\dot{r}}{c} + f_o, \tag{7.4}$$

where f_o is the additive frequency shift. The decoupled model has an extended state space which includes a frequency modulation parameter. The observed frequency shift is the sum of the Doppler frequency shift due to target motion and the frequency modulation shift. The decoupled model is used to represent transponders that use a mixer to modulate the transmitted signal. These transponders are usually geographically fixed and are used to infer ionospheric model parameters. However, they display apparent motion due to travelling ionospheric disturbances (waves in the ionosphere). These disturbances may cause variations in the elevation of arrival and the delay of the received signal, which is interpreted by sensor as motion in azimuth and range. Range variations are usually small, but azimuth effects can be significant. The decoupled model can also be used to model very low speed targets, such as ships, where the Doppler frequency shift due to movement of the ionosphere may be significant compared with the Doppler frequency shift due to target motion.

7.2.2.1 Coupled Model

The main state model is the coupled model and is used for aircraft and sea targets (when ionospheric motion compensation is applied before tracking). It consists of an almost constant velocity linear motion model in two dimensions: range and azimuth. The motion of most targets will not have constant velocity in the radar coordinate space, however it is a reasonable approximation to make at the long ranges encountered with OTHR. The process noise is assumed to be sufficient to compensate for this mismatch. The state vector also contains the target amplitude, which is modelled as an almost constant feature of the target. Thus, the coupled state vector for the mth target, $m = M_Y + 1 \dots M$, is

$$\boldsymbol{x}_t^m = \left\{ x_t^m[r], x_t^m[\dot{r}], x_t^m[\theta], x_t^m[\dot{\theta}], x_t^m[a] \right\}^{\mathsf{T}}. \tag{7.5}$$

The almost constant velocity model is presented in more detail in section 3.3. The state evolution is assumed to be a linear process contaminated with white additive Gaussian noise $\boldsymbol{u}_t^m$ with covariance matrix Q_t^m such that

$$\boldsymbol{x}_t^m = \mathsf{F}_t \boldsymbol{x}_{t-1}^m + \mathsf{G}_t \boldsymbol{u}_t^m. \tag{7.6}$$

The noise process $\boldsymbol{u}_t^m$ contains three elements modelling acceleration in range and azimuth and fluctuations in signal level. The covariance matrix, Q_t^m, is assumed to be diagonal. Thus,

$$\boldsymbol{u}_t^m = \{u_t^m[r], u_t^m[\theta], u_t^m[a]\}^{\mathsf{T}}, \tag{7.7}$$

and

$$\mathsf{Q}_t^m = \begin{bmatrix} Q_t^m[r] & 0 & 0 \\ 0 & Q_t^m[\theta] & 0 \\ 0 & 0 & Q_t^m[a] \end{bmatrix}, \tag{7.8}$$

where the scalars $Q_t^m[r]$, $Q_t^m[\theta]$, and $Q_t^m[a]$ are design parameters.

The matrices F_t and G_t are determined by the equations of motion and are given by

$$\mathsf{F}_t = \begin{bmatrix} \mathsf{F}(1\mathrm{D}) & 0 & 0 \\ 0 & \mathsf{F}(1\mathrm{D}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7.9}$$

$$\mathsf{G}_t = \begin{bmatrix} \mathsf{G}(1\mathrm{D}) & 0 & 0 \\ 0 & \mathsf{G}(1\mathrm{D}) & 0 \\ 0 & 0 & \tau_t - \tau_{t-1} \end{bmatrix}, \tag{7.10}$$

where $\mathsf{F}(1\mathrm{D}) = \begin{bmatrix} 1 & \tau_t - \tau_{t-1} \\ 0 & 1 \end{bmatrix}$, and $\mathsf{G}(1\mathrm{D}) = \begin{bmatrix} \frac{1}{2}(\tau_t - \tau_{t-1})^2 \\ \tau_t - \tau_{t-1} \end{bmatrix}$, as defined in section 3.3.

The measurement is assumed to be a linear process contaminated with additive white Gaussian noise $\boldsymbol{v}_{tr}$ with covariance matrix R_t such that

$$\boldsymbol{z}_{tr}^{(x)} = \mathsf{H}_t \boldsymbol{x}_t^m + \boldsymbol{v}_{tr}, \tag{7.11}$$

when $k_{tr} = m$ indicates that $\boldsymbol{z}_{tr}^{(x)}$ is due to target model m.

The noise process $\boldsymbol{v}_{tr}$ contains four elements corresponding to each of the four components of the measurement vector. The covariance matrix of this noise is assumed to be diagonal. Thus,

$$\boldsymbol{v}_{tr} = \{v_{tr}[r], v_{tr}[f_d], v_{tr}[\theta], v_{tr}[a]\}^{\mathsf{T}}, \tag{7.12}$$

$$\mathsf{R}_t = \begin{bmatrix} R_t^m[r] & 0 & 0 & 0 \\ 0 & R_t^m[f_d] & 0 & 0 \\ 0 & 0 & R_t^m[\theta] & 0 \\ 0 & 0 & 0 & R_t^m[a] \end{bmatrix}, \tag{7.13}$$

where the scalars $R_t^m[r]$, $R_t^m[f_d]$, $R_t^m[\theta]$, and $R_t^m[a]$ are design parameters.

The matrix H_t is given by

$$\mathsf{H}_t = \begin{bmatrix} 1 & \frac{2f_c}{cf_w}\Delta_r & 0 & 0 & 0 \\ 0 & -\frac{2f_c}{c} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7.14}$$

where f_w is the waveform repetition frequency, f_c is the carrier frequency, c is the speed of light, and Δ_r is the range extent of each range bin. The coupling between the range and doppler measurements occurs because both measurements are obtained from the phase law of the received signal. Notice that the azimuthal velocity is not observed and must be inferred from the progression of measurements over time. The doppler measurement is not strictly linear because the velocity measurement is ambiguous and some velocities are thus aliased. This is equivalent to a fixed offset which can be compensated. This aliasing is described in more detail in section 7.3.

7.2.2.2 Decoupled Model

The second target state model is the decoupled model and is used for transponders and an artificially injected calibration signal used in the JFAS radar. Transponders are positioned at various known locations within the coverage area of the radar and are used to estimate ionospheric model parameters. The transponder itself is a fixed device with a modulator that produces a frequency shift in the transmitted radar waveform. This frequency shift is observed as a doppler frequency shift by the radar processing. Thus the observed doppler shift is independent of the target dynamics. Since the physical location of the transponder is known, tracks formed on transponders can be used as reference points for ionospheric modelling, which is required to estimate the ground position of other radar targets.

The calibration signal is the other type of decoupled target. These signals are injected into the receiver subarrays at a fixed frequency modulation and in fixed range bins. Like transponder signals, they are observed by the radar processing as targets whose doppler shift is independent of their (stationary) dynamics.

As with the coupled model, the state vector contains amplitude. The model is identical to the coupled model, except that it also contains a frequency modulation parameter, $x[f_o]$. Thus the decoupled state vector is

$$\boldsymbol{x}_t^m = \left\{ x_t^m[r], x_t^m[\dot{r}], x_t^m[f_o], x_t^m[\theta], x_t^m[\dot{\theta}], x_t^m[a] \right\}^{\mathsf{T}}. \tag{7.15}$$

The state evolution is the same as the coupled model, with the frequency modulation assumed to be a constant process contaminated with white additive Gaussian noise. The noise process $\boldsymbol{u}_t^m$ contains four elements modelling fluctuations in range, azimuth, doppler frequency shift and signal level. These fluctuations may be caused by both random noise and ionospheric disturbances, such as travelling waves. The covariance matrix of this noise is assumed to be diagonal. Thus,

$$\boldsymbol{u}_t^m = \{u_t^m[r], u_t^m[f_o], u_t^m[\theta], u_t^m[a]\}^{\mathsf{T}}, \tag{7.16}$$

$$\mathsf{Q}_t^m = \begin{bmatrix} Q_t^m[r] & 0 & 0 & 0 \\ 0 & Q_t^m[f_o] & 0 & 0 \\ 0 & 0 & Q_t^m[\theta] & 0 \\ 0 & 0 & 0 & Q_t^m[a] \end{bmatrix}, \tag{7.17}$$

where again, $Q_t^m[r]$, $Q_t^m[f_d]$, $Q_t^m[\theta]$, and $Q_t^m[a]$ are scalar design parameters. The decoupled model uses lower values for the process noise parameters, because those targets which obey this model have sedate dynamics. In fact, any apparent dynamic behaviour is primarily due to ionospheric effects for targets of this type.

The matrices F_t and G_t are given by

$$\mathsf{F}_t = \begin{bmatrix} \mathsf{F}(1\mathrm{D}) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathsf{F}(1\mathrm{D}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7.18}$$

$$\mathsf{G}_t = \begin{bmatrix} \mathsf{G}(1\mathrm{D}) & 0 & 0 & 0 \\ 0 & \tau_t - \tau_{t-1} & 0 & 0 \\ 0 & 0 & \mathsf{G}(1\mathrm{D}) & 0 \\ 0 & 0 & 0 & \tau_t - \tau_{t-1} \end{bmatrix}. \tag{7.19}$$

The measurement model is the same as the coupled model, but with a measurement matrix given by

$$\mathsf{H}_t = \begin{bmatrix} 1 & -\frac{f_c}{cf_w}\Delta_r & 0 & 0 & 0 & 0 \\ 0 & \frac{f_c}{c} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7.20}$$

7.2.3 Clutter state model

The distribution of clutter measurements in azimuth, range and doppler frequency is modelled as uniform. A clutter state model is used to model the distribution of the amplitude of clutter measurements. The amplitude distribution for clutter measurements is modelled as a mixture of two components in the current tracking system. Two clutter models will therefore be used for the analysis of recorded OTHR data in this work. These two components can be thought of as false measurements caused by fluctuations in the

background noise and false measurements caused by interference. Common interference sources in the OTHR environment are meteors, range ambiguous spread doppler ground clutter and signals from other users of the HF band.

In the analysis to follow, the distribution of the amplitude of measurements from each clutter model is assumed to be exponential. As stated above, the distribution in other dimensions is assumed to be uniform. The clutter model state vector is a vector of the parameters of the assumed amplitude distribution which is a single parameter for the exponential distribution, i.e. $\boldsymbol{x}_t^m \equiv x_t^m$ is a scalar for the clutter models. The clutter state is assumed to be fixed over the batch. Thus the dynamics of the clutter model $m = 1 \dots M_Y = 2$ are modelled as

$$\psi_t^m \left(x_t^m | x_{t-1}^m \right) = \delta \left(x_t^m - x_{t-1}^m \right), \tag{7.21}$$

where $\delta(.)$ is the Kronecker delta function, an identity function taking unity value at the origin and zero elsewhere.

The probability density of clutter measurements is given by

$$\zeta_t^m \left(\boldsymbol{z}_{tr} | x_t^m \right) = \frac{1}{A} \frac{1}{x_t^m} \exp \left\{ - \frac{z_{tr}[a] - \nu_t}{x_t^m} \right\}, \tag{7.22}$$

where ν_t is the detection threshold which is assumed to be known, and A is the volume of the surveillance region in the dimensions of azimuth, range and doppler frequency.

7.3 Initialising from Ambiguous Doppler Measurements

As described earlier, the measurement received by an OTHR consists of an observation of range, azimuth, doppler frequency and amplitude. The waveform parameters used for OTHR are typically in the medium waveform repetition frequency range. This means that both the range and doppler frequency measurements may be aliased.

The ambiguous range of the waveform is usually very large - nominally a few thousand kilometres. This means that the signal attenuation from targets at this range will be so high that range ambiguous targets will not be detected. The exception to this is clutter that may be received from a double refraction through the ionosphere. This clutter will be detected from a long range that becomes aliased onto the region of interest because of the ambiguous waveform. The clutter is detected where targets are not because the backscatter from the sea surface is much greater than that of a target. Also, at longer ranges, returns very close to the sensor may also be aliased. This interference is referred to as *range folded clutter*. Typically, range folded clutter may also be spread through the doppler domain, making target detection problematic. Problems arising from range folded clutter and targets ambiguous in range can be resolved using a variable waveform repetition frequency, as discussed in more detail below. No special efforts will be made within the tracker to address ambiguous range measurements.

The ambiguous doppler of the waveform is a significant problem. Common waveform parameters result in a relatively low ambiguous doppler frequency. This means that the doppler frequency measurement will suffer from aliasing. Figure 7.5 shows observed the doppler shift as a function of target radial velocity and highlights the aliasing problem. The horizontal dotted line illustrates how several feasible target velocities might cause an observed normalised doppler frequency of 0.3. The normalised doppler frequency is

the actual frequency divided by the waveform repetition frequency, and the normalised radial velocity is the actual radial velocity divided by the ambiguous velocity, v_{amb} which is proportional to the ambiguous doppler frequency.

A result of the doppler aliasing is that the posterior velocity distribution is multimodal. Each possible unwrapping of the aliased doppler corresponds to a peak in the posterior velocity distribution. These unwrapped velocities (and hence the distribution peaks) are each separated by the ambiguous velocity. Thus, the posterior distribution is comb-like in structure with the spread around each mode determined by the measurement noise. Knowledge of the dynamic constraints of physical aircraft can be used to form a prior distribution of target velocities. Combining this prior with the observed doppler frequency and its measurement probability density gives the posterior probability density of the target velocity. Figure 7.6 illustrates an hypothetical posterior density given an observed doppler frequency of 0.3. Peaks are seen in figure 7.6 where the corresponding radial velocity wraps to the observation. The outer peaks are lower because they correspond to relatively high radial velocities that are considered unlikely. Further peaks might exist at higher velocities depending upon the ambiguous velocity of the waveform which defines the spacing between adjacent peaks. For OTHR these higher velocity wrappings usually correspond to infeasibly high target speeds.

For OTHR, the ambiguous doppler measurement is further complicated by the existence of *stationary targets*. Stationary targets are targets that do not physically move through space, but nevertheless produce an apparent doppler frequency shift, due to an additive frequency offset at the target. These are modelled using the decoupled model described in section 7.2.2.2. The existence of stationary targets provides a second family of unwrapped states - each with zero radial velocity, but varying aliased mixing frequencies. Just as physical constraints can be used to limit the number of unwrapped radial speeds considered, knowledge of the range of mixing frequencies used by transponders can be used to limit the number of stationary options considered.

A common technique to resolve the velocity unwrapping problem is to change the waveform repetition frequency from scan to scan. This changes the ambiguous range and doppler frequency, and hence the collection of unwrapped velocities. Only the correct velocity will be a possible unwrapping for all repetition frequencies. The correctly unwrapped velocity can then be determined by correlating the measurements over time. This approach eliminates different dynamic models and different stationary models, however it does not distinguish between a moving target model at the correct unwrapping and the stationary model at the equivalent frequency shift. The only way to discriminate between these is through the range history of the target measurements - i.e. whether the target actually moves. This is best done in the tracker itself. So, it is necessary to integrate a method for resolving the doppler ambiguity with the tracker.

The result of the ambiguity in the doppler measurement is that the initial state probability density function, $\psi_0^m(\boldsymbol{x}_0^m)$, is multimodal in the range rate and offset frequency dimensions. Let the index μ^m designate which of the possible unwrapped velocity options is the true target radial velocity. μ^m is an integer taking a value in the range $1 \ldots M_\mu$, where M_μ is the total number of modes of the initial state density. The density $\psi_0^m(x_0^m; \mu^m)$ is now unimodal, and the state prior is given by

$$\psi_0^m(\boldsymbol{x}_0^m) = \sum_{\mu^m=1}^{M_\mu} \psi_0^m(\boldsymbol{x}_0^m; \mu^m) \, {P_\mu}^m(\mu^m), \tag{7.23}$$

where ${P_\mu}^m(\mu^m)$ is the prior probability mass of the index, μ^m.

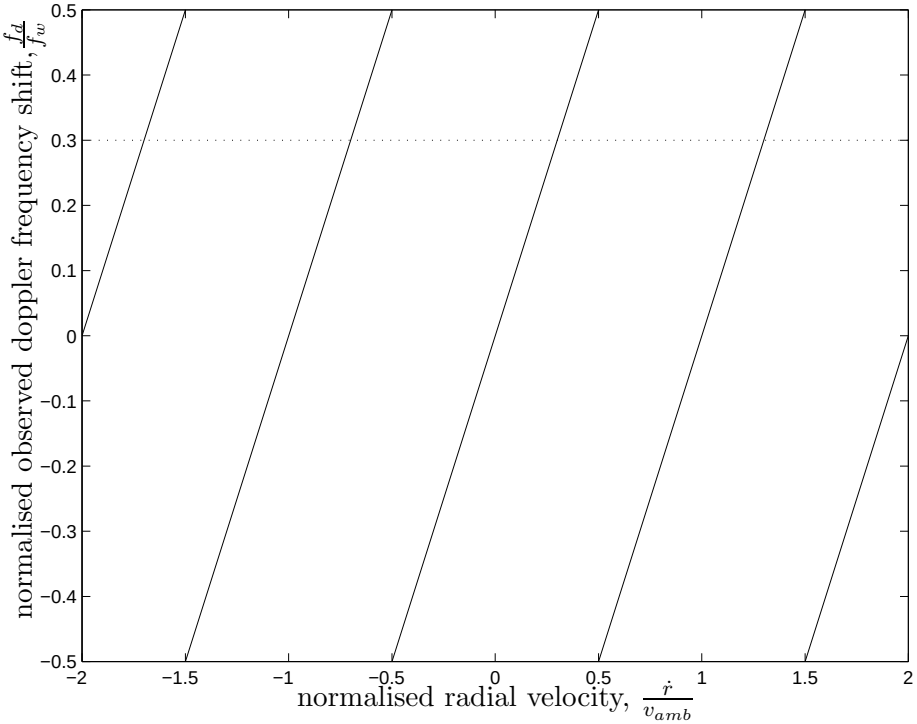


Figure 7.5: Observed Doppler frequency shift due to aliasing

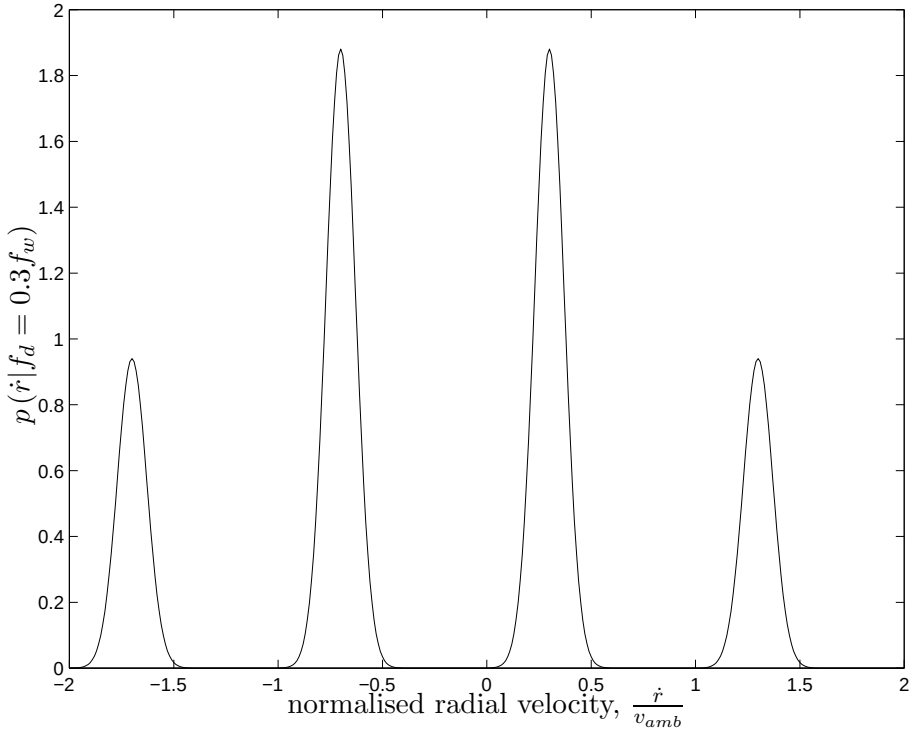


Figure 7.6: Posterior probability density of radial velocity

The design decision is now whether to treat μ^m as an extra piece of missing data (and thereby marginalise it out of the maximisation step) or to treat it as a variable to be estimated by the algorithm. There are two disadvantages of treating μ^m as missing data. Firstly, it will result in the need to calculate assignment weights for each different possible value of μ^m. This increases the algorithm's computation requirements since many more measurement density calculations must be made. These calculations are typically the most expensive part of the tracking algorithm. Secondly, the state estimate will be a superposition of the M_μ different possible states, weighted according to their probabilities. While each of these options is a viable state sequence, the superposition of many of them is most likely not. Further, the different dynamic options for the model must all occupy the same state space for the superposition to be sensible, and this is not the case since the decoupled and coupled models have different state vectors. So, the algorithm is likely to give a state estimate that is poor. This is the same problem that occurs with the MAP estimate of a discrete variable.

If μ^m is instead treated as a parameter to be estimated, then there is only one set of weights for each target model, and they are used to estimate μ^m. The estimated μ^m for the next iteration is that velocity that maximises the auxiliary function, namely the one with the lowest measurement scatter. However, these weights are determined using a particular value of μ^m derived from the previous iteration (or from initialisation). This means that the measurements that receive high weights might not be the true target measurements, but rather false detections that happen to lie along the wrongly assumed trajectory. These measurements reinforce the incorrect initial assumption. So, the estimation approach suffers from a bias towards whichever unwrapped velocity is initially assigned to the model. This problem may be exacerbated in dense target or multi-path conditions where it is more likely that measurements will be found along incorrect trajectories.

Both of the above approaches have undesirable features. These features arise because both approaches use a single state sequence to estimate the target. Whilst the true trajectory is a single sequence, it is one of a set of possible sequences and selecting the correct member of this set is the problem at hand. Another solution can be found by using an alternative target model. The alternative model uses a mixture to describe the target measurements. Each component of the mixture is one of the possible state sequences, and the mixing proportions of these components are the unknown prior probabilities, ${P_\mu}^m(\mu^m)$. The prior probability will be estimated by the algorithm and the hope is that this prior will converge to unity for the value of μ^m corresponding to the correct unwrapped velocity.

This model is somewhat like the homothetic measurement model described in chapter 3 and chapter 6. In the homothetic measurement model, the measurement density is modelled as a mixture of components with the same mean, but with different measurement variances. In this model, the components have the same measurement probability density function, but different means.

Two different approaches for ambiguous velocity unwrapping were implemented and tested on radar data. These approaches were to treat the index of the correct dynamic model as a parameter to be estimated, and the mixture approach described above. The missing data option was not implemented since different state vectors are used for the different models.

7.3.1 Estimation Approach for Velocity Unwrapping

This approach is substantially the same as the standard PMHT algorithm. What is introduced in addition is a target dynamic index, which is to be estimated. On the first run, a particular value for this index is chosen - for example the target is initially assumed to be outbound with an unaliased velocity. This value is used to generate the initial state sequence, which in turn is used to calculate the assignment weights. Once these weights are obtained, the estimated state sequence and dynamic index are chosen to jointly maximise the auxiliary function. For each value of the index, a particular initial state pdf is assumed, and a Kalman smoother run over the measurements. The new estimated index is then the one corresponding to the smoother output with the highest likelihood (the maximum associated auxiliary value). The new state sequence is the output of the smoother, and the algorithm moves to the next iteration.

Thus the estimation approach consists of the following steps:

1. Initialisation.
   (a) Choose an initial unwrapped velocity index, $\mu^{m(0)}$.
   (b) Initialise the state estimate sequence where $\boldsymbol{x}_0^{m(0)}$ is determined by the index $\mu^{m(0)}$.
2. Calculate measurement assignment weights, $w_{mtr}^{(i)}$, using the standard PMHT definition.
3. Maximisation.
   (a) For each index $\mu^m = 1 \dots M_\mu$ determine the optimal state sequence using a Kalman Smoother where $\boldsymbol{x}_0^m$ is determined by μ^m, and synthetic measurements and covariances are defined in the standard PMHT manner.
   (b) For each index $\mu^m = 1 \dots M_\mu$ determine the conditional likelihood of the optimal state sequence, $Q_X^m(\mu^m)$, using the target model dependent part of the standard PMHT auxiliary function.
   (c) Update the state estimate sequence, $\mathbf{X}^{(i)}$, by choosing the optimal sequence for the index with the highest $Q_X^m(\mu^m)$, and select the updated velocity index, $\mu^{m(i)}$, as the index of the highest $Q_X^m(\mu^m)$.
4. Repeat steps 2 and 3 until convergence.

7.3.2 Unwrapped Velocity as a Mixture Model

Recall that the target dynamics follows one of M_μ known models, but that the particular one of which is unknown. Each of these component models has its own state, prior density, evolution density and measurement density functions. Let the state of the pth component of target model m at scan t be denoted as $\boldsymbol{x}_t^{mp}$. The state of model m at scan t is the set of all component states

$$\boldsymbol{x}_t^m \equiv \left\{ \boldsymbol{x}_t^{m1}, \boldsymbol{x}_t^{m2}, \dots \boldsymbol{x}_t^{mM_\mu} \right\}. \tag{7.24}$$

Each component has its own measurement and evolution probability densities, $\zeta_t^{mp}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{mp}\right)$, $\psi_t^{mp}\left(\boldsymbol{x}_t^{mp}|\boldsymbol{x}_{t-1}^{mp}\right)$, and $\psi_0^{mp}\left(\boldsymbol{x}_0^{mp}\right)$.

For each measurement, $\boldsymbol{z}_{tr}$, the index k_{tr} denotes the model that was the true source of that measurement. Let the index μ_{tr} define which component of model m produced the measurement $\boldsymbol{z}_{tr}$. μ_{tr} is an integer in the range $1 \dots M_\mu$. Define the sets

$$\mu_t \equiv \{\mu_{t1}, \mu_{t2}, \dots \mu_{tn_t}\}, \tag{7.25}$$

and

$$\mu \equiv \{\mu_1, \mu_2, \dots \mu_T\}. \tag{7.26}$$

Notice that the component index, μ_{tr}, does not have an explicit model dependence, i.e. μ_{tr}^m. This is because there is only one index per measurement, not an index for each measurement and model pair. This is similar to the assignment index k_{tr}.

The underlying physical model for the problem requires that $\mu_{tr} = \mu_{ts}$ for all scans t and measurements r, s if $k_{tr} = k_{ts}$. That is, there is really only one target model. Under the physical model, the prior ${P_\mu}^m(\mu^m)$ is the probability that μ^m represents the proper dynamic model for the target corresponding to model m. Here, a different, nonphysical approach is adopted. Instead, it is assumed that m represents a mixture process and that μ_{tr} is independent of μ_{ts} provided that r and s are not identical. The probability ${P_\mu}^m(\mu_{tr})$ is now the unknown mixing proportions of the model. The probability is independent of the measurement index, r, implicitly assuming the μ_{tr} to be independent identically distributed random variables, given that they are due to the same target, i.e. the corresponding k_{tr} is the same.

Let

$${\mathbf{P}_\mu}^m \equiv \{{P_\mu}^m(1), {P_\mu}^m(2), \dots {P_\mu}^m(M_\mu)\}, \tag{7.27}$$

and

$$\mathbf{P}_\mu \equiv \left\{{\mathbf{P}_\mu}^1, {\mathbf{P}_\mu}^2, \dots {\mathbf{P}_\mu}^M\right\}, \tag{7.28}$$

The indices μ shall now be treated as missing data and marginalised from the optimisation problem.

Since the μ are missing data, the new auxiliary function is the expectation of the complete data likelihood over both the assignments, $\mathbf{K}$, and the μ. So the auxiliary function is

$$Q\left(\mathbf{X}, \mathbf{\Pi}, \mathbf{P}_\mu | \mathbf{X}^{(i)}, \mathbf{\Pi}^{(i)}, {\mathbf{P}_\mu}^{(i)}\right) = \sum_\mu \sum_K \log L(\mathbf{O}, \mathbf{Z}) P\left(\mathbf{K}, \mu | \mathbf{X}^{(i)}, \mathbf{Z}\right), \tag{7.29}$$

where

$$\sum_\mu \{\cdot\} \equiv \sum_{\mu^1=1}^{M_\mu} \sum_{\mu^2=1}^{M_\mu} \dots \sum_{\mu^M=1}^{M_\mu} \{\cdot\}. \tag{7.30}$$

The complete data likelihood is given by

$$\begin{aligned} L(\mathbf{O}, \mathbf{Z}) &= P(\mathbf{X})P(\mathbf{K})P(\mu)P(\mathbf{Z}|\mathbf{X}, \mathbf{K}, \mu) \\ &= \prod_{m=1}^{M} \left\{ \prod_{p=1}^{M_\mu} \left[\psi_0^m\left(\boldsymbol{x}_0^{mp}\right) \prod_{t=1}^{T} \psi_t^{mp}\left(\boldsymbol{x}_t^{mp} | \boldsymbol{x}_{t-1}^{mp}\right) \right] \right\} \\ &\quad \prod_{t=1}^{T} \left\{ \prod_{r=1}^{n_t} \left[\pi_t^{k_{tr}} {P_\mu}^{k_{tr}}(\mu_{tr}) \zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr} | \boldsymbol{x}_t^{k_{tr}\mu_{tr}}\right) \right] \right\}. \end{aligned} \tag{7.31}$$

Using the same simplifying steps as in section 4.1.3, the conditional probability of the indices K and μ is given by

$$\begin{aligned} P\left(\mathbf{K},\mu|\mathbf{X},\mathbf{Z}\right) &= \frac{L(\mathbf{O},\mathbf{Z})}{P\left(\mathbf{X}^{(i)},\mathbf{Z}\right)} \\ &= \frac{P(\mathbf{X})P(\mathbf{K})P(\mu)P(\mathbf{Z}|\mathbf{X},\mathbf{K},\mu)}{\sum_{\mu}\sum_{K} P(\mathbf{X}^{(i)})P(\mathbf{K})P(\mu)P(\mathbf{Z}|\mathbf{X},\mathbf{K},\mu)} \\ &= \frac{\prod_{t=1}^{T}\prod_{r=1}^{n_t}\left[\pi_t^{k_{tr}} {P_\mu}^{k_{tr}}\left(\mu_{tr}\right)\zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}\mu_{tr}}\right)\right]}{\sum_{\mu}\sum_{K}\prod_{t=1}^{T}\prod_{r=1}^{n_t}\left[\pi_t^{k_{tr}} {P_\mu}^{k_{tr}}\left(\mu_{tr}\right)\zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}\mu_{tr}}\right)\right]} \\ &= \prod_{t=1}^{T}\prod_{r=1}^{n_t}\frac{\pi_t^{k_{tr}} {P_\mu}^{k_{tr}}\left(\mu_{tr}\right)\zeta_t^{k_{tr}}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}\mu_{tr}}\right)}{\sum_{p}\sum_{m}\pi_t^m {P_\mu}^m\left(p\right)\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{mp}\right)} \qquad (7.32) \\ &\equiv \prod_{t=1}^{T}\prod_{r=1}^{n_t} w_{mptr}|_{m=k_{tr},p=\mu_{tr}}, \qquad (7.33) \end{aligned}$$

where the superscript (i) has been omitted from $\mathbf{X}$, $\mathbf{\Pi}$ and $\mathbf{P}_\mu$ to slightly simplify notation. The weight w_{mptr} is implicitly dependent on these estimates from the previous EM iteration.

Following the steps set out in section 4.1.5, the auxiliary function can now be written as

$$\begin{aligned} Q\left(\mathbf{X},\mathbf{\Pi},\mathbf{P}_\mu|\mathbf{X}^{(i)},\mathbf{\Pi}^{(i)},{\mathbf{P}_\mu}^{(i)}\right) &= \sum_{\mu}\sum_{\mathbf{K}}\Bigg\{\log\left[P(\mathbf{X})P(\mathbf{K})P(\mu)P(\mathbf{Z}|\mathbf{X},\mathbf{K},\mu)\right] \\ &\qquad \prod_{t=1}^{T}\prod_{r=1}^{n_t} w_{mptr}|_{m=k_{tr},p=\mu_{tr}}\Bigg\} \\ &= \sum_{m=1}^{M}\sum_{p=1}^{M_\mu} Q_X^{mp} + \sum_{m=1}^{M} Q_\mu^m + \sum_{t=1}^{T} Q_{t\pi}, \qquad (7.34) \end{aligned}$$

where

$$Q_X^{mp} = \log\psi_0^{mp}\left(\boldsymbol{x}_0^{mp}\right) + \sum_{t=1}^{T}\log\psi_0^{mp}\left(\boldsymbol{x}_t^{mp}|\boldsymbol{x}_{t-1}^{mp}\right) + \sum_{t=1}^{T}\sum_{r=1}^{n_t} w_{mptr}\log\zeta_t^{mp}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{mp}\right), \qquad (7.35)$$

$$Q_\mu^m = \sum_{p=1}^{M_\mu}\log {P_\mu}^m(p)\sum_{t=1}^{n_t}\sum_{r=1}^{n_t} w_{mptr}, \qquad (7.36)$$

and

$$Q_{t\pi} = \sum_{m=1}^{M}\log\pi_t^m\sum_{p=1}^{M_\mu}\sum_{r=1}^{n_t} w_{mptr}. \qquad (7.37)$$

Apart from trivial notational differences, (7.35) is identical to the standard PMHT model expression for Q_x^m, given in (4.28) and is optimised in the same way. As shown in

section 4.1.5, the solution for the states $\boldsymbol{x}_t^{mp}$ can be realised using a Kalman Smoother when the statistics are linear and Gaussian. The important result in (7.34) is that the state sequences of each of the components of model m can be estimated independently.

Similarly, (7.37) varies only slightly from the standard PMHT expression, given in (4.29). Using a Lagrangian approach, as in section 4.1.5, it is simple to show that optimising (7.37) gives the updated prior

$$\pi_t^m = \frac{\sum_{p=1}^{M_\mu} \sum_{r=1}^{n_t} w_{mptr}}{\sum_{s=1}^{M} \sum_{p=1}^{M_\mu} \sum_{r=1}^{n_t} w_{sptr}}. \tag{7.38}$$

The expression for Q_μ^m in (7.36) is again almost identical to (4.29) and can easily be shown to be optimised by

$$P_\mu{}^m(p) = \frac{\sum_{t=1}^{T} \sum_{r=1}^{n_t} w_{mptr}}{\sum_{s=1}^{M_\mu} \sum_{t=1}^{T} \sum_{r=1}^{n_t} w_{sptr}}. \tag{7.39}$$

When the algorithm converges, the component with the highest $P_\mu{}^m(p)$ can be chosen as the dynamic model for target model m and the state estimate is given by the $\boldsymbol{x}_t^{mp}$ sequence for that component. Prior knowledge (or belief) about the prior probability of particular target velocities can be incorporated though the initial values of the $\mathbf{P}_\mu$. In particular, setting $P_\mu{}^{m(0)}(p) = 0$ for a specific component will result in $P_\mu{}^m(p) = 0$ for that component on convergence.

This approach significantly increases the computation requirements of the algorithm, since each target model now has M_μ weights per measurement and requires M_μ parallel Kalman Smoothers for state estimation. This increased cost is similar to that incurred by the PDAF approach for resolving the ambiguous velocity problem, given in [Col99].

7.3.3 Doppler Unwrapping Performance

The two above approaches were implemented and run over a collection of JFAS data sets. The data used has a higher than average update rate (a small time delay between scans). This makes the doppler ambiguity problem more difficult because there is little time to observe target motion. The fraction of correctly unwrapped tracks is shown in table 7.1. The trials were conducted for various batch lengths - longer batches provide more time for the target to move and hence make velocity resolution easier. However, with a real-time sensor, the batch length should be limited to enable speedy track initiation.

The results in table 7.1 show that the estimation approach gives much better performance than the mixture model. The performance of the estimation approach is almost constant with batch length, which allows for the use of a short batch. As stated above, this is desirable for timely response of the tracker.

This result may be somewhat surprising - intuition suggests that the hard decision made in the estimation approach might cause the performance to be poor. In practice, the estimation approach is better because it incorporates the state likelihood as well as the measurement likelihood. The mixture approach makes a decision based on the relative probability of each component. This probability is determined by the measurement

	batch length			
method	4	6	8	10
estimated	0.74	0.77	0.78	0.78
mixture	0.17	0.23	0.52	0.63

Table 7.1: Proportion of correctly unwrapped velocities

assignment weights as given in (7.39). This probability depends on how well the measurements match the component states, but does not consider how consistent the component states are with their dynamic models. When each component state sequence is adjusted, they are made to follow the measurements, and the relative values of the process and measurement noise covariances, Q_t^m and R_t^m, determines the trade-off between following measurements and model consistency. Over short periods of time, all of the components can be made to fit the measurements, and random noise fluctuations become important to the final decision. When the batch is longer, the dynamics prevent all components from being close to later measurements, and the performance is better. In contrast, the estimation approach includes both measurement terms and evolution terms, since it maximises (7.35) over the component index, p. This penalises components whose dynamics poorly match the optimal state sequence, and a better result is obtained.

7.4 Using PMHT-c for Clutter Parameterisation

The standard JFAS tracker models the clutter distribution as a mixture of two components. Each component is uniform in the spatial dimensions (azimuth, range and doppler) and non-uniform in amplitude. The tracker formulation allows for variation in this model, such as a non-uniform doppler distribution [Col99]. The non-uniform distribution is assumed to have a known functional form, with unknown parameters. The labels are assumed to perfectly divide peaks into the two classes, and the peak subsets are used to estimate the unknown distribution parameters. This process is represented in figure 7.7.

The process in figure 7.7 implicitly assumes that the segmentation of the clutter measurements is perfect, that is the segmentation labels are correct. In fact, these labels do not have to be taken as perfectly valid. Rather, they can be incorporated into the tracking algorithm as classification measurements, as is preempted by the notation used earlier in this chapter. The veracity of the classifier that produces these labels is unknown, and it cannot be estimated by the use of training data, since there is no way of determining true classifications for radar measurements. Besides which, the two component mixture is only an assumed model, and will likely not be the true underlying density of the clutter.

Even though training data is unavailable, the results of chapter 4 indicate that significant advantage might be gained by relaxing the assumption that the classifier is perfect, and that the precise confusion matrix need not be known. The confusion matrix can be estimated, or can be assumed to be some arbitrary imperfect matrix, and it can be expected that performance will be improved.

Assume that the clutter is caused by a two component mixture. These components will be described using models 1 and 2. Hence, $M_Y = 2$ and models $m = 3 \dots M$ represent targets. The classification measurement (the segmentation output) takes either the value 1 or 2, indicating which clutter component the preprocessor believes to be the source of

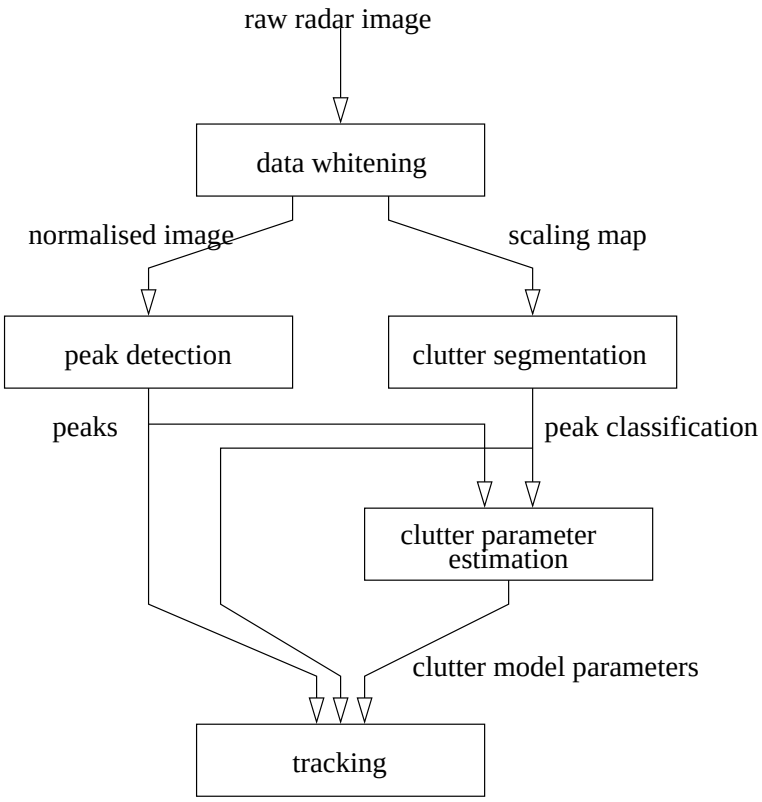


Figure 7.7: Clutter Segmentation Process

the corresponding state measurement. This means that the confusion matrix will not be square; there are two observable classes, but M models. This is because the classification measurement provides no information about targets.

Assume that the confusion matrix is of the form

$$\mathsf{C} = \begin{bmatrix} \alpha_1 & 1-\alpha_2 & \beta & \cdots & \beta \\ 1-\alpha_1 & \alpha_2 & 1-\beta & \cdots & 1-\beta \end{bmatrix}. \tag{7.40}$$

The current approach treats the classification information as perfect, which is equivalent to assuming that $\alpha_1 = \alpha_2 = 1$.

The feature used for classification is a measure of the local noise floor. The probability that a target measurement will give rise to a classification $z_{tr}^{(k)} = 1$ is therefore the proportion of the surrounding surveillance region with a local noise level corresponding to clutter model 1. This proportion can be approximated by the mixing proportions of the clutter components, and β is assumed to be given by

$$\beta = \frac{\sum_{t=1}^{n_t} \pi_t^1}{\sum_{t=1}^{n_t} \left(\pi_t^1 + \pi_t^2\right)}. \tag{7.41}$$

The problem can now directly be approached as one of the form solved by the PMHT-c algorithm presented in chapter 4. The value of the classifier veracity, α, can be estimated, or simply assumed to be a particular number.

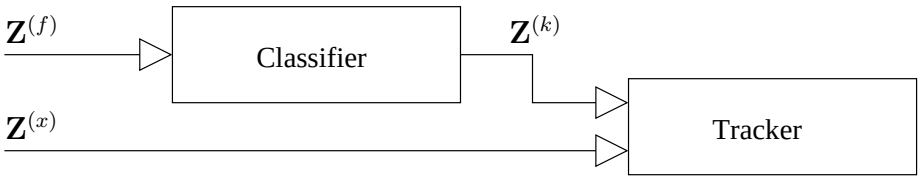


Figure 7.8: Distributed Fusion Model

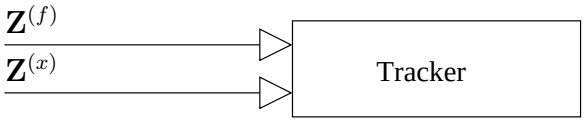


Figure 7.9: Centralised Fusion Model

7.4.1 Centralised Fusion Option

Using the PMHT-c algorithm to incorporate the classifier output can be viewed as a distributed data fusion approach. The sensor receives state observations and feature information. The feature information is used as the input to a classification algorithm that in turn acts as input to the tracker. This architecture is shown in figure 7.8. For each measurement, $\boldsymbol{z}_{tr}^{(x)}$, there is a corresponding feature, $\boldsymbol{z}_{tr}^{(f)}$. The set of all feature information is labelled as $\mathbf{Z}^{(f)}$.

An alternative to this strategy is to feed the feature information as a direct input to the tracker. The feature itself becomes a form of augmented measurement. The tracker then assumes a probabilistic model for the distribution of the feature for each model, and it can be used in much the same way as the amplitude measurement. This architecture is shown in figure 7.9.

In the centralised architecture, the tracker jointly performs the tasks of classification and state estimation. Since the tracker has access now to the feature data, rather than simply a binary decision from the classifier, it is expected that the performance of the algorithm will improve. Some of the decisions in the classifier will be indisputable while others are borderline. By coupling the classification decision into the tracker, the tracker gains knowledge about the strength of the evidence for class assignments and this will improve performance. If the classifier instead provided probabilities of each class, then there would be no advantage to the centralised approach.

The viability of joint classification and tracking in the form shown in figure 7.9 depends on the features and how much information is available about their distribution. The centralised architecture concatenates the features and the state observations to form augmented measurements. For the tracker to be able to use the augmented measurements, the probability density of the each feature for each class must be known, or estimated.

The feature that is used in the Jindalee OTHR (a measure of the local noise floor) is a continuous valued scalar and its distribution is unknown. However, the feature values for all measurements were collected together and used to produce a kernel estimate of the probability density of the logarithm of the feature. This density estimate is shown in figure 7.10. The figure shows the estimated density for two different data sets. The first data set contains relatively benign clutter, and the second data set contains more challenging clutter caused by range folded spread clutter. In each case, the distribution is multimodal, but the positions of the modes is different. Also the variance of the

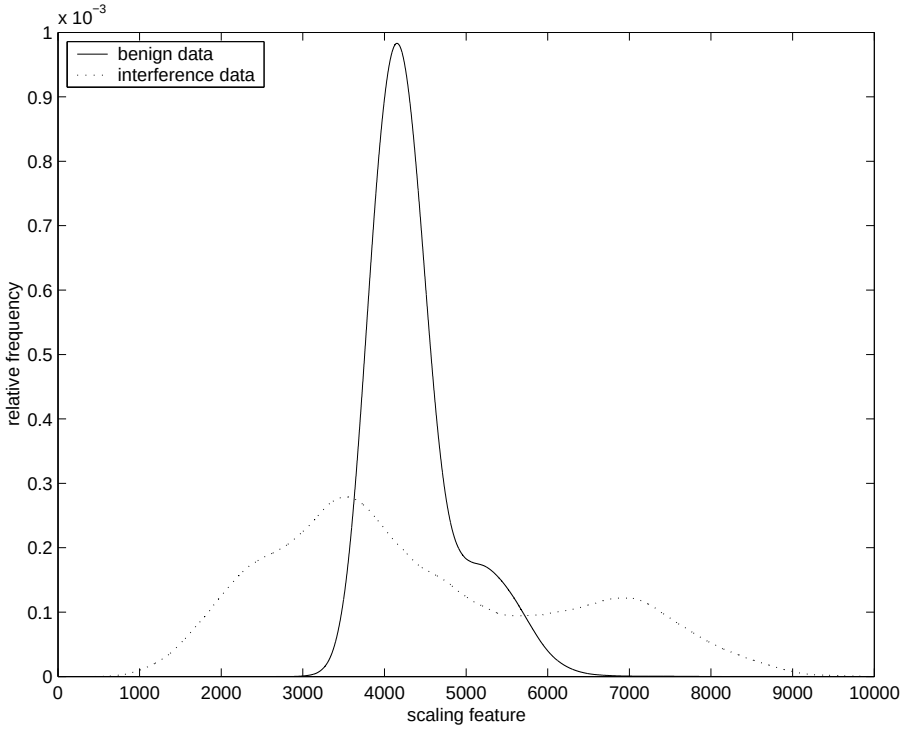


Figure 7.10: Estimated feature pdf, $p\left(z_{tr}^{(f)}\right)$

distribution is much higher for the interference data than the benign data.

The two different feature densities in figure 7.10 can each be approximated as a two component Gaussian mixture. As observed above, the variance of the components is much higher in the interference data, and the means are farther apart. So it is necessary to use a dynamic feature pdf which will adapt to the data. The bimodal nature of the densities in figure 7.10 suggests that two different processes are at work, and was the motivation for the original decision to use a two component model for clutter in the JFAS tracker.

Past analysis of the feature data and the measurement amplitude data has shown that high values of the feature correspond to regions contaminated by interference where the amplitude distribution is more spread [RCD97, RC98]. Thus, it is reasonable to attribute the rightmost peak of the feature distribution to interference measurements and the leftmost peak to benign background measurements. Each clutter component is assumed to give rise to features which have a Gaussian distribution. The means and variances of these two Gaussians are dynamic unknowns, but are assumed to be constant over the measurement batch.

Thus the clutter state vector for $m = 1, 2$ is now

$$\boldsymbol{x}_t^m \equiv \boldsymbol{x}^m = \left\{x^m[a], x^m[\mu], x^m[\sigma^2]\right\}^{\mathsf{T}}, \tag{7.42}$$

where $x^m[a]$ is the parameter of the exponential distribution for measurement amplitude, and $x^m[\mu]$ and $x^m[\sigma^2]$ are the parameters of the Gaussian feature pdf. The augmented measurement pdf is

$$\zeta_t^m\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) = \frac{1}{A}\frac{1}{x^m[a]}\exp\left\{-\frac{z_{tr}[a]-\nu_t}{x^m[a]}\right\}\frac{1}{\sqrt{2\pi x^m[\sigma^2]}}\exp\left\{-\frac{\left(z_{tr}^{(f)}-x^m[\mu]\right)^2}{2x^m[\sigma^2]}\right\}, \tag{7.43}$$

where it has been assumed that the feature measurement is independent of the amplitude measurement.

The clutter state components are then estimated using their Maximum Likelihood Estimators (MLEs) given weighted measurements. These MLEs for the feature parameters are the weighted sample mean and variance given by

$$\hat{x}^m[\mu] = \frac{\sum_{t=1}^{T}\sum_{r=1}^{n_t} w_{mtr} z_{tr}^{(f)}}{\sum_{t=1}^{T}\sum_{r=1}^{n_t} w_{mtr}}, \tag{7.44}$$

and

$$\hat{x}^m[\sigma^2] = \frac{\sum_{t=1}^{T}\sum_{r=1}^{n_t} w_{mtr}\left(z_{tr}^{(f)} - \hat{x}^m[\mu]\right)^2}{\sum_{t=1}^{T}\sum_{r=1}^{n_t} w_{mtr}}. \tag{7.45}$$

The feature pdf of target measurements is assumed to be a mixture of the clutter pdfs, where the two components are weighted by the estimated mixing proportions of the clutter models. The state observation is assumed independent of the feature. Thus for targets,

$$p\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) = \zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m\right) p\left(z_{tr}^{(f)}\right), \tag{7.46}$$

where

$$p\left(z_{tr}^{(f)}\right) = \frac{\hat{\pi}_t^1}{\hat{\pi}_t^1 + \hat{\pi}_t^2} p\left(z_{tr}^{(f)}|\hat{\boldsymbol{x}}^1\right) + \frac{\hat{\pi}_t^2}{\hat{\pi}_t^1 + \hat{\pi}_t^2} p\left(z_{tr}^{(f)}|\hat{\boldsymbol{x}}^2\right). \tag{7.47}$$

7.4.2 Simulated Performance of PMHT-c for Clutter Parameterisation

The performance of the various approaches for incorporating the clutter feature information is first investigated through simulations. These simulations are similar to the those used to compare initiation methods in chapter 6. As with the analysis in chapter 6 the performance of the competing approaches will be compared using an experimental initiation ROC curve. For all cases, PMHT-v method is used for track initiation.

The measurement space is two dimensional, as before, but now the measurements are augmented with a third observation which provides classification information about the clutter. This observation is the feature measurement.

Two alternative processing schemes are considered. The first method is the distributed fusion process described in the previous section. Under this approach, the feature measurement is thresholded to provide a binary estimate, indicating which clutter component is associated with the measurement. This is analogous to the current JFAS processing. Thus the feature measurement, $z_{tr}^{(f)}$, is converted to a classification measurement, $z_{tr}^{(k)}$. The PMHT-c is then used, as outlined in section 7.4. Three forms of the PMHT-c are considered. Firstly, the classifier veracity is assumed to be perfect ($\alpha_1 = \alpha_2 = 1$), which is analogous to the current JFAS approach. Next, the veracity is assumed to be $\alpha_1 = \alpha_2 = 0.9$, and thirdly, it is estimated. The analysis in chapter 4 suggests that the second of these three will provide the best performance.

clutter component	$x^m[y]$	uninformative $x^m[\mu]$	uninformative $x^m[\sigma^2]$	medium $x^m[\mu]$	medium $x^m[\sigma^2]$	perfect $x^m[\mu]$	perfect $x^m[\sigma^2]$
model 1	1	0	1	-2.5	2	-2.5	0.1
model 2	7	0	1	0.5	1	0.5	0.1

Table 7.2: Clutter parameters

The second processing scheme is the centralised fusion approach described in the previous section. Here, the feature itself is simply treated as a third measurement dimension. The target is assumed to have a mixture distribution in the feature domain, where the components of the mixture are the two clutter distributions and the mixing proportion is derived from their relative frequency, namely β as defined in section 7.4.

Finally, two reference curves are generated. For the first curve, the tracker is provided with perfect classification information (at least between the clutter distributions). This will then give the best performance attainable by any of the other approaches. The second curve is the performance obtained using the standard PMHT algorithm, which effectively ignores the classification information. This provides a benchmark; performance worse than the standard PMHT with useful classifications would be a poor result for any approach.

The measurements used in the simulation are random realisations of the two component clutter process, and target measurements (when it is present and detected). Performance is quantified through ROC curves, and the scenario is chosen to make initiation deliberately difficult to bring out differences in the approaches.

The two dimensional measurement vector is denoted by $\boldsymbol{z}_{tr}^{(x)} \equiv \{z_{tr}[x], z_{tr}[y]\}^{\mathsf{T}}$. For both clutter components, the measurement vector is uniformly distributed in $z_{tr}[x]$, exponentially distributed in $z_{tr}[y]$ and has a Gaussian distribution in $z_{tr}^{(f)}$. The exponential parameters (namely the means) are fixed, and the Gaussian parameters are varied to achieve different classification veracities. Three different classifier veracities are considered: uninformative classifications, medium veracity classifications, and perfect classifications. These parameters are summarised in table 7.2. The clutter state vector is given by $\boldsymbol{x}^m = \{x^m[y], x^m[\mu], x^m[\sigma^2]\}^{\mathsf{T}}$, where $x^m[y]$ is the parameter of the exponential distribution for the y coordinate measurement, and $x^m[\mu]$ and $x^m[\sigma^2]$ are the parameters of the Gaussian distribution for the feature measurement.

As given in (7.47), the target feature distribution is a mixture of the clutter distributions, when the target is present. The target dynamic model and measurement process are the same as that used in the initiation comparisons of chapter 6. Namely, the target uses an almost constant velocity model with independent velocity perturbations in the x and y coordinate directions.

When the classifier is used, it thresholds the $z_{tr}^{(f)}$ measurement at 0. In the uninformative example, both classes have equal probability of being labelled class 1 or 2, so $\alpha_1 = \alpha_2 = 0.5$. In the medium veracity case, the classification veracities can be obtained by integrating the $z_{tr}^{(f)}$ pdf over the decision region, and are $\alpha_1 = 0.95$ and $\alpha_2 = 0.76$. In the perfect classification case, $\alpha_1 = \alpha_2 = 1$.

The mixing proportions of the components are chosen so that three quarters of the clutter measurements are due to component 1. So, when no target is present, $\pi_t^1 = 0.75$ and $\pi_t^2 = 0.25$.

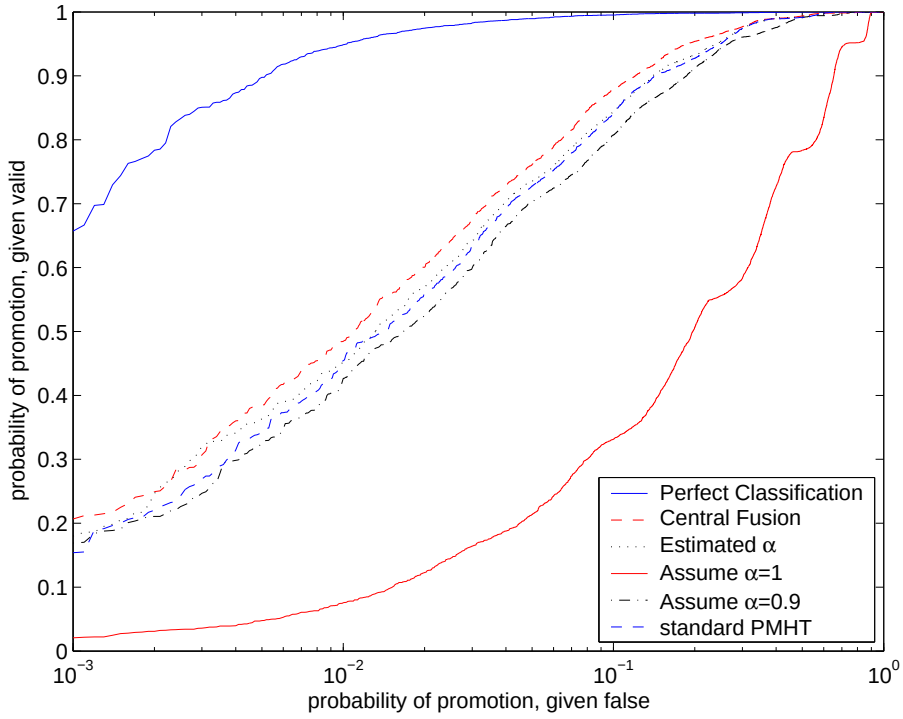


Figure 7.11: PMHT-c for clutter parameterisation, uninformative classifications

7.4.3 Simulation Results

ROC curves measured for the three feature distributions are now presented. In each case, the performance of the PMHT-c based algorithms is at least as good as the standard PMHT. The adaptive PMHT-c performs better than assuming perfect classifications, or assuming $\alpha = 0.9$. The centralised fusion algorithm performs better than the PMHT-c. This centralised algorithm is only available because the pdf of the feature is a known function. In many practical cases, this algorithm may not be realisable.

7.4.3.1 Uninformative Classification

The ROC curves for the case of an uninformative feature are given in figure 7.11. Under this condition, it is not expected that any of the new approaches will perform better than the standard PMHT. There is no extra information available to exploit. The perfect classification curve is still shown to give a performance bound. The PMHT-c algorithm which assumes a diagonal confusion matrix gives very poor results. This is because it is assuming that useless information is faultless. The other approaches all give roughly similar results, with no significant differences. The central fusion algorithm is slightly better than the others, and the PMHT-c which assumes that the classifier veracity is $\alpha = 0.9$ is slightly worse. The PMHT-c which estimates the confusion matrix, and the standard PMHT (which does not use the classification measurements) give almost exactly the same performance. This tends to indicate that the estimated α is behaving correctly.

7.4.3.2 Medium Veracity Classification

The ROC curves for the medium veracity example are given in figure 7.12. This case is the most interesting because it demonstrates the differences between the various approaches.

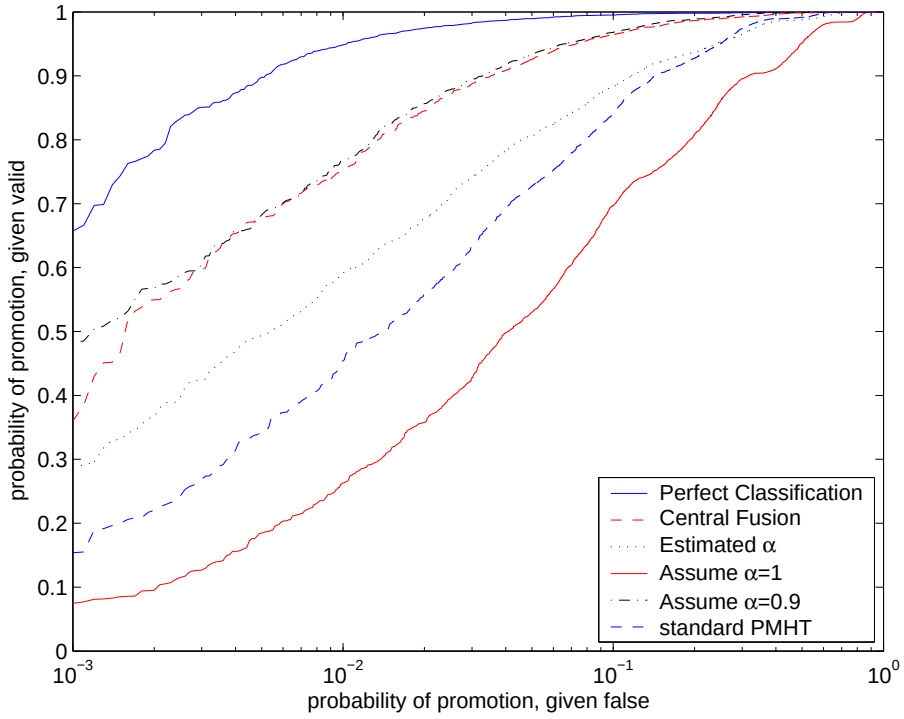


Figure 7.12: PMHT-c for clutter parameterisation, medium veracity classifications

The PMHT-c algorithm which assumes $\alpha = 1$ actually degrades the performance over the standard PMHT. Thus, assuming the classifications to be perfect is more detrimental than discarding them completely. The adaptive PMHT-c has better performance. However, assuming the veracity to be $\alpha = 0.9$ is superior. This is in accordance with the results reported in chapter 4, where assuming $\alpha = 0.9$ appeared to be a sound strategy, independent of the true veracity. The centralised fusion algorithm performs better than the adaptive PMHT-c because it has access to the feature information which is a much richer source than the classification measurements, however it has the same performance as the PMHT-c with $\alpha = 0.9$. This result is somewhat surprising; intuition would predict that the centralised algorithm would be a clear winner. It indicates that the classifications adequately summarise the feature data, for the purposes of this problem.

7.4.3.3 Perfect Classification

The ROC curves for perfect classification accuracy are given in figure 7.13. All of the algorithms have very similar performance to the reference curve for perfect classification knowledge, because all have this knowledge. The adaptive PMHT-c, which estimates α, does slightly worse that the others. The PMHT performance is significantly worse, since it ignores the classifications.

7.4.4 Summary of Clutter Modelling with PMHT-c

The standard JFAS processing gives rise to feature measurements which the JFAS tracker uses to segment the state observations into background and interference classes. This segmentation step is a classification process where the class output is a binary random variable. The JFAS tracker assumes that this classification information is error free and

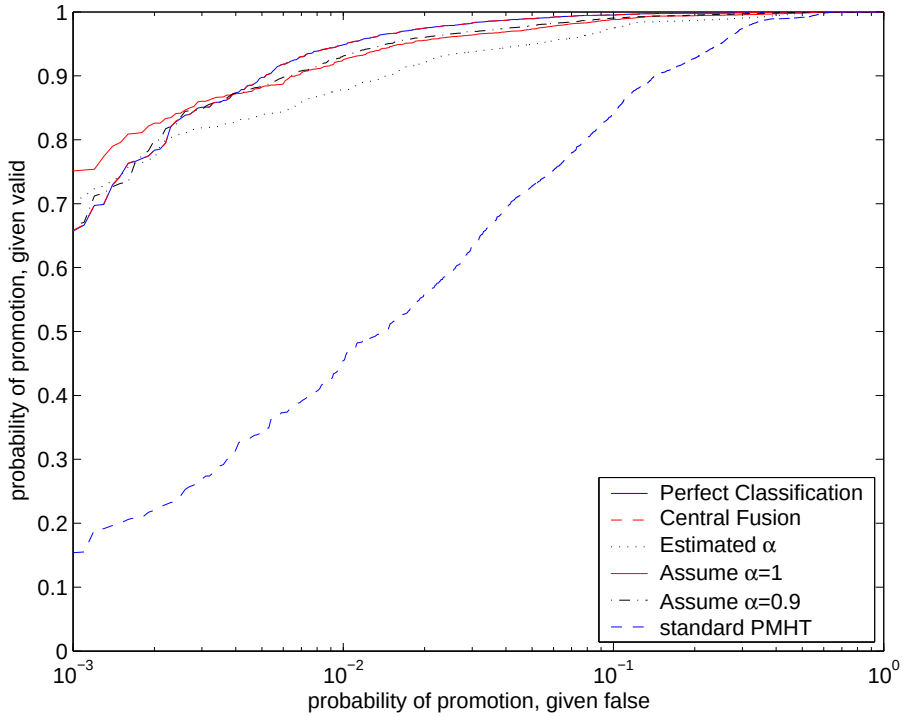


Figure 7.13: PMHT-c for clutter parameterisation, perfect classifications

uses it to estimate the parameters of a two component clutter model, and to identify which component of the clutter is the source of each measurement if it is not due to a target.

The PMHT-c provides a framework for incorporating this classification information which allows for errors in the classifier. Since the true confusion matrix of the classifier is unknown, the PMHT-c must either assume a particular confusion matrix, or estimate it from the data. Both of these possibilities were considered, with different assumed confusion matrices. The algorithm with an assumed confusion matrix of the identity matrix is analogous to the JFAS approach which assumes that the classifications are correct.

An alternative approach was identified whereby the classification step is removed completely, and the tracker instead uses the feature information directly. Under this approach, the tracker must assume distributions for the feature data for clutter and target measurements. Empirical data

The various PMHT based approaches for clutter modelling were simulated and their relative performance observed via track initiation ROC curves. It was seen that the PMHT-c which assumes a non-ideal confusion matrix, and the PMHT-c with an estimated confusion matric both gave performance benefit over the standard PMHT which ignored the classification information. The PMHT-c which assumed the classifications were always correct performed worse than the standard PMHT - it was more detrimental to assume the information was perfect than to ignore it. The approach which fed the feature information directly to a standard PMHT algorithm gave the best performance of the approaches considered.

7.5 Full Radar Algorithm

Chapters 4 and 6 provided a number of theoretical advancements of the PMHT algorithm, and this chapter has also presented solutions to particular OTHR data issues in the previous sections. All of these algorithmic extensions have been presented in isolation. Realistically, it is desirable to incorporate many of these advances into a single algorithm for application to the radar data. The derivation of such an algorithm is essentially a matter of careful notational accounting and is not presented here. This is because of the amenable structure of the PMHT framework. Instead, an example of one of the algorithms run on radar data is presented in Appendix B. The example used does not necessarily use the best of competing approaches (for example ambiguous velocity resolution) but instead demonstrates how the various algorithm changes are incorporated in a modular fashion.

This algorithm incorporates:

- the PMHT-c approach for clutter parameterisation with estimated classifier confusion matrix
- the innovation homothetic model for initialisation
- the PMHT-vm approach for track initiation
- the mixture approach for ambiguous velocity resolution

7.6 Radar Data Performance

The results of applying the PMHT enhancements derived in this thesis to recorded OTHR data are now analysed. The performance of the various algorithms is quantified through an empirical ROC curve for track initiation. This curve is generated by a similar method to that derived from simulation in chapter 6 and in section 7.4.3. The difference is that each of the tracks used in the simulated examples is known to be either valid or false, since this is predetermined by the simulation. When data from a real sensor is used, it is not axiomatic which measurements are valid and which are false. This underlying truth is difficult to obtain, yet it is vital to the analysis. Mislabelled tracks with high or low quality might easily lead to incorrect ROC curves.

The first step required in the production of real data ROC curves is the generation of truth. In a coordinated experiment, this might be done using accurate position logging devices, such as a Global Positioning System (GPS). For the purpose of tracking truth, the target position should be known with an accuracy less than the sensor resolution. Since the resolution of OTHR is typically tens of kilometres, a microwave sensor could also be used. However, any such truth method would give estimates of the target position in a geographic co-ordinate system (for example, latitude and longitude). Since the ionospheric truth is inevitably unknown, this makes registration of ground truth and radar measurements unwieldy. So, rather than using secondary measurements, the validity of measurements is determined by direct examination of the data using the JFAS operator displays which are capable of overlaying track reports with the radar image. This approach is used to form a set of valid tracks, and then the candidates are correlated against these, rather than ground truth.

7.6.1 Data Set Features

Two data sets have been selected for use in this analysis. The first data set was collected in early evening. Due to the propagation conditions, and the waveform parameters used, it contains large amounts of interference, in the form of spread doppler clutter. The whitening algorithm somewhat suppresses this interference, but not enough to prevent the standard JFAS tracker producing numerous false tracks. It is known that no targets are present, and the only valid measurements are those due to the calibration track. This data is primarily a test of false track performance. It presents a particularly difficult clutter scenario because the spread doppler clutter occupies a particular spatial region and false detections produced by it are concentrated in this area - in effect the spatial clutter distribution is non-uniform.

The second data set was recorded under more amiable propagation. The data contains a single target of opportunity which is supported by a single mode of propagation. This target is of relatively low amplitude and is sometimes undetected due to signal fading. This data provides a low detectability valid target.

These two data sets are chosen because they avoid some of the more complex problems encountered under multiple mode propagation. Such data, particularly in the presence of multiple targets, are difficult to produce truth for. The data sets are combined to produce a single ROC curve for each algorithm tested.

The calibration signals and the target of opportunity provide two types of target. The calibration signals are high amplitude and are stationary, so they are relatively easy to track. In contrast the target of opportunity has a low amplitude and is relatively difficult to track. The effect of these two classes of target can be seen in some of the ROC curves, where an initial decline in promotion probability is followed by a plateau. In this plateau region, the difficult target has been suppressed, but the calibration tracks are still promoted with quality levels much higher than the threshold.

7.6.2 Clutter Parameterisation

The various different clutter parameterisation approaches described in section 7.4 were implemented and the performance of each quantified through ROC curves. These results are shown in figure 7.14. For each clutter approach, initiation was performed using the PMHT-v algorithm. As mentioned earlier in this chapter, since the data used to produce these ROC curves has been obtained from the JFAS radar, it is not possible to show quantitative scales on the graph axes.

The ROC curves in figure 7.14 confirm the simulated conclusion that the best performance is obtained by removing the classifier and allowing the tracker direct access to the feature information. However, there is not a large advantage gained by this approach. Further, the other methods give almost identical results. These methods are the standard PMHT which ignores the classification measurements, the PMHT-c assuming perfect classification, and the PMHT-c with an estimated confusion matrix. This result is somewhat counterintuitive. The simulation results in section 7.4.2 do not provide a case where the PMHT-c with assumed perfect classifications gives the same performance as the standard PMHT.

A possible cause for the lack of discrimination between clutter approaches is that the estimated clutter parameters for real data are closer than those assumed for the simulation. This means that the difference between the two clutter models being used by the PMHT is not as great. If this is true, then it would be expected that removing one of the clutter

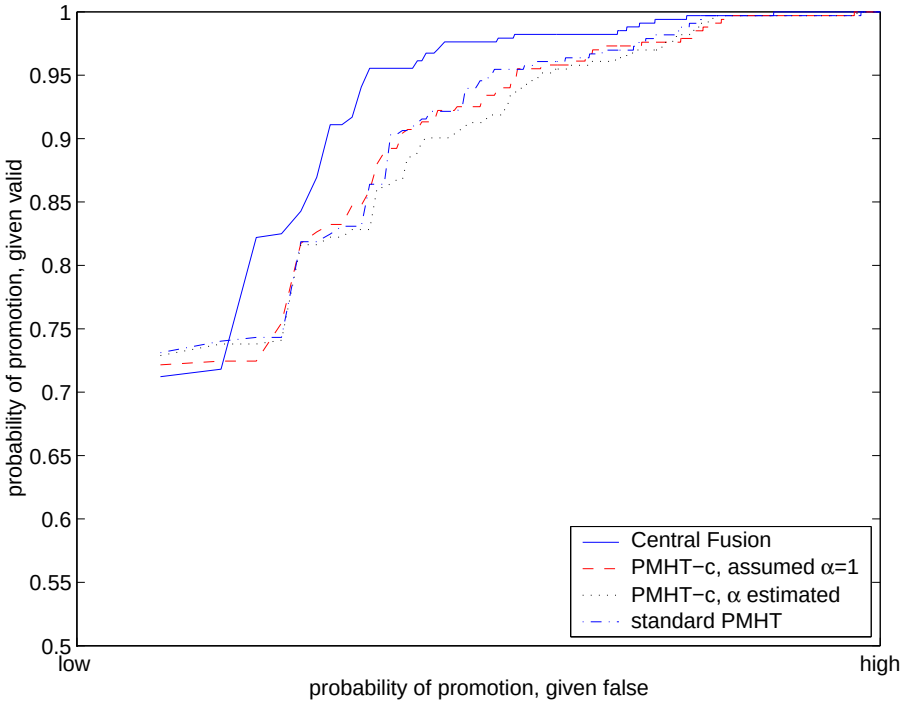


Figure 7.14: ROC curves for different PMHT clutter approaches

models would not make much of an adverse impact on performance. Figure 7.15 shows ROC curves formed using the PMHT-c assuming perfect classifications, and the standard PMHT with only one clutter model. The difference between the curves is only marginal.

The results in figure 7.15 are rather surprising given the experience of introducing a second clutter model with the PDAF. With the JFAS tracking algorithm, the use of two clutter models reduced the false track rate by an order of magnitude [Col99]. The reason why the PMHT does not show the same difference appears to be that the PMHT is more robust to pdf mismatch than the PDAF. Intuitively, this would occur because the PMHT is an iterative hill climbing approach, whereas the PDAF makes only a single adjustment to the state estimate. If the clutter pdf is elevated, then the likelihood ratio between the target and clutter pdfs, which ultimately drives both algorithms, is reduced. This means that the correction to the state estimate due to the measurement is reduced. For the PMHT, this is less important, because the elevated pdf tends to reduce the rate of convergence, rather than changing the convergence point. To confirm this intuition, a study of pdf mismatch with PDAF and PMHT would need to be conducted. This has not been performed as part of this thesis.

7.6.3 Track Initiation

Three different track initiation approaches were presented in chapter 6. Recall that these approaches are

1. The standard PMHT algorithm using the sum of the candidate assignment weights as a promotion test statistic. This approach is referred to as the *weights sum*.

2. The standard PMHT algorithm using the incremental improvement in the EM auxiliary function by introducing a candidate as a promotion test statistic. This approach is referred to as the *cost increment*.

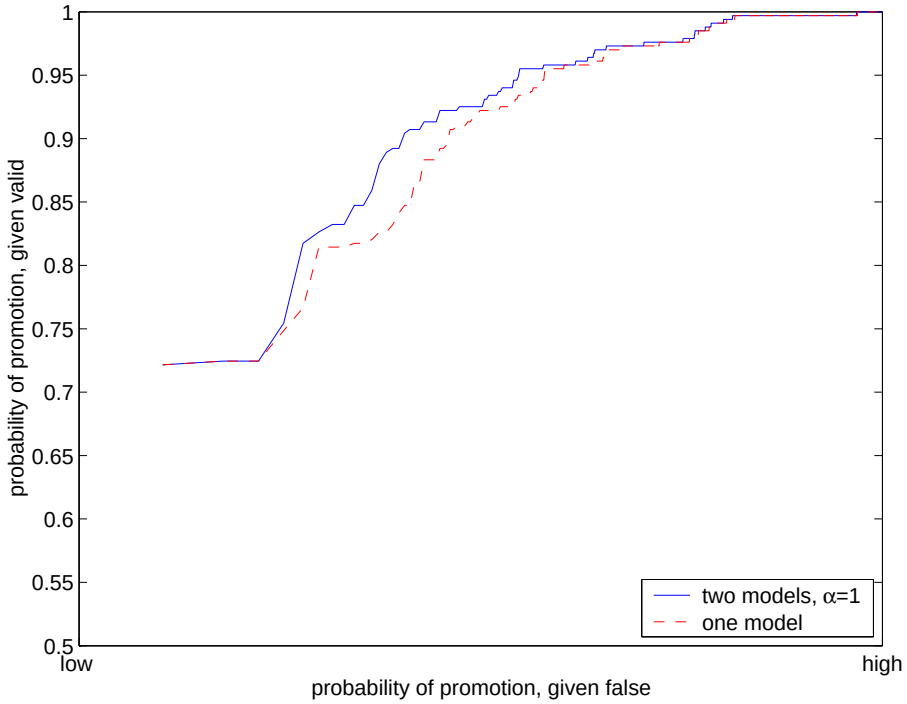


Figure 7.15: PMHT clutter model sensitivity

3. The PMHT-ym algorithm using visibility for the assignment state model. The candidate promotion test statistic is the average probability that the assignment state is that of a visible target. This approach is referred to as PMHT-v.

ROC curves generated using radar data for these three approaches are shown in figure 7.16. As was the general trend in the simulations presented in chapter 6, the PMHT-v gives the best performance.

The clutter pdf for the OTHR data is non-uniform, so the weights sum approach is expected to give the worst performance, this is confirmed by the results. The weights sum also has a step drop off in performance. This occurs because the strong targets are detected in almost all scans. If there are no nearby clutter measurements, the weights sum will thus be approximately equal to the batch length. Once it becomes necessary to raise the promotion threshold above this level, all of those tracks become suppressed.

7.7 Summary

This chapter has presented the results of the implementation of the PMHT for Over the Horizon Radar. The OTHR problem highlights the application of the theoretical enhancements made to the PMHT through the development of the PMHT-c and PMHT-v algorithms. Performance results on radar data have confirmed the simulation results given in earlier chapters and demonstrated the practical utility of the earlier work.

The chapter first reviewed the models used for OTHR tracking and the standard JFAS data processing approach. This lead to the problem of ambiguous velocity measurements, where the ambiguous waveform of the radar and the nature of OTHR targets leads to a multimodal initialisation problem. Alternative methods for addressing this problem with PMHT were identified, and their performance measured on radar data.

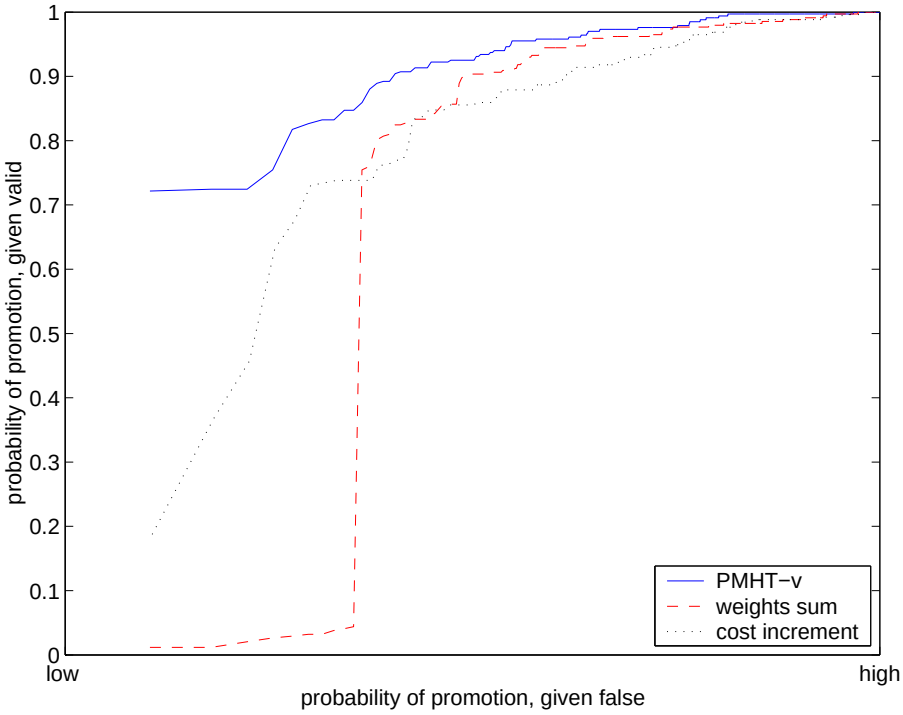


Figure 7.16: Recorded JFAS data initiation ROC

The PMHT-c algorithm was applied to the problem of clutter density parameter estimation. The current JFAS processing uses a multiple model clutter density, and has a classification scheme that segments the radar data. The PMHT-c was used to incorporate the segmentation measurements into the tracker, and allow for errors in the segmentation process. The performance of this approach was measured on simulated data and on radar data. It was found that the greatest gains were made through proper modelling of the segmentation output for target measurements, which substantially reduced false tracks.

The various methods for track initiation with PMHT and PMHT-v were implemented and their performance measured for radar data. As simulations predicted, it was found that the PMHT-v gave the best performance.

Chapter 8

Comparison of the Probabilistic Multi-Hypothesis Tracker with the Probabilistic Data Association Filter

The previous chapters demonstrated that the extensions to the PMHT developed in this thesis provide improved performance over the standard algorithm. However, this is of limited interest if the extended PMHT does not outperform other existing algorithms. In section 1.1, a requirement for an efficient multi-target tracking algorithm for the JFAS radar was cited as motivation for pursuing research into PMHT. To address this, the PMHT algorithm that has been developed should be compared with the current JFAS tracking algorithm, the Unified PDAF (UPDAF). The PMHT is a multitarget algorithm, and the UPDAF is a single target algorithm. Subsequent research since the installation of the UPDAF at JFAS has produced a Unified Joint PDAF (UJPDAF) [DC01], however only the single target UPDAF is considered. The track initiation scenarios considered consist of isolated targets, and multi-target tracking is not required.

In addition to the imperative to justify PMHT research, the comparison of PMHT and PDAF is of interest in itself. Since the PMHT is a recent algorithm, exhaustive studies have not been performed. In particular, the initiation performance of the PMHT has not been studied, since this thesis and [LSW01] are the first approaches presented for automatic initiation. To focus completely on the operational algorithm would give results practical, but rather specific. Therefore simulation studies with a simplified UPDAF are also considered. These simulations provide more general information about the relative performance of PMHT and PDAF in track initiation.

This chapter presents a comparison of track initiation with the PMHT and the PDAF. Firstly, the PDAF extensions leading to the UJPDAF are briefly described. Next, an overview of existing comparisons and higher level considerations is presented. Then the track initiation performance is measured via simulations and using recorded JFAS data. The performance of each algorithm will be quantified via the ROC curve for the track initiation decision.

8.1 The Probabilistic Data Association Filter

The Probabilistic Data Association Filter (PDAF) [BST75, BSF88] is an approximation to the optimal recursive Bayesian data association strategy. The PDAF is a single target algorithm, so each track is filtered in isolation, and it is assumed that any measurements

due to other targets can be lumped into the clutter. The PDAF enforces the single measurement assignment constraint, namely each target track is only allowed to form at most one measurement. Since the PDAF is a single target algorithm, each target model, $m = M_Y + 1 \dots M$, determines independent association probabilities.

The single measurement per target constraint means that there are $n_t + 1$ assignment hypotheses for model m. One for each measurement, and the hypothesis that no measurements are due to the target. Whereas the PMHT derives assignment weights, w_{mtr}, the PDAF uses *event probabilities*. The event probability, $\beta_t^m(r)$, is the posterior probability that the rth measurement at scan t is due to target model m. This probability is determined assuming that each measurement is due either to model m or clutter (the single target assumption).

The standard PDAF makes a number of assumptions about the target and clutter behaviour. The targets are assumed to have linear Gaussian statistics, and all target tracks (i.e. filter models assumed to represent targets) must correspond to valid targets. The clutter is assumed to be uniformly distributed over the surveillance region, and the number of measurements due to clutter is assumed to follow a Poisson distribution.

Further background information about the PDAF is presented in Appendix C.

8.1.1 The Unified PDAF

The Multiple Model Unified PDAF (MM-UPDAF) [CD03] is a non-parametric PDAF that incorporates nearest neighbour validation, multiple non-uniform clutter models, multiple target dynamic models and target visibility. These PDAF extensions are presented in more detail in Appendix C. The MM-UPDAF algorithm is used in the operational JFAS detection and tracking software. The term *multiple model* refers to the use of a set of dynamic models for each target, not to multi-target tracking. These dynamic models can be used both for ambiguous doppler unwrapping, as described in the previous chapter, and for manoeuvring target tracking. With M_μ (known) target dynamic component models, the target model state for model m is $\boldsymbol{x}_t^{mp}$, for $m = M_Y + 1 \dots M$ and $p = 1 \dots M_\mu$. The event probabilities for target model m are denoted $\beta_t^{mp}(r)$, with $\beta_t^{mp}(r)$ assumed to independent of $\beta_t^{nl}(s)$ if $n \neq m$. The MM-UPDAF uses nearest neighbour gating, so that only the I closest measurements to the predicted target measurement are assumed to have non-zero association probabilities.

The JFAS system uses a preprocessor that tags each measurement as being caused by clutter component 1 or 2, this is the classification part of the measurement, $\boldsymbol{z}_{tr}^{(k)} \in \{1, 2\}$. This tagging procedure is assumed to be perfect. There are two clutter models, $M_Y = 2$, and the event probabilities for target models $m = 3 \dots M$ are given by

$$\beta_t^{mp}(-1) = b_t^m P m_t^{mp} \frac{1 - P v_{t|t-1}^m}{P v_{t|t-1}^m}, \tag{8.1}$$

$$\beta_t^{mp}(0) = b_t^m P m_t^{mp} \left(1 - P d_t^{mp} P s_t^{mp}\right), \tag{8.2}$$

$$\beta_t^{mp}(r) = b_t^m P m_t^{mp} P d_t^{mp} \frac{\tilde{\zeta}_t^{mp}\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^{mp}\right) n_t^c A_t^c}{\zeta_t^c\left(\boldsymbol{z}_{tr}^{(x)} | \boldsymbol{x}_t^c\right) N_t^c I}, \quad c = \boldsymbol{z}_{tr}^{(k)}, \tag{8.3}$$

where b_t^m is a normalising constant, ensuring that $\sum_{r=-1}^{I} \beta_t^{mp}(r) = 1$. The term $P m_t^{mp}$ is the a priori probability for model component p, $P v_{t|t-1}^m$ is the predicted probability that

the target is visible, given data up to the previous scan, Ps_t^{mp} is the probability that the target orientated measurement is within the I nearest, given model p is the correct model, n_t^c is the number of measurements tagged as type $c = \boldsymbol{z}_{tr}^{(k)}$ from those within the I nearest for all model components, N_t^c is the total number of measurements tagged as type $c = \boldsymbol{z}_{tr}^{(k)}$ and A_t^c is the spatial area of clutter model c (provided by the preprocessor). The measurement pdf, $\tilde{\zeta}_t^{mp}\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^{mp}\right)$, is the probability density of the measurement given that it is due to $\boldsymbol{x}_t^{mp}$, but with only an estimate of the state. This is a normal distribution with covariance given by the innovation covariance matrix, S_t^{mp}. Notice that the PDAF uses the innovation covariance to assign measurements, not the measurement variance. This feature was discussed in more detail in chapter 6.

The event probabilities are then used to determine synthetic measurements for each target component model, and a Kalman filter is run for each $\boldsymbol{x}_t^{mp}$. The derivation of the MM-UPDAF event probabilities and state recursions is presented in detail in [Col99].

8.1.2 The Unified Joint PDAF

The MM-UPDAF makes a single target approximation, namely that the assignment of measurements can be performed independently for different target tracks. This assumption is invalid when targets are closely spaced with respect to the sensor resolution. This circumstance may also arise in OTHR when multiple propagation paths are closely spaced. In order to improve the tracking performance of the MM-UPDAF, this single target assumption was relaxed and a Multiple Model Unified Joint PDAF was developed [DC01]. The event probabilities for this filter are found using substantially more complicated expressions than those above, and are not presented here. Since the track initiation comparison uses only isolated targets, multi-target tracking is not required, and for simplicity the UJPDAF is not used.

8.2 JPDAF compared with PMHT

The PMHT solves the tracking problem using an approach different from that used by standard tracking algorithms. This results in an elegant algorithm for multitarget tracking. However, this elegance is not a particularly compelling reason for choosing it over more established tracking approaches. It is important to compare the PMHT with standard algorithms, to see if practical advantage can be gained through its use. Rather than compare PMHT with all of the tracking pantheon, this work focusses on the PDAF, since this algorithm is widely used, and because it is used in the JFAS radar, data from which will be used to provide a realistic evaluation of algorithmic performance. Since the PMHT is by nature a multitarget algorithm, it is sensible to compare it with the JPDAF, or to use single target scenarios. The latter has been chosen because the comparison focusses on track initiation. Multitarget scenarios would unnecessarily complicate the initiation analysis and the results would most likely depend on the particular scenario chosen.

8.2.1 Philosophical Differences

The fundamental philosophical differences between PMHT and JPDAF were discussed in the original presentation of the PMHT [SL95]. When the batch length is one, the PMHT becomes a recursive filter and the two appear superficially similar. Both algorithms can be implemented using probabilistic weights to form synthetic measurements that feed

a Kalman Filter for state estimation. However, the PDAF association probabilities are based on the predicted state, which is a deterministic function of the state estimate at the previous scan. In contrast, the PMHT uses the ML estimate of the current state. [SL95] cites this as a source of bias in the JPDAF.

The JPDAF updates the state with each measurement and then combines the state estimates into a single estimate. The PMHT combines the measurements to form synthetic measurements which are then smoothed.

The JPDAF assumes a one to one assignment of measurements to targets (provided that all targets are detected). The PMHT assumes that each measurement is assigned independently. This effectively means that the PMHT allows the event that more than one measurement is caused by a particular target. The effect of this depends on the validity of these assumptions for the particular data set.

The covariances calculated by the JPDAF reflect the current uncertainty in the state estimate, whereas those computed by the PMHT are not statistically defined in this manner. [SL95] provides a discussion of the interpretation of the PMHT covariance matrices.

8.2.2 Practical Differences

The most obvious differences between the algorithms can be broadly described as implementation issues. The PMHT is a very simple algorithm to implement because of the independence of the assignments. This independence results in an algorithm with computation requirements linearly scaling with the problem size, and which can be implemented in a massively parallel manner. In contrast the JPDAF must deal with an event space whose dimension grows combinatorially with the problem size (the number of targets and measurements within each scan). This combinatorial growth makes the JPDAF a greedy algorithm and the efficient implementation of the JPDAF is an area of research in itself (for example, [ZB95, DC01]). A further issue, besides speed, is the memory required to store the joint event probabilities. In a naive implementation, all the events would be enumerated and their probabilities stored for updating tracks. For challenging scenarios, when multiple dynamic models are used, the number of the joint events can make storing all their probabilities impractical (requiring gigabytes of memory).

A more subtle difference is that the PMHT associates measurements using the target measurement covariance, whereas the JPDAF uses the innovation covariance (which is the expected scatter of the target measurement given the current state estimate and its covariance). This means that the JPDAF gives higher probability to distant measurements than the PMHT does. Also, it gives the JPDAF an ability to dynamically inflate the association process if a track starts to diverge from the true target trajectory. This impacts on the PMHT's ability to handle manoeuvres and heavy clutter.

The algorithms also vary significantly when multiple measurements are sufficiently close to be given high assignment weights. In the JPDAF, this causes the covariance of the state estimate to increase, reflecting the uncertainty about which measurement is the true target measurement (recall that the JPDAF assumes that there can only be one true target measurement). This occurs through the measurement scatter term in (C.7). In the PMHT, each of the measurements is assigned with high probability to the target model. Since there are more measurements of the target, the covariance is reduced. This happens because the sum of the weights for this model is greater than unity and so $\tilde{\mathsf{R}}$ reflects a more accurate measurement than R (namely $|\tilde{\mathsf{R}}| < |\mathsf{R}|$).

8.2.3 Existing Comparison Studies

The existing comparisons of PMHT and (J)PDAF have been mainly orientated at the analysis of steady state tracking performance. The emphasis has been on what Ruan and Willett call "the game of lost-tracks". Namely, given established tracks, what is the probability that the algorithm will maintain the track over a series of scans. In both [RWS95a] and [RW01a], the PMHT algorithm is demonstrated to show similar performance to the JPDAF under easy conditions, but superior performance under difficult conditions.

In [RWS99], the Cramer Rao lower bounds for estimator variance are examined for the PMHT and PDAF measurement models. These models differ due to the different assumptions made about the assignment of measurements. This work showed that the bound for the PMHT is higher than the PDAF. This indicates that a minimum variance unbiased estimator for the PDAF measurement model would have a lower variance than a minimum variance unbiased estimator for the PMHT. Unfortunately, this result does little to illuminate the performance of the PMHT and PDAF algorithms.

8.3 Track Initiation on Simulated Data

In chapter 6, simulated experiments were used to examine the track initiation performance of the PMHT using various initiation schemes. Of these competing approaches, the PMHT-v was found to give the best overall performance. As described earlier in chapter 6, the visibility model, which is a special case of the hysteresis assignment model, is analogous to the model used by the UPDAF for track quality decisions.

The initiation simulations in chapter 6 are now repeated with a simplified version of the UPDAF algorithm. This simplified algorithm needs not deal with multiple clutter components or target dynamic models. The performance of the UPDAF is compared with the PMHT-v performance. Two versions of the PMHT-v are used: the standard batch processor, and a time recursive algorithm. Intuition suggests that batch processing will provide an advantage to performance. Since the UPDAF is a time recursive algorithm, it may seem a little unfair to compare it with a batch approach. However, for the track initiation process, a real time estimate of candidate tracks is not provided to the user. Instead, the candidate is hidden until it passes the promotion test. This means that batch processing could be used for track initiation, even in a system requiring immediate target state estimates for each scan. Both the recursive and batch versions of the PMHT-v are retained to demonstrate the benefit gained by using batch processing.

Three different clutter distributions were considered in chapter 6; uniform clutter, chi-squared clutter, and polynomial clutter. ROC curves for these clutter distributions are shown in figures 8.1, 8.2, and 8.3 respectively. These figures show the high false detection rate clutter examples from chapter 6. Low false detection rate examples are not shown because they give results similar to the high rate, but with reduced differences between the curves.

An interesting feature of the ROC curves is that sometimes the recursive PMHT-v gives a higher probability for promoting valid tracks than the batch PMHT-v for a particular probability of promoting false tracks. This occurs because the ROC is a normalised picture. The recursive PMHT-v has a much higher rate of divergent candidates than the batch PMHT-v (see the following section). The tracks which diverge in the recursive PMHT-v are invariably those which are assigned a low probability of being visible by the batch PMHT-v. This means that the average quality of tracks in the PMHT-v is reduced, and the ROC is therefore effected.

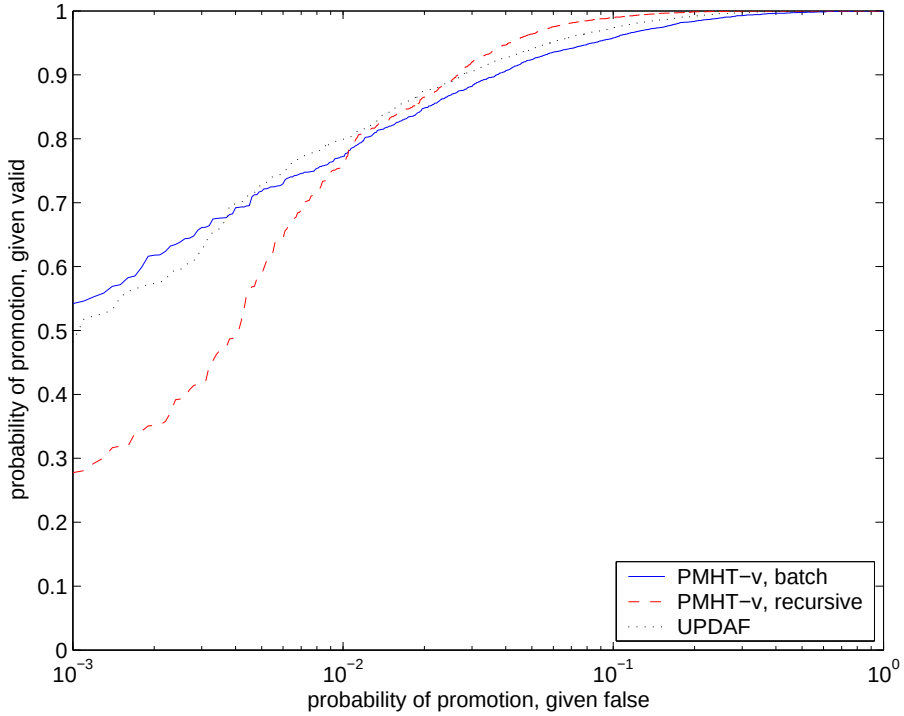


Figure 8.1: Uniform Clutter, high density

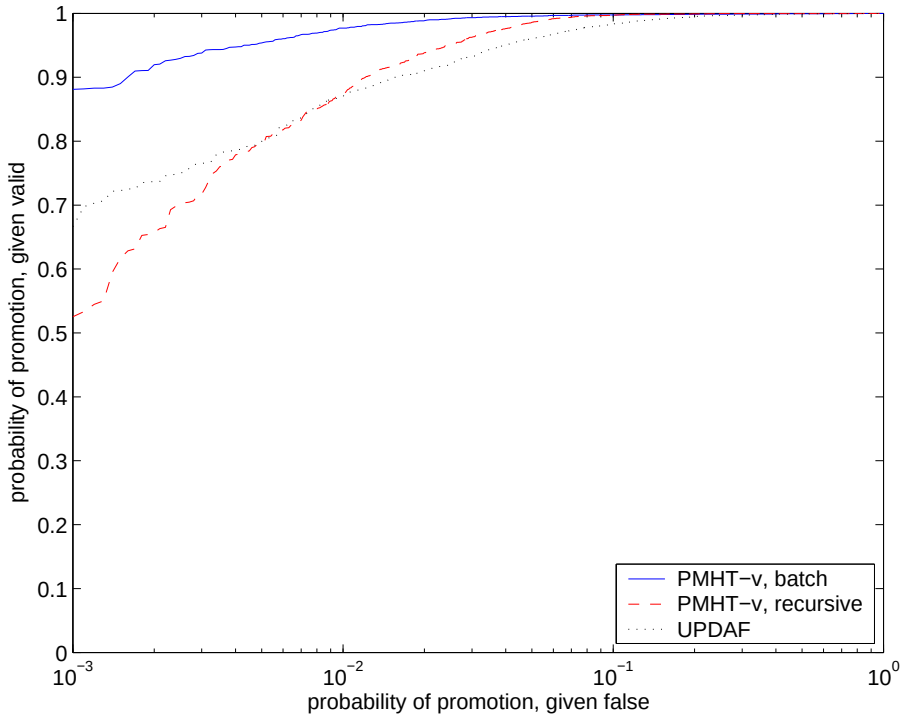


Figure 8.2: Chi-Squared Clutter, high density

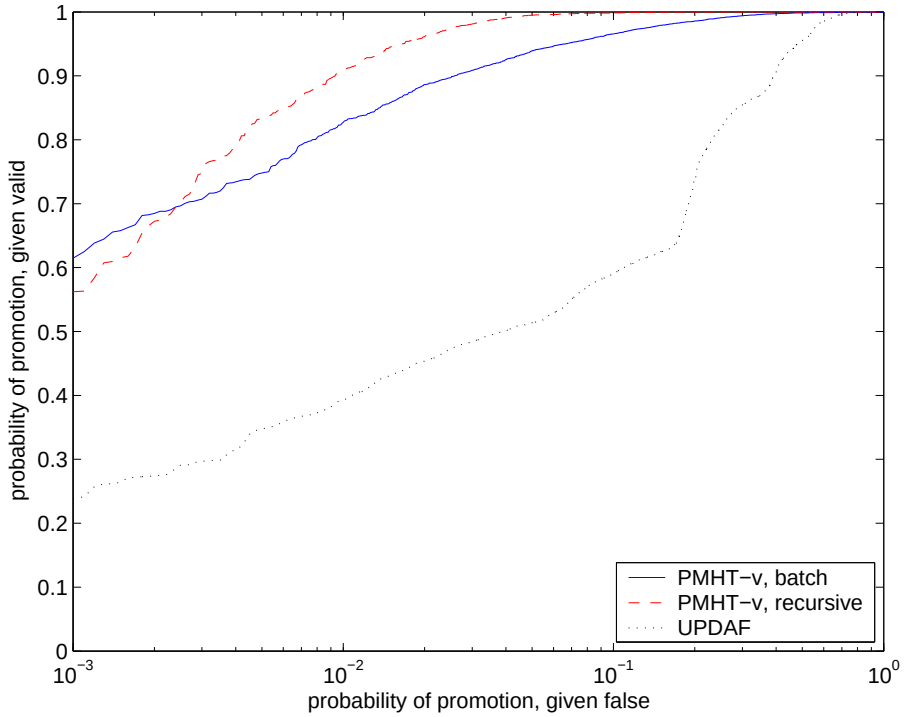


Figure 8.3: Polynomial Clutter, high density

On the uniformly distributed clutter, all give similar performance, except for very low probability of promoting a false track, where the recursive PMHT-v drops off. For chi-squared distributed clutter, recursive PMHT-v and UPDAF give similar performance, with batch PMHT-v significantly better. For the polynomial distributed clutter, the PMHT-v batch is slightly better than the recursive version, and the UPDAF gives very poor performance. The UPDAF fails in this last case because of the high concentration of false detections at low measurement values. Although the distribution itself is non-physical, the primary difficulty in OTHR radar data (for initiation at least) is highly concentrated false detections due to phenomenon such as range folder clutter.

Overall, the batch PMHT-v is clearly the best track initiation approach in these simulated examples.

8.3.1 Initialisation Robustness

Table 8.1 lists the number of divergent trials for each of the scenarios. As a general rule, the batch PMHT-v gives a much lower rate of divergent tracks. The recursive PMHT-v and the UPDAF give similar results, with the recursive PMHT-v slightly worse. The recursive PMHT-v does particularly poorly in the high false alarm rate, uniform clutter example. These results serve to highlight the advantage of using batch processing, which is not a viable option with the UPDAF. The similarity between the recursive PMHT-v and the UPDAF results indicates that the innovation homothetic model has done an acceptable job of making the PMHT robust to initialisation errors.

clutter type	uniform		chi-squared		polynomial	
false alarm rate	low	high	low	high	low	high
batch PMHT	619	2346	292	563	340	901
recursive PMHT	1655	3460	975	1574	1177	1999
UPDAF	1336	2511	850	1238	1017	1700

Table 8.1: Number of divergent trials

8.4 Track Initiation on JFAS Data

As with the earlier track initiation analysis, JFAS data is now used to support the simulation findings. The UPDAF algorithm is the current tracking algorithm for the JFAS radar. This algorithm reports a track quality measure referred to as track *confidence* which is derived from the posterior visibility probability. To produce empirical ROC curves, the confidence of candidate tracks is reported after a fixed number of scans. This is the same as the batch length used for the PMHT algorithm, namely 8 scans.

The ROC curve derived for the JFAS UPDAF tracker is shown in figure 8.4 with PMHT-v curves for batch and recursive processing. In chapter 7, a number of different approaches for modelling clutter were tested. The PMHT-v curve in 8.4 is derived using the degenerate PMHT-c filter which assumes that the clutter classification information is perfect. This is the same philosophy as that adopted by the UPDAF, so it should be used for a fair comparison.

It can be seen that the PMHT-v performance is superior, particularly when the false track promotion probability is low. The recursive PMHT-v algorithm gives worse performance than the batch PMHT-v. The UPDAF shows a sudden drop off in performance, resulting in an effective minimum false track promotion probability below which detection is impossible. This occurs because the track quality measure for the UPDAF is bounded between 0 and 100. To achieve very low false track rates, the promotion threshold must be increased until it becomes very close to 100 where all tracks are rejected. In essence this means that some false tracks can never be suppressed with UPDAF. The sharpness of the transition occurs because most of the UPDAF tracks formed on strong targets have the same (or very similar) confidence values. In contrast, the PMHT-v is able to suppress all of the false candidate tracks and still promote a high proportion of the valid candidates.

8.4.1 Initialisation Robustness

In each of the initiation comparisons performed in earlier parts of this thesis, the analysis has included ROC curves and tables of initialisation robustness. The initialisation robustness has been quantified by a count of the number of times a candidate *that was initialised on a valid detection* diverged from the true target trajectory. To analyse this quantity for real data, would require accurate knowledge of whether each target was detected or not for each scan. Acquiring this knowledge would be very time consuming, and it has not been done. Consequently, the initialisation robustness on real data is not considered.

8.5 Established Track Performance

The focus of the work in this thesis has been track initiation. However, this is only one aspect of tracking performance. There is also a raft of tracking performance metrics that

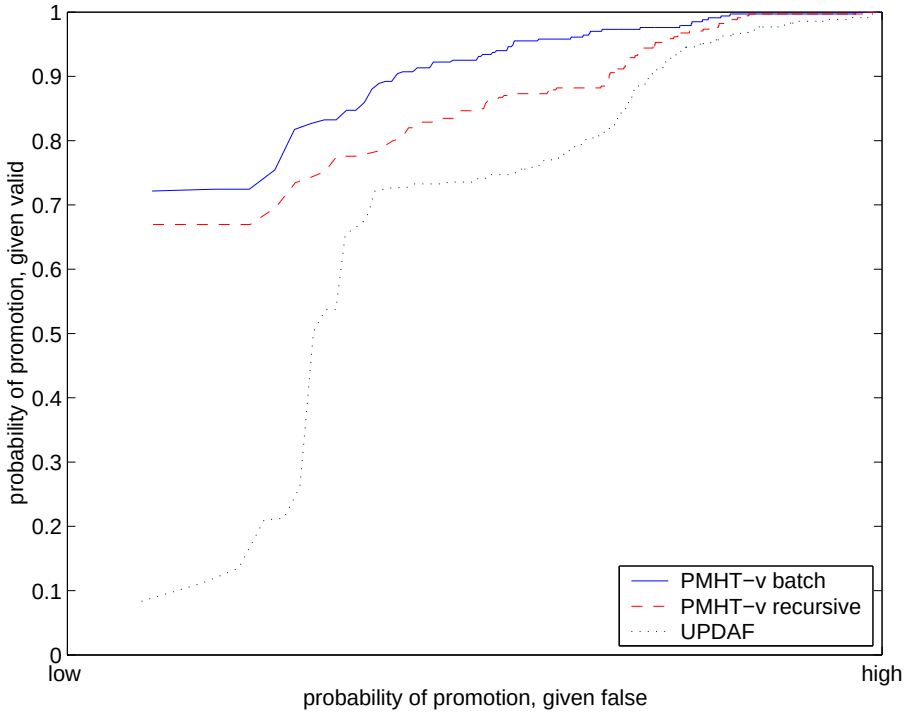


Figure 8.4: ROC curves for JFAS data

can be proposed for established track performance. For example, [CDD96] lists numerous metrics which include track initiation performance as well as estimation accuracy and multitarget correctness measures. Consideration of these is beyond the scope of this thesis, as is any further PMHT development that may be required to address application specific problems in established tracking.

8.6 Summary

This chapter has presented a comparison of the track initiation performance of the PMHT-v and UPDAF algorithms. The UPDAF is the algorithm currently used for tracking in the JFAS radar. The PMHT-v was implemented in both batch and time recursive forms. For a variety of simulated clutter conditions, the batch PMHT-v was found to give better track initiation performance than the UPDAF, and to recover better from poor initialisation when the innovation homothetic initialisation scheme was used. The recursive PMHT-v gave comparable performance to the UPDAF, except in one example where the UPDAF performed very poorly.

These algorithms were also compared using JFAS radar data. In this case, the UPDAF used was the operational radar tracking algorithm, a result of considerable research effort. The track initiation performance of the PMHT-v was found to be superior in both recursive and batch forms, with the batch PMHT-v giving the best results.

Chapter 9

Summary

This thesis has introduced a number of enhancements to the PMHT algorithm, motivated by the Over the Horizon Radar tracking problem. The primary two enhancements are the incorporation of classification information, and the introduction of a discrete state model for the assignment prior probability. The modified PMHT algorithms achieved through these enhancements are referred to as PMHT-c and PMHT-y respectively.

The benefit of the PMHT-c and PMHT-y algorithms has been demonstrated through simulation studies, and through the application of both to recorded OTHR data.

9.1 Classification Measurements

The standard PMHT algorithm deals with measurements that are observations of the state of a particular model. The algorithm provides state estimates by determining the probability of an assignment index that links measurements with the model that caused them. The PMHT-c is an extension of the PMHT that deals with the more general problem where the tracker is also provided with measurements of the assignment index itself. These measurements are classifications, and are assumed to be independent of the model state, and hence the normal measurements. The PMHT-c simplifies to the standard PMHT if the classification measurements are uninformative.

The PMHT-c algorithm has been shown to improve state estimation accuracy, and manoeuvre tracking, by using extra measurement information to increase the probability of the true assignment index. The probability mass function of the classification measurements can be represented as a matrix and is referred to as the confusion matrix. The PMHT-c was derived under the two conditions of known and unknown confusion matrix. Simulations demonstrated that the PMHT-c is not sensitive to the confusion matrix values if it is assumed to be known. Further, in the experiments conducted, it was seen to be advantageous to assume an artificially high probability of correct classifications.

9.2 Dynamic Assignment Prior

The second extension to the PMHT algorithm was the addition of a discrete Markov process as an underlying dynamic model for the assignment prior probability. Under the standard PMHT model, the assignments are independent identically distributed random variables drawn from a possibly unknown prior probability mass. If this mass is unknown, the PMHT allows for its estimation, but only under the restrictive assumption that it is

either independent between scans, or identical. This assumption was relaxed by introducing a state model for the assignment prior, referred to as the hysteresis model. The only restriction on the assignment state model was that the state was required to be a discrete random variable. This restriction was imposed to enable the optimal solution of the estimation of the assignment state, or its probability. A continuous assignment state could be used, but this would require an approximate solution method, such as particle filtering or a Taylor Series analytic approximation. This thesis dealt with the situation where the statistics of the assignment state are known, however methods were identified for the case where these statistic are unknown.

Two algorithms were derived under this generalised assignment model: the PMHT-ym was derived by treating the assignment state as missing data. The resulting algorithm calculates the probability mass of the assignment state, similar to the way the standard PMHT calculates the assignment probability mass. The PMHT-ym algorithm can be implemented using an iterative scheme alternating between a bank of parallel Kalman Smoothers for model state estimation, and a joint Hidden Markov Model Smoother to determine the probability of the assignment state.

The PMHT-ye was derived by treating the assignment state as a quantity to be estimated. This means that the assignment state is treated as an extension of the target model states, and is iteratively estimated along with the target dynamic states. In this case, the assignment state is estimated using a joint Viterbi algorithm, run in parallel with the Kalman Smoother bank for dynamic state estimation.

Each of the algorithms derived under the Hysteresis model requires joint target model processing for part of the algorithm. This is a result of the normalisation across models in the assignment prior and can only be avoided through making approximations. This means that both the PMHT-ym and PMHT-ye contain steps with exponential complexity in the number of targets.

Simulations were used to demonstrate how both the PMHT-ym and PMHT-ye algorithms were able to give much better estimation accuracy for the assignment prior when the prior was smoothly varying with time. This improvement was observed to be a result of the Hysteresis model allowing the PMHT to incorporate data from adjacent scans to estimate the assignment prior.

9.3 Track Initiation

The problem of track initiation was addressed for PMHT: under the standard PMHT assumptions, the number of models is fixed and known, and prior information is available for the models. This is rarely the case in practice. An approach for initialisation based on the homothetic measurement model was introduced and demonstrated to greatly improve the robustness of PMHT to assumptions about the state prior. A dynamic number of target models was achieved by introducing methods for automatic track initiation. This was achieved by over modelling the system using a number of candidate models, and discarding redundant candidate models based on a significance test. Different test statistics were considered, including those based on intuition, Model Order Estimation, and the application of the Hysteresis model.

It was shown that the *visibility* model used for track initiation with PDAF and other tracking algorithms was a special case of the Hysteresis model where the assignment state is assumed to be binary. The PMHT-ym algorithm incorporating visibility was referred to as the PMHT-v.

Simulations were used to investigate the performance of the proposed initiation methods on a variety of clutter conditions. It was found that the PMHT-v algorithm gave the best overall initiation performance, and gave particular advantage when the probability density function of the clutter was non-uniform.

9.4 Radar Data Performance

In order to verify the simulation results, and because of a perceived need for a true multitarget tracking algorithm for OTHR, the PMHT and the enhancements above were implemented for data from the JFAS radar. This implementation required the tracker to also address a number of practical problems with the JFAS data. The two main issues were the JFAS clutter modelling approach and the problem of unwrapping ambiguous doppler measurements.

Several approaches were identified for addressing the doppler unwrapping problem. These included modelling the true doppler frequency as an unknown track parameter, treating it as missing data in an EM context, and using a non-physical mixture model which modelled the target as a superposition of all possible unwrapped dopplers. It was found that the unknown parameter approach gave the best results.

The clutter modelling problem was addressed by applying the PMHT-c to the JFAS data. The JFAS pre-track processing produces feature information which is used in a classifier to segment the measurements into different classes. The PMHT-c was used to incorporate this classification information, and to allow for classification errors. An alternative approach whereby the tracker was given direct access to the feature information was also identified. These approaches were tested using simulated data and using recorded JFAS data. It was found that the best performance was obtained by removing the classifier and giving the tracker direct access to the feature measurements.

The proposed track initiation approaches were applied to recorded JFAS data and the performance of each observed. As with the simulated results, it was found that the PMHT-v, which is a special case of the Hysteresis dynamic assignment model, gave the best performance.

9.5 Comparison with PDAF

The enhancements made to the PMHT enabled it to deal with a broader range of data conditions. However, the problems solved by the new PMHT variants have already been addressed using other tracking algorithms. In order to gauge the usefulness of these new variants, the PMHT performance was compared with that of the Unified PDAF, the algorithm currently used for tracking with the JFAS radar. Track initiation performance was compared based on simulated data and on recorded radar data. It was found that the PMHT-v algorithm gave better performance than the UPDAF.

9.6 Future Work

As with any research, this thesis leaves areas which might be explored by future work.

9.6.1 Classification Information

An early assumption in the PMHT-c formulation is that the classification measurements are independent of the state, and the state measurements. This assumption may not always be valid. In some applications, the classifier output may be dependent on the state. For example, a ship classifier may use a length estimate as a feature. The accuracy of the estimate, and hence the classification accuracy, would be dependent on the ship's aspect angle and range. These are components of the dynamic state. It is reasonably straightforward to use a state dependent confusion matrix for measurement association, however under this circumstance, the classification measurement provides state information. The proper incorporation of this information in the state optimisation requires further investigation.

The analysis of the PMHT-c in this thesis has considered how a classifier can be used to enhance state estimation. The complementary problem is the influence of state estimates on the classifier output. Further, the classifier and tracker could be tightly coupled, such as in [CP01] to improve both by cross fertilisation.

9.6.2 Prior Dynamics

The Hysteresis model presented in this thesis was solved under the assumption of a discrete assignment state. This might be viewed as an approximation to a continuous assignment state, using a fixed grid method. Alternative methods could be used, such as particle filtering.

An obvious shortcoming of the PMHT-ve and PMHT-vm algorithms is that both have exponential complexity in the number of targets. This is disappointing, since one of the advantages of the standard PMHT over other tracking approaches is its linear computational growth. The exponential complexity is an inevitable result of the model coupling which occurs due to normalisation of the prior. However, with further research, it may be possible to achieve similar performance with an approximate algorithm that has reduced complexity.

9.6.3 Track Initiation and Initialisation

The Innovation Homothetic Model presented in this thesis is an ad hoc approach for tackling the PMHT initialisation problem. In its use, no consideration was given to issues such as convergence. There are many other methods that might be used, and initialisation remains an open question for PMHT.

While the track initiation research presented in this thesis was underway, other parallel work on initiation with PMHT was done using a Hough Transform based approach [LSW01]. Future comparisons need to be performed to ascertain the relative merits of this approach compared with PMHT-v.

9.6.4 OTHR Implementation

The prototype OTHR algorithm developed in this thesis has been demonstrated to give good performance for track initiation. However, there are numerous other aspects of tracking which need to be addressed before the algorithm would be appropriate for operational evaluation. There are many problems in established tracking which have already been addressed with the UPDAF, and some of these solutions may be incorporated into the PMHT algorithm.

9.6.5 PMHT and PDAF Comparison

The comparison of PMHT and PDAF presented in this thesis considered only the problem of track initiation. While other studies have compared aspects of established track performance, an overall comparison between the two algorithms would be of interest. This is especially true of the recorded radar data comparison.

Through the implementation of the JFAS mixture model for clutter, evidence has been found that the PMHT is less sensitive to the assumed density functions than PDAF. If this is the case, it is an important fundamental property of the algorithms, and could be further investigated through simulation studies.

9.7 Conclusions

This thesis has presented new extensions to the Probabilistic Multi-Hypothesis Tracker, motivated by the Over the Horizon Radar tracking problem. Some of the research has provided solutions to application focused problems, such as track initiation and clutter modelling. Other work has given rise to fundamental generalisations of the PMHT algorithm to allow it to properly deal with a wider range of measurement processes.

Appendix A

Innovation Homothetic Model for Initialisation Robustness

This appendix presents details of the initialisation approach introduced in chapter 6, which uses the innovation covariance matrix as a homothetic measurement pdf. The use of the innovation covariance matrix allows the second homothetic model to effectively expand its assignment gate when the state initialisation is poor.

The Homothetic PMHT presented in [RWS95a] makes use of a scaled measurement covariance matrix for the homothetic model. This has the convenience that the covariance of the mixture components for each target model are simply scalar multiples of each other. The result is that the EM auxiliary function is maximised by a Kalman filter that uses the measurement covariance modified by a scalar factor. In general, the innovation covariance will not be a scalar multiple of the measurement covariance since it also includes the covariance of the current state estimate. Recall that the innovation covariance is given by

$$\mathsf{S}_t^m = \mathsf{H}_t^m \mathsf{P}_{t|t-1}^m \mathsf{H}_t^{m\mathsf{T}} + \mathsf{R}_t^m \tag{A.1}$$

When the innovation covariance is used as a second homothetic measurement model, the resulting solution will not be a Kalman smoother using a scalar multiple of the measurement covariance.

The solution for the homothetic measurement model where the second component of the mixture density has an arbitrary covariance matrix now follows. Assume that the two mixture components are not equal components of the mixture, but that they have unknown mixing proportions which may be a function of time. This is a generalisation of the homothetic mixture model first used in [RWS95a]. The extension of the homothetic model to use arbitrary covariance matrices is known [Str00a, Lug01], but has not been published. Having derived this PMHT, the extension to a mixture of more than two arbitrary covariance components will be obvious.

The missing data are the assignment indices $\mathbf{K}$. As introduced in section (3.5.1), the homothetic mixture model uses two index lists. For measurement $\boldsymbol{z}_{tr}$, the index k_{tr}^1 takes a value in the range $1 \dots M$ and indicates the model that caused the measurement. The index k_{tr}^2 takes either the value 1 or 2, indicating the component of the homothetic mixture for model k_{tr}^1 that caused the measurement. So,

$$p\left(\boldsymbol{z}_{tr} | \mathbf{X}_t, k_{tr}^1, k_{tr}^2\right) = \zeta_t^{k_{tr}^1, k_{tr}^2}\left(\boldsymbol{z}_{tr} | \boldsymbol{x}_t^{k_{tr}^1}\right) \tag{A.2}$$

where the mixture components $\zeta_t^{k_{tr}^1, k_{tr}^2}$ are Gaussian distributions given by

$$\zeta_t^{m1}\left(\boldsymbol{z}_{tr} | \boldsymbol{x}_t^m\right) \sim N\left(\mathsf{H}_t \boldsymbol{x}_t^m, \mathsf{R}_t^m\right) \tag{A.3}$$

$$\zeta_t^{m2}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) \sim N\left(\mathsf{H}_t\boldsymbol{x}_t^m, \mathsf{S}_t^m\right) \tag{A.4}$$

For each target at each scan, introduce the mixing proportions ρ_t^{mp} so that the prior probability mass of the indices is given by

$$P\left(k_{tr}^1 = m\right) \equiv \pi_t^m \tag{A.5}$$
$$P\left(k_{tr}^2 = p|k_{tr}^1 = m\right) \equiv \rho_t^{mp} \tag{A.6}$$

The mixing proportions are constrained by $\rho_t^{m1} + \rho_t^{m2} = 1$ and for each clutter model $\rho_t^{m1} = 1$.

Also define temporal sets as follows

$$\rho_t \equiv \left[\rho_t^{11}, \rho_t^{12}, \rho_t^{21}, \ldots \rho_t^{M2}\right] \tag{A.7}$$

and

$$\rho \equiv [\rho_1, \rho_2, \ldots, \rho_T] \tag{A.8}$$

Since the observer and measurement model have been modified, the complete data likelihood is modified. The conditional probability of the received measurements is given by

$$P\left(\mathbf{Z}|\mathbf{K}, \mathbf{X}\right) = \prod_{t=1}^{T}\prod_{r=1}^{n_t} \zeta_t^{k_{tr}^1 k_{tr}^2}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}^1}\right) \tag{A.9}$$

The probability of the assignments is given by

$$P\left(\mathbf{K}\right) = \prod_{t=1}^{T}\prod_{r=1}^{n_t} \pi_t^{k_{tr}^1}\rho_t^{k_{tr}^1 k_{tr}^2} \tag{A.10}$$

Thus the new complete data likelihood is given by

$$L(\mathbf{O}, \mathbf{Z}) = \prod_{m=1}^{M}\left\{\psi_0^m\left(\boldsymbol{x}_0^m\right)\prod_{t=1}^{T}\psi_T^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)\right\}\prod_{t=1}^{T}\prod_{r=1}^{n_t}\pi_t^{k_{tr}^1}\rho_t^{k_{tr}^1 k_{tr}^2}\zeta_t^{k_{tr}^1 k_{tr}^2}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}^1}\right) \tag{A.11}$$

The conditional probability of the assignments is also modified. There are now two weights for each track and measurement because of the two mixture components in the target measurement density. The new conditional probability of the assignments is thus

$$P(\mathbf{K}|\mathbf{X}, \mathbf{Z}) = \prod_{t=1}^{T}\prod_{r=1}^{n_t} w_{k_{tr}^1 k_{tr}^2 tr} \tag{A.12}$$

where the weight $w_{k_{tr}^1 k_{tr}^2 tr}$ is given by

$$w_{k_{tr}^1 k_{tr}^2 tr} = \frac{\pi_t^{k_{tr}^1}\rho_t^{k_{tr}^1 k_{tr}^2}\zeta_t^{k_{tr}^1 k_{tr}^2}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}^1}\right)}{\sum_{m=1}^{M}\sum_{p=1}^{2}\pi_t^m\rho_t^{m,p}\zeta_t^{m,p}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right)} \tag{A.13}$$

Equations (A.11) and (A.12) are now substituted into auxiliary function to obtain

$$\begin{aligned} &Q\left(\mathbf{X},\mathbf{\Pi},\rho|\mathbf{X}^{(i)},\mathbf{\Pi}^{(i)},\rho^{(i)}\right) = \\ &\sum_{\mathbf{K}}\left\{\sum_{m=1}^{M}\left\{\log\psi_0^m\left(\boldsymbol{x}_0^m\right)+\sum_{t=1}^{T}\log\psi_T^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)\right\}\right\}\prod_{t=1}^{T}\prod_{r=1}^{n_t}w_{k_{tr}^1k_{tr}^2tr}^{(i)}+ \\ &\sum_{\mathbf{K}}\left\{\sum_{t=1}^{T}\sum_{r=1}^{n_t}\log\pi_t^{k_{tr}^1}\right\}\prod_{t=1}^{T}\prod_{r=1}^{n_t}w_{k_{tr}^1k_{tr}^2tr}^{(i)}+ \\ &\sum_{\mathbf{K}}\left\{\sum_{t=1}^{T}\sum_{r=1}^{n_t}\log\rho_t^{k_{tr}^1k_{tr}^2}\right\}\prod_{t=1}^{T}\prod_{r=1}^{n_t}w_{k_{tr}^1k_{tr}^2tr}^{(i)}+ \\ &\sum_{\mathbf{K}}\left\{\sum_{t=1}^{T}\sum_{r=1}^{n_t}\log\zeta_t^{k_{tr}^1k_{tr}^2}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{k_{tr}(i)}\right)\right\}\prod_{t=1}^{T}\prod_{r=1}^{n_t}w_{k_{tr}^1k_{tr}^2tr}^{(i)} \end{aligned} \tag{A.14}$$

where the dependence on the $.^{(i-1)}$ terms is implicit in the weight $w_{\kappa_{tr}k_{tr}tr}^{(i-1)}$.

In a similar way as for the standard PMHT, rewrite (A.14) as

$$Q\left(\mathbf{X},\mathbf{\Pi},\rho|\mathbf{X}^{(i)},\mathbf{\Pi}^{(i)},\rho^{(i)}\right)=\sum_{m=1}^{M}Q_X^m+\sum_{t=1}^{T}Q_{t\pi}+\sum_{t=1}^{T}\sum_{m=1}^{M}Q_{t\rho}^m \tag{A.15}$$

The Q_X^m term in (A.15) depends only on the target model states and the measurements and is given by

$$Q_X^m=\log\psi_0^l\left(\boldsymbol{x}_0^m\right)+\sum_{t=1}^{T}\log\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)+\sum_{t=1}^{T}\sum_{r=1}^{n_t}\sum_{p=1}^{2}\log\zeta_t^{mp}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right)w_{mptr}^{(i)} \tag{A.16}$$

The $Q_{t\pi}$ term in (A.15) depends only on the prior probability Π and is given by

$$Q_{t\pi}=\sum_{r=1}^{n_t}\sum_{k=1}^{M}\log\pi_t^k\left(w_{k1tr}^{(i)}+w_{k2tr}^{(i)}\right) \tag{A.17}$$

The $Q_{t\rho}^m$ term in (A.15) depends only on the prior probability ρ and is given by

$$Q_{t\rho}^m=\sum_{r=1}^{n_t}\sum_{p=1}^{2}\log\rho_t^{mp}w_{mptr}^{(i)} \tag{A.18}$$

A.1 Maximisation Step

Since it is the simplest, start by maximising the $Q_{t\pi}$ term in (A.15). Notice that the expression for $Q_{t\pi}$ is almost the same as that found for the standard PMHT whose solution is given in (3.51). By substituting $w_{k1tr}^{(i)}+w_{k2tr}^{(i)}$ for $w_{ktr}^{(i)}$ in (3.51), the estimate for π_t^k at iteration i is given by

$$\pi_t^{k(i+1)}=\frac{1}{n_t}\sum_{r=1}^{n_t}w_{k1tr}^{(i)}+w_{k2tr}^{(i)} \tag{A.19}$$

The $Q_{t\rho}^m$ term in (A.15) is maximised in a similar way to the $Q_{t\pi}$ term. $Q_{t\rho}^m$ is to be maximised subject to the constraint $\rho_t^{m1}+\rho_t^{m2}=1$. This is achieved by using the Lagrangian

$$L_\rho(m,t)=\sum_{r=1}^{n_t}\sum_{p=1}^{2}\log\rho_t^{mp}w_{mptr}^{(i)}+\lambda_{m,t}^{\rho}\left(1-\sum_{k=1}^{2}\rho_t^{mk}\right) \tag{A.20}$$

where $\lambda^{\rho}_{m,t}$ is the Lagrange multiplier.

Differentiating the Lagrangian with respect to ρ_t^{mp} and solving for stationary points gives

$$\rho_t^{mp(i+1)} = \frac{1}{\lambda^{\rho}_{m,t}} \sum_{r=1}^{n_t} w^{(i)}_{mptr} \tag{A.21}$$

Reapplying the constraint gives

$$\lambda^{\rho}_{m,t} = \sum_{r=1}^{n_t} \sum_{p=1}^{2} w^{(i)}_{mptr} \tag{A.22}$$

So the estimate for ρ_t^{mp} at iteration i is given by the unique stationary point of the Lagrangian

$$\rho_t^{mp(i+1)} = \frac{\sum_{r=1}^{n_t} w^{(i)}_{mptr}}{\sum_{r=1}^{n_t} \left(w^{(i-1)}_{m1tr} + w^{(i)}_{m2tr} \right)} \tag{A.23}$$

The estimate of ρ_t^{mp} is thus the proportion of weights for that measurement model at scan t.

A.2 Target State Maximisation

A modified Kalman filter algorithm is now derived for the target state update following the approach used in [SL95].

Recall that (A.4) defines ζ_t^{m1} and ζ_t^{m2} as

$$\begin{aligned} \zeta_t^{m1}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) &\sim N\left(\mathsf{H}_t\boldsymbol{x}_t^m, \mathsf{R}_t\right) \\ \zeta_t^{m2}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) &\sim N\left(\mathsf{H}_t\boldsymbol{x}_t^m, \mathsf{S}_t\right) \end{aligned}$$

and that (A.16) gives Q_X^m as

$$Q_X^m = \log \psi_0^l\left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T} \log \psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) + \sum_{t=1}^{T} \sum_{r=1}^{n_t} \sum_{p=1}^{2} \log \zeta_t^{mp}\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m\right) w^{(i)}_{mptr}$$

Consider the measurement dependent part of Q_X^m at scan t, namely

$$\sum_{r=1}^{n_t} \sum_{p=1}^{2} \log \zeta^p\left(\boldsymbol{z}_r|\boldsymbol{x}\right) w_{pr}$$

where the scan index t, the model index m and the iteration index i are suppressed for simplicity of notation.

Substituting the ζ^p definitions from (A.4) write

$$\begin{aligned}\sum_{r=1}^{n_t}\sum_{p=1}^{2}\log\zeta^p\left(\boldsymbol{z}_r|\boldsymbol{x}\right)w_{pr} &= \sum_{r=1}^{n_t} w_{1r}(\boldsymbol{z}_r-\mathsf{H}\boldsymbol{x})^{\mathsf{T}}\mathsf{R}^{-1}\left(\boldsymbol{z}_r-\mathsf{H}\boldsymbol{x}\right)+\\ &\qquad \sum_{r=1}^{n_t} w_{2r}(\boldsymbol{z}_r-\mathsf{H}\boldsymbol{x})^{\mathsf{T}}\mathsf{S}^{-1}\left(z_r-Hx\right)+A\\ &= \sum_{r=1}^{n_t}\left(\boldsymbol{z}_r-\mathsf{H}\boldsymbol{x}\right)^{\mathsf{T}}\left\{w_{1r}\mathsf{R}^{-1}+w_{2r}\mathsf{S}^{-1}\right\}\left(\boldsymbol{z}_r-\mathsf{H}\boldsymbol{x}\right)+A \qquad \text{(A.24)}\end{aligned}$$

where A is independent of the target state $\boldsymbol{x}_t^m$ and can be ignored.

Expanding the quadratic terms and collecting all state independent expressions into the constant A gives

$$\begin{aligned}\sum_{r=1}^{n_t}\sum_{p=1}^{2}\log\zeta^p\left(\boldsymbol{z}_r|\boldsymbol{x}\right)w_{pr} &= \left(\mathsf{H}\boldsymbol{x}\right)^{\mathsf{T}}\left\{\sum_{r=1}^{n_t} w_{1r}\mathsf{R}^{-1}+\sum_{r=1}^{n_t} w_{2r}\mathsf{S}^{-1}\right\}\left(\mathsf{H}\boldsymbol{x}\right)-\\ &\qquad \sum_{r=1}^{n_t}\boldsymbol{z}_r{}^{\mathsf{T}}\left(w_{1r}\mathsf{R}^{-1}+w_{2r}\mathsf{S}^{-1}\right)\left(\mathsf{H}\boldsymbol{x}\right)-\\ &\qquad \left(\mathsf{H}\boldsymbol{x}\right)^{\mathsf{T}}\sum_{r=1}^{n_t}\left(w_{1r}\mathsf{R}^{-1}+w_{2r}\mathsf{S}^{-1}\right)\boldsymbol{z}_r+A \qquad \text{(A.25)}\end{aligned}$$

Complete the square, giving

$$\sum_{r=1}^{n_t}\sum_{p=1}^{2}\log\zeta^p\left(\boldsymbol{z}_r|\boldsymbol{x}\right)w_{pr}=\left(\tilde{\boldsymbol{z}}-\mathsf{H}\boldsymbol{x}\right)^{\mathsf{T}}\tilde{\mathsf{R}}^{-1}\left(\tilde{\boldsymbol{z}}-\mathsf{H}\boldsymbol{x}\right)+A \qquad \text{(A.26)}$$

where

$$\begin{aligned}\tilde{\mathsf{R}}^{-1} &= \sum_{r=1}^{n_t} w_{1r}\mathsf{R}^{-1}+\sum_{r=1}^{n_t} w_{2r}\mathsf{S}^{-1} \qquad \text{(A.27)}\\ &= \sum_{r=1}^{n_t}\left(w_{1r}^{(i)}+w_{2r}^{(i)}\right)\left\{\rho^{1(i+1)}\mathsf{R}^{-1}+\rho^{2(i+1)}\mathsf{S}^{-1}\right\} \qquad \text{(A.28)}\end{aligned}$$

and

$$\tilde{\boldsymbol{z}}=\tilde{\mathsf{R}}\left\{\sum_{r=1}^{n_t}\left(w_{1r}\mathsf{R}^{-1}+w_{2r}\mathsf{S}^{-1}\right)\boldsymbol{z}_r\right\} \qquad \text{(A.29)}$$

The result in (A.26) is the same expression as that maximised by a Kalman smoother: the log likelihood of a measurement $\tilde{\boldsymbol{z}}$ with measurement covariance $\tilde{\mathsf{R}}$. Thus (A.28) and (A.29) define a *synthetic* measurement at each time point for each target model and give the covariance of the measurement. The maximum likelihood state estimate can be found using a Kalman smoother. The results in (A.28) and (A.29) simplify to the homothetic synthetic measurements given in [RWS95a] when the innovation covariance is a scalar multiple of the measurement variance and to the standard PMHT when there is only a single homothetic model.

Appendix B

Full Radar Algorithm

Chapters 4, 5 and 6 provided a number of theoretical advancements of the PMHT algorithm, and chapter 7 presented solutions to particular OTHR data issues. All of these algorithmic extensions have been presented in isolation. Realistically, it is desirable to incorporate many of these advances into a single algorithm for the radar data. The derivation of such an algorithm is essentially a matter of careful notational accounting and is not presented here. This is because of the amenable structure of the PMHT framework. Instead, an example of one of the algorithms run on radar data is presented.

This algorithm incorporates:

- the PMHT-c approach for clutter parameterisation with estimated classifier confusion matrix
- the innovation homothetic model for initialisation
- the PMHT-v approach for track initiation
- the mixture approach for ambiguous velocity resolution

B.1 Statement of Algorithm

1. The first step with all PMHT variants is initialisation. There are several different variables that need to be initialised for this algorithm.
   (a) Clutter model states, $\mathbf{X}^{1(0)}$ and $\mathbf{X}^{2(0)}$: mean of amplitude distribution
   (b) Clutter mixing proportions, $\sigma_t^{m(0)} = 0.5$
   (c) Classifier veracity, $\alpha_1^{(0)}$ and $\alpha_2^{(0)}$
   (d) Target model states, $\mathbf{X}^{3(0)}, \ldots \mathbf{X}^{M(0)}$
   (e) Target model visibilities, $P\left(d_0^m = 1\right)^{(0)} = 0.1\rfloor$ for all m.
   (f) Target homothetic measurement component mixing proportions, $\rho_t^{mp(0)} = 0.5$ for all target models m, for all scans t and all dynamic components, p.
   (g) Target dynamic component mixing proportions, $P_\mu{}^m(p) = M_\mu^{-1}$
2. Calculate the assignment weights for each model (component model) and measurement. The expressions for target model weights and clutter model weights are different because the target models are substantially more sophisticated than the clutter

models. Let $\tilde{w}$ denote a pre-normalised assignment weight. The pre-normalised assignment weights are given by

(a) For clutter model $m \in 1, 2$ and measurement $\boldsymbol{z}_{tr}$

$$\begin{aligned} \tilde{w}_{mtr} &= p\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i)}\right) \\ &= \sigma_t^{m(i)} c_{z_{tr}^{(x)}m} \zeta_t^m \left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{m(i)}\right). \end{aligned} \tag{B.1}$$

(b) For homothetic measurement component k of dynamic component p of target model $m \in 3, \ldots M$ and measurement $\boldsymbol{z}_{tr}$

$$\begin{aligned} \tilde{w}_{mtr}(k,p) &= \frac{Pd}{n_t} \rho_t^{mk(i)} {P_\mu}^{m(i)}(p) p\left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{mp(i)}\right) \\ &= \frac{Pd}{n_t} \rho_t^{mk(i)} {P_\mu}^{m(i)}(p) c_{z_{tr}^{(x)}m} \zeta_t^{mk} \left(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^{mp(i)}\right). \end{aligned} \tag{B.2}$$

3. Calculate the visibility probabilities for each target model using the HMM smoother.

$$P\left(\boldsymbol{D}_t|\mathbf{X},\mathbf{Z}\right) = \frac{\alpha_t\left(\boldsymbol{D}_t\right)\beta_t\left(\boldsymbol{D}_t\right)}{\sum_{\boldsymbol{D}_t}\alpha_t\left(\boldsymbol{D}_t\right)\beta_t\left(\boldsymbol{D}_t\right)} \tag{B.3}$$

$$\alpha_t\left(\boldsymbol{D}_t\right) = \sum_{\boldsymbol{D}_{t-1}} \left\{ \prod_{m=1}^{M_X} \Delta_t^m\left(d_t^m|d_{t-1}^m\right) \right\} P\left(\mathbf{Z}_t|\boldsymbol{D}_t,\mathbf{X}_t\right)\alpha_{t-1}\left(\boldsymbol{D}_{t-1}\right) \tag{B.4}$$

$$\beta_t\left(\boldsymbol{D}_t\right) = \sum_{\boldsymbol{D}_{t+1}} \left\{ \prod_{m=1}^{M_X} \Delta_{t+1}^m\left(d_{t+1}^m|d_t^m\right) \right\} P\left(\mathbf{Z}_{t+1}|\boldsymbol{D}_{t+1},\mathbf{X}_{t+1}\right)\beta_{t+1}\left(\boldsymbol{D}_{t+1}\right) \tag{B.5}$$

$$\begin{aligned} P\left(\mathbf{Z}_t|\boldsymbol{D}_t,\mathbf{X}_t\right) &= \prod_{r=1}^{n_t} P\left(\boldsymbol{z}_{tr}|\boldsymbol{D}_t,\mathbf{X}_t\right) \\ &= \prod_{r=1}^{n_t} \left\{ \left(1 - \frac{Pd}{n_t}\boldsymbol{D}_t{\boldsymbol{D}_t}^{\mathsf{T}}\right)\left(\tilde{w}_{1tr} + \tilde{w}_{2tr}\right) + \sum_{m=3}^{M}\sum_{p=1}^{M_\mu}\sum_{k=1}^{2} d_t^m \tilde{w}_{mtr}(k,p) \right\} \end{aligned} \tag{B.6}$$

The weights are then normalised according to

$$w_{mtr} = \sum_{\boldsymbol{D}_t} P\left(\boldsymbol{D}_t|\mathbf{X},\mathbf{Z}\right) \frac{\left(1 - \frac{Pd}{n_t}\boldsymbol{D}_t{\boldsymbol{D}_t}^{\mathsf{T}}\right)\tilde{w}_{mtr}}{P\left(\boldsymbol{z}_{tr}|\boldsymbol{D}_t,\mathbf{X}_t\right)} \quad \text{if } m = 1,2 \tag{B.7}$$

$$w_{mtr}(k,p) = \sum_{\boldsymbol{D}_t} P\left(\boldsymbol{D}_t|\mathbf{X},\mathbf{Z}\right) \frac{d_t^m \tilde{w}_{mtr}(k,p)}{P\left(\boldsymbol{z}_{tr}|\boldsymbol{D}_t,\mathbf{X}_t\right)} \quad m = 3, \ldots M \tag{B.8}$$

4. Update the estimated variables

   (a) Clutter model states

$$\boldsymbol{x}_t^{m(i+1)} = \frac{\sum_{r=1}^{n_t} w_{mtr} \boldsymbol{z}_{tr}^{(x)}(A)}{\sum_{r=1}^{n_t} w_{mtr}} \tag{B.9}$$

   (b) Clutter mixing proportions

$$\sigma_t^{m(i+1)} = \frac{\sum_{r=1}^{n_t} w_{mtr}}{\sum_{s=1}^{2} \sum_{r=1}^{n_t} w_{str}} \tag{B.10}$$

   (c) Classifier veracity

$$\alpha_m^{(i+1)} = \frac{\sum_{t=1}^{T} \sum_{r=1}^{n_t} w_{mtr} \delta\left(z_{tr}^{(k)} - m\right)}{\sum_{t=1}^{T} \sum_{r=1}^{n_t} w_{mtr}} \tag{B.11}$$

   (d) The estimated state sequence for each dynamic component of each target model is computed using a Kalman Smoother with synthetic measurement

$$\tilde{z}_t^{mp} = \tilde{\mathsf{R}}_t^{mp} \left\{ \sum_{r=1}^{n_t} \left(w_{mtr}(1,p) \mathsf{R}^{-1} + w_{mtr}(2,p) \left(S_t^{mp}\right)^{-1} \right) \boldsymbol{z}_{tr} \right\} \tag{B.12}$$

   and synthetic measurement covariance

$$\tilde{\mathsf{R}}_t^{mp} = \sum_{r=1}^{n_t} \left(w_{mtr}(1,p) + w_{mtr}(2,p) \right) \left\{ \rho_t^{mp(i+1)}(1) R^{-1} + \rho_t^{mp(i+1)}(2) \left(S_t^{mp}\right)^{-1} \right\} \tag{B.13}$$

   (e) Target homothetic measurement component mixing proportions

$$\rho_t^{mp(t+1)}(k) = \frac{\sum_{r=1}^{n_t} w_{mtr}(k,p)}{\sum_{r=1}^{n_t} \sum_{s=1}^{2} w_{mtr}(s,p)} \tag{B.14}$$

(f) Target dynamic component mixing proportions

$$P_{\mu}{}^{m}(p) = \frac{\sum_{t=1}^{T}\sum_{r=1}^{n_t}\sum_{k=1}^{2} w_{mtr}(k,p)}{\sum_{t=1}^{T}\sum_{r=1}^{n_t}\sum_{k=1}^{2}\sum_{s=1}^{M_\mu} w_{mtr}(k,s)} \tag{B.15}$$

5. Calculate the value of the auxiliary function at the new parameter estimate

$$\begin{aligned}
&Q\left(\mathbf{X}^{(i+1)}, \mathbf{P}_{\mu}{}^{(i+1)}, \rho^{(i+1)}, \sigma^{(i+1)}, \mathsf{C}^{(i+1)} | \mathbf{X}^{(i)}, \mathbf{P}_{\mu}{}^{(i)}, \rho^{(i)}, \sigma^{(i)}, \mathsf{C}^{(i)}\right) \\
&= \sum_{m=1}^{2}\left\{\log\psi_0^m\left(\boldsymbol{x}_0^m\right) + \sum_{t=1}^{T}\log\psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right)\right\} \\
&\quad + \sum_{m=3}^{M}\sum_{p=1}^{M_\mu}\left\{\log\psi_0^{mp}\left(\boldsymbol{x}_0^{mp}\right) + \sum_{t=1}^{T}\log\psi_t^{mp}\left(\boldsymbol{x}_t^{mp}|\boldsymbol{x}_{t-1}^{mp}\right)\right\} \\
&\quad + \sum_{m=1}^{M_X}\sum_{p=1}^{M_\mu}\left\{\log P_{\mu}{}^{m}(p)\sum_{t=1}^{T}\sum_{r=1}^{n_t}\sum_{k=1}^{2} w_{mtr}(k,p)\right\} \\
&\quad + \sum_{m=1}^{M_X}\sum_{k=1}^{2}\sum_{t=1}^{T}\left\{\log\rho_t^{mk}\sum_{r=1}^{n_t}\sum_{p=1}^{M_\mu} w_{mtr}(k,p)\right\} \\
&\quad + \sum_{m=1}^{2}\sum_{t=1}^{T}\left\{\log\sigma_t^m\sum_{r=1}^{n_t} w_{mtr}\right\} \\
&\quad + \sum_{t=1}^{T}\sum_{r=1}^{n_t}\left\{\sum_{m=1}^{2}\log c_{z_{tr}^{(k)}m} w_{mtr} + \sum_{m=3}^{M}\log c_{z_{tr}^{(k)}m}\sum_{k=1}^{2}\sum_{p=1}^{M_\mu} w_{mtr}(k,p)\right\} \\
&\quad + \sum_{t=1}^{T}\sum_{r=1}^{n_t}\left\{\sum_{m=1}^{2} w_{mtr}\log\zeta_t^m\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^m\right)\right. \\
&\qquad\qquad \left. + \sum_{m=3}^{M}\sum_{k=1}^{2}\sum_{p=1}^{M_\mu} w_{mtr}(k,p)\log\zeta_t^{mk}\left(\boldsymbol{z}_{tr}^{(x)}|\boldsymbol{x}_t^{mp}\right)\right\} \\
&\quad + \sum_{\boldsymbol{D}_0}\sum_{m=1}^{M_X}\log\Delta_0^m\left(d_0^m\right)P\left(\boldsymbol{D}_0|\mathbf{X},\mathbf{Z}\right) \\
&\quad + \sum_{t=1}^{T}\sum_{\boldsymbol{D}_{t-1}}\sum_{\boldsymbol{D}_t}\sum_{m=1}^{M_X}\log\Delta_t^m\left(d_t^m|d_{t-1}^m\right)P\left(\boldsymbol{D}_{t-1},\boldsymbol{D}_t|\mathbf{X},\mathbf{Z}\right) \\
&\quad + \sum_{t=1}^{T}\sum_{\boldsymbol{D}_t}P\left(\boldsymbol{D}_t|\mathbf{X},\mathbf{Z}\right)\sum_{r=1}^{n_t}\left\{\sum_{m=1}^{2}\tilde{w}_{mtr}\log\left(1-\frac{Pd}{n_t}\boldsymbol{D}_t\boldsymbol{D}_t{}^{\mathsf{T}}\right)\right. \\
&\qquad\qquad \left. + \sum_{m=3}^{M}\sum_{k=1}^{2}\sum_{p=1}^{M_\mu} w_{mtr}(k,p)\log\frac{Pd}{n_t}\right\}
\end{aligned} \tag{B.16}$$

6. Repeat steps 2 3 and 4 until the auxiliary function $Q\left(\cdot^{(i+1)}|\cdot^{(i)}\right)$ converges

Appendix C

Review of the Probabilistic Data Association Filter

Chapter 8 presented a comparison of the PMHT track initiation performance and the Unified Probabilistic Data Association Filter (UPDAF) track initiation performance. This appendix gives a summary of the Probabilistic Data Association Filter (PDAF) and its various extensions leading to the UPDAF.

The Probabilistic Data Association Filter [BST75, BSF88] is an approximation to the optimal recursive Bayesian data association strategy. The PDAF is a single target algorithm, so each track is filtered in isolation, and it is assumed that any measurements due to other targets can be lumped into the clutter. The PDAF enforces the single measurement assignment constraint, namely each target track is only allowed to form at most one measurement. At scan t there are $n_t + 1$ possible assignment hypotheses for target model m. These hypotheses are the assignment of each measurement in turn, and the hypothesis that target model m formed no measurement at scan t.

The standard PDAF makes a number of assumptions about the target and clutter behaviour. The targets are assumed to have linear Gaussian statistics, and all target tracks (i.e. filter models assumed to represent targets) must correspond to valid targets. The clutter is assumed to be uniformly distributed over the surveillance region, and the number of measurements due to clutter is assumed to follow a Poisson distribution. This means that:

$$\begin{aligned} M_Y &= 1 \\ \zeta_t^1\left(\boldsymbol{z}_{tr}|x_t^1\right) &= \frac{1}{V} \\ P\left[\sum_{r=1}^{n_t}\delta\left(k_{tr}-1\right)=n\right] &= \frac{(\lambda V)^n}{n!}\exp\left(-\lambda V\right) \end{aligned}$$

where V is the volume of the surveillance region, and λ is the rate parameter of the Poisson distribution, which is assumed to be known.

Using Bayes Rule, the recursive state estimation problem can be written as:

$$\begin{aligned} p\left(\boldsymbol{x}_t^m|\mathbf{Z}_t,\boldsymbol{x}_{t-1}^m\right) &= \sum_{r=0}^{n_t} p\left(\boldsymbol{x}_t^m,\theta_t(r)|\mathbf{Z}_t,\boldsymbol{x}_{t-1}^m\right) \\ &= \sum_{r=0}^{n_t} p\left(\theta_t^m(r)|\mathbf{Z}_t,\boldsymbol{x}_{t-1}^m\right) p\left(\boldsymbol{x}_t^m|\theta_t^m(r),\mathbf{Z}_t,\boldsymbol{x}_{t-1}^m\right) \end{aligned}$$

$$= \beta_t^m(0)\ \psi_t^m\left(\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right) + \sum_{r=1}^{n_t}\beta_t^m(r)\ p\left(\boldsymbol{x}_t^m|\boldsymbol{z}_{tr},\boldsymbol{x}_{t-1}^m\right), \quad \text{(C.1)}$$

where $\theta_t^m(r)$ represents the rth assignment hypothesis at scan t and $\beta_t^m(r)$ is the posterior probability of that hypothesis, and $\sum_{r=0}^{n_t}\beta_t^m(r) = 1$. Under hypothesis $\theta_t^m(r)$, measurement r is caused by target model m and all other measurements are due to clutter - since the algorithm is a single target one, any measurement not caused by model m must be due to clutter, i.e. model 1. Under hypothesis $\theta_t^m(0)$ all measurements are due to clutter. The term $p\left(\boldsymbol{x}_t^m|\boldsymbol{z}_{tr},\boldsymbol{x}_{t-1}^m\right)$ in (C.1) is the probability density function of the target state when it is updated by measurement z_{tr}. If the probability density of the state at the previous scan is Gaussian, then this expression can be gained through a Kalman Filter.

The density in (C.1) is clearly multimodal. If the prior state distribution is unimodal, then (C.1) will have $n_t + 1$ modes. This is the exponential growth in complexity of the Bayesian association approach. The PDAF makes the assumption that the density (C.1) can be approximated by a single Gaussian, thus reducing the distribution back to a unimodal one. The PDAF thus requires the mean and the variance of the posterior state density. First, consider the mean. Using (C.1), write:

$$E\left\{\boldsymbol{x}_t^m|\mathbf{Z}_t,\boldsymbol{x}_{t-1}^m\right\} = \beta_t^m(0)\ E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\} + \sum_{r=1}^{n_t}\beta_t^m(r)\ E\left\{\boldsymbol{x}_t^m|\boldsymbol{z}_{tr},\boldsymbol{x}_{t-1}^m\right\}. \quad \text{(C.2)}$$

Using a Kalman Filter,

$$E\left\{\boldsymbol{x}_t^m|\boldsymbol{z}_{tr},\boldsymbol{x}_{t-1}^m\right\} = E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\} + \mathsf{W}_t^m\nu_{tr}^m, \quad \text{(C.3)}$$

where the innovation, ν_{tr}^m is given by

$$\nu_{tr}^m = \left[\boldsymbol{z}_{tr} - \mathsf{H}_t^m E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\}\right], \quad \text{(C.4)}$$

and W_t^m is the Kalman gain, which is independent of the measurement.

Substituting (C.3) into (C.2) gives

$$\begin{aligned} E\left\{\boldsymbol{x}_t^m|\mathbf{Z}_t,\boldsymbol{x}_{t-1}^m\right\} &= \beta_t^m(0)\ E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\} + \sum_{r=1}^{n_t}\beta_t^m(r)\ E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\} + \mathsf{W}_t^m\nu_{tr}^m \\ &= E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\} + \sum_{r=1}^{n_t}\beta_t^m(r)\ \mathsf{W}_t^m\nu_{tr}^m \\ &= E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\} + \mathsf{W}_t^m\tilde{\nu}_t^m, \end{aligned} \quad \text{(C.5)}$$

where $\tilde{\nu}_t^m$ is a synthetic innovation given by

$$\tilde{\nu}_t^m = \sum_{r=1}^{n_t}\beta_t^m(r)\left[\boldsymbol{z}_{tr} - \mathsf{H}_t^m E\left\{\boldsymbol{x}_t^m|\boldsymbol{x}_{t-1}^m\right\}\right]. \quad \text{(C.6)}$$

Comparing (C.5) with (C.3) it is clear that the mean of the track state posterior probability density can be obtained using a single Kalman filter with a synthetic innovation. This provides for an efficient implementation for the mean update.

Using similar manipulations, the covariance of the posterior state estimate can be shown [BSF88] to be

$$\begin{aligned}\mathrm{cov}\left\{\boldsymbol{x}_t^m|\mathbf{Z}_t,\boldsymbol{x}_{t-1}^m\right\} &= \beta_{t0}^m\mathsf{P}_t^m(t|t-1)+\sum_{r=1}^{n_t}\beta_{tr}^m\mathsf{P}_t^m(t|t)\\ &\quad+\mathsf{W}_t^m\left[\sum_{r=1}^{n_t}\beta_t^m(r)\tilde{\boldsymbol{z}}_{tr}^m\tilde{\boldsymbol{z}}_{tr}^{m\mathsf{T}}-\tilde{\boldsymbol{z}}_t^m\tilde{\boldsymbol{z}}_t^{m\mathsf{T}}\right]\mathsf{W}_t^{m\mathsf{T}},\end{aligned}\tag{C.7}$$

where $\mathsf{P}_t^m(t|t-1)$ is the covariance of the state estimate given the previous state estimate, and $\mathsf{P}_t^m(t|t)$ is the covariance of the state estimate when it is updated by a measurement at scan t and assignment is known (under this condition, the value of the measurement does not effect this covariance). Both of the covariances $\mathsf{P}_t^m(t|t-1)$ and $\mathsf{P}_t^m(t|t)$ are provided by the Kalman Filter. So, (C.7) can be viewed as a correction to the Kalman Filter Covariance to account for the data association uncertainty.

The PDAF is thus implemented using a Kalman Filter, with a synthetic innovation, and a correction to the posterior covariance. This means the PDAF is a very efficient algorithm, carrying little extra cost than the optimal estimator when the assignments are known.

Many of the β_{tr}^m will be close to zero, and (C.11) is expensive to calculate because of the exponential. To reduce the computation load, it is usual practice to introduce a validation gate, which is generally a physical boundary set at some multiple of the innovation covariance, i.e.

$$\left(\boldsymbol{z}-\mathsf{H}_t^m\hat{\boldsymbol{x}}_{t|t-1}^m\right)^{\mathsf{T}}\mathsf{S}_t^{m-1}\left(\boldsymbol{z}-\mathsf{H}_t^m\hat{\boldsymbol{x}}_{t|t-1}^m\right)=r^2.\tag{C.8}$$

Measurements outside this boundary are assumed to have negligible probability of being due to the target. The number of measurements in the validation region then replaces n_t in the above equations.

Let Pd_t^m be the probability that target m is detected at scan t. This probability may be a function of the target state, especially in the case where physical obstacles may obscure the target. Let Pg_t^m be the probability that a target measurement is within the validation gate, given that it was detected. Usually, the gate is chosen to make Pg_t^m constant.

There are two forms of the PDAF, known as the *parametric* and *nonparametric* PDAF. In the parametric PDAF, it is assumed that the rate parameter of the clutter density, λ, is known. In the nonparametric PDAF, the rate parameter is unknown and is approximated using $\hat{\lambda}=m_t/V$. An alternative nonparametric PDAF in [CDA86] uses the approximation $\hat{\lambda}=n_t/A$ where A is the area of the entire surveillance region. This latter form will be more accurate, if the data really is uniformly distributed.

For the parametric PDAF, the association probabilities, $\beta_t^m(r)$, are given by

$$\beta_t^m(r)=\frac{b_t^m(r)}{\sum_{r=0}^{n_t}b_t^m(r)},\tag{C.9}$$

where

$$b_t^m(0) = \lambda\left|2\pi\mathsf{S}_t^m\right|^{1/2}\frac{(1-Pd_t^mPg_t^m)}{Pg_t^m},\tag{C.10}$$

$$b_t^m(r) = \exp\left\{-\frac{1}{2}\tilde{\boldsymbol{z}}_{tr}^{m\mathsf{T}}\mathsf{S}_t^{m-1}\tilde{\boldsymbol{z}}_{tr}^m\right\},\quad r=1,\ldots,n_t.\tag{C.11}$$

It is important to recognise that the association probabilities, $\beta_t^m(r)$, use the innovation covariance matrix, S_t^m, not the measurement process $\zeta_t^m(\cdot)$. Thus, measurements are assigned based on the current state estimate uncertainty. This is both a boon and a bane for the PDAF. If a target manoeuvres, the state moves away from the predicted state and the innovation covariance grows, allowing the PDAF to recapture the track. However, tracks started on clutter may be allowed to assign distant measurements because the innovation covariance is large.

C.1 Nearest Neighbour Gating

In the standard PDAF, a validation gate is used to restrict the number of possible assignment hypotheses, and hence the computational load. It is assumed that measurements outside the validation gate have a negligible probability of being caused by the target corresponding to the track. The validation gate itself is usually defined by the hyper-ellipse

$$(\boldsymbol{z} - \mathsf{H}_t^m \boldsymbol{x}_t^m)^{\mathsf{T}} \mathsf{S}_t^{m-1} (\boldsymbol{z} - \mathsf{H}_t^m \boldsymbol{x}_t^m) = r^2$$

where S_t^m is the innovation covariance for track m at scan t and r^2 is some fixed threshold setting. [Dru01] calls gating first stage data association.

The purpose of gating is primarily to remove the need to make costly density calculations, such as $\zeta_t^m(\boldsymbol{z}_{tr}|\boldsymbol{x}_t^m)$, for very distant measurements that obviously do not belong to the track. This need not be done by using a fixed radius hyper-ellipse. In [CDA86], Nearest Neighbour Gating was introduced. Here, each track validates the I measurements closest to it. The number of validated measurements is fixed, at I, and the physical size of the effective validation region changes with each scan. This method was originally introduced to improve pipelining of the data association algorithm by removing the dynamic variation in the number of assignment hypotheses. However, it has since shown to give other advantages over a fixed gate [CD00a].

The main advantage of using a fixed number of validated measurements, rather than a spatially fixed gate, is that it makes the algorithm adaptive to the clutter conditions. When the clutter is low (i.e. there are very few false detections) then the effective gate is very large, and it is possible to validate measurements distant from the extrapolated target position, such as may occur after a manoeuvre. When the clutter is high (i.e. there are many false detections) then the effective gate is small, and the track is not easily seduced. While it is possible to design the gate based on the prevailing clutter condition, in practice clutter varies with time and perhaps spatially. [CD00a] demonstrated that using a low number of nearest neighbours, say $I = 3$, does not degrade tracking performance.

C.2 Target Visibility

The standard PDAF makes the assumption that every track corresponds to a valid target. In practice this is an unrealistic assumption. Targets move out of the surveillance region of the sensor, and aircraft land. Thus, what begins as a track corresponding to a valid target may become a track corresponding to no target. Similarly, the target tracks must come from somewhere. Even if the tracks are initiated by a radar operator, it is inevitable that there will be some tracks that have been falsely initiated. A tracking filter for a realistic scenario should be able to deal with these conditions.

The target visibility model is a method for enabling the tracker to make automatic initiation and termination decisions. The model was first introduced in [CDA86] and

has become a popular approach for track decisions with the PDAF, for example [MES94, LL01a, CD03]. The name *Integrated* PDAF (IPDAF) has become associated with the nonparametric PDAF with visibility.

Target visibility is a binary attribute of each target, and it determines whether the sensor is capable of detecting the target or not. If a target is visible, then the sensor receives energy from the target which may be detected with probability Pd. If the target is invisible, then it cannot be detected. A target may be become invisible if it passes into an area occluded by physical limitations of the radar; it might be an aircraft that lands or moves into a blind Doppler zone. By estimating the probability that the target corresponding to each track is visible, tracks can be automatically terminated when their respective targets becomes invisible.

In cluttered environments where the sensor receives false detections, the source of the sensor measurements (target or clutter) is unknown, and it is inevitable that some tracks will be initialised on clutter (that is false detections). Such tracks will fail to associate measurements and appear to the sensor to be invisible targets. Even though the clutter measurements used to start the track were not caused by a target, the sensor cannot differentiate them from measurements from a true target that became invisible. Such tracks can be considered to correspond to a fictitious target that has become invisible. None of the estimated target parameters (such as range and azimuth) have a physical meaning under the hypothesis that there is no target, however the estimated probability of target visibility can be used to test whether there really is a target corresponding to that track. Tracks that are estimated to correspond to invisible targets are discarded and those corresponding to visible targets are retained. This provides a method of automatic track initiation. Further, the probability of target visibility can also be used to discard tracks that associate some valid target measurements but mis-model the dynamics, since these will also be estimated to be invisible targets.

The target visibility attribute for target model m is denoted v_t^m and takes either the value zero or unity, indicating an invisible or visible target respectively. There is no visibility for clutter and so there is no v_t^1. The target visibility attribute is modelled as a first order Markov process and is assumed to be independent of the target dynamics. That is, the probability that the target is visible given it was visible (or invisible) at the previous scan is independent of all other prior information. This probability mass, $P\left(v_t^m | v_{t-1}^m\right)$, can be presented as a transition matrix,

$$P\left(v_t^m | v_{t-1}^m\right) = \begin{bmatrix} 1-\Delta_{01} & \Delta_{01} \\ 1-\Delta_{11} & \Delta_{11} \end{bmatrix}, \tag{C.12}$$

where Δ_{01} is the probability that the target switches from invisible at scan $t-1$ to visible at scan t. Δ_{11} is the probability that the target remains visible at scan t when it was visible at scan $t-1$.

Let $Pv_{t_1|t_2}^m$ represent the probability that the visibility variable $v_{t_1}^m = 1$ given the measurements up to scan t_2. That is,

$$Pv_{t_1|t_2}^m \equiv P\left(v_{t_1}^m = 1 | \mathbf{Z}_1, \ldots, \mathbf{Z}_{t_2}\right). \tag{C.13}$$

Using the Markov property of the visibility, write

$$Pv_{t|t-1} = \Delta_{01}\left(1 - Pv_{t-1|t-1}\right) + \Delta_{11} Pv_{t-1|t-1}. \tag{C.14}$$

The introduction of visibility splits the event $\theta_t^m(0)$ into two events: the target is invisible (and therefore cannot produce a measurement), $\theta_t^m(-1)$, and the target is visible, but

either undetected, or the detected measurement is not validated, $\theta_t^m(0)$. The probability of the invisibility event is given by

$$b_t^m(-1) = \lambda \left|2\pi \mathsf{S}_t^m\right|^{1/2} \frac{\left(1 - Pv_{t|t-1}\right)}{Pg_t^m Pv_{t|t-1}}, \tag{C.15}$$

and the other b's are unchanged (the normalisation equation for β now spans $m+2$ events).

The updated visibility probability, $Pv_{t|t}^m$, is obtained using

$$Pv_{t|t}^m = 1 - \beta_t^m(-1) \tag{C.16}$$

The visibility model has been shown to provide a substantial reduction in false tracks [CDA86].

C.3 Augmented PDAF

The PDAF algorithm is derived for measurements that are a linear function of the target state, such as position and bearing estimates. In practical applications, there may be other features of the observed data that can be used to discriminate target measurements from false detections due to clutter. The extension of the PDAF to deal with such information is referred to as the Augmented PDAF [BSL95].

Suppose that the sensor provides features with each measurement, in addition to the usual state observation components. These features may be continuous valued or discrete, or a combination of both. The combined measurement vector can be written as

$$\boldsymbol{z}_{tr} = \left\{ \begin{array}{c} \boldsymbol{z}_{tr}^{(x)} \\ \boldsymbol{z}_{tr}^{(k)} \end{array} \right\}, \tag{C.17}$$

where $\boldsymbol{z}_{tr}^{(x)}$ is a vector of state observations, and $\boldsymbol{z}_{tr}^{(k)}$ is a vector of features.

Assume that the probability density (mass) function of the feature vector, $\boldsymbol{z}_{tr}^{(k)}$, is known for both targets and clutter. Denote these probabilities as $c_t^m\left(z_{tr}^{(k)}\right)$. Then, the likelihood ratio of the feature vector is

$$\mathcal{L}^m\left(z_{tr}^{(k)}\right) = \frac{c_t^m\left(z_{tr}^{(k)}\right)}{c_t^1\left(z_{tr}^{(k)}\right)}. \tag{C.18}$$

The augmented measurements are then incorporated by multiplying (C.11) by the likelihood ratio, $\mathcal{L}^m\left(z_{tr}^{(k)}\right)$

$$b_t^m(r) = \mathcal{L}^m\left(z_{tr}^{(k)}\right) \exp\left\{-\frac{1}{2}\tilde{\boldsymbol{z}}_{tr}^{m\mathsf{T}} \mathsf{S}_t^{m-1} \tilde{\boldsymbol{z}}_{tr}^m\right\}, \qquad r = 1, \ldots, n_t. \tag{C.19}$$

C.4 Sophisticated Clutter Models

The standard PDAF makes the assumption that the clutter is uniformly distributed. This assumption is rarely accurate. For most sensors, the distribution of false detections is a function of at least one of the measurement components. For example, false detections are more likely at low elevation angles for a ground based radar, or at low Doppler frequencies.

Furthermore, the clutter may not be well modelled by a single distribution at all, but may instead be better described by a mixture model.

When the clutter distribution is known, then the Augmented PDAF approach can be used. Rather than consider the likelihood of the feature, the likelihood of the entire measurement vector can be used,

$$\mathcal{L}^m\left(z_{tr}\right) = \frac{c_t^m\left(z_{tr}^{(k)}\right)}{c_t^1\left(z_{tr}^{(k)}\right)} \frac{\left|2\pi \mathsf{S}_t^m\right|^{-1/2} \exp\left\{-\frac{1}{2}\tilde{\boldsymbol{z}}_{tr}^{m\mathsf{T}} \mathsf{S}_t^{m-1} \tilde{\boldsymbol{z}}_{tr}^m\right\}}{p^1\left(\boldsymbol{z}_{tr}\right)}, \tag{C.20}$$

where $p^1\left(\boldsymbol{z}_{tr}\right)$ is the clutter pdf for the measurement. This likelihood ratio can then be used directly in the event probability equations, with

$$b_t^m(-1) = \lambda \frac{\left(1 - Pv_{t|t-1}\right)}{Pg_t^m Pv_{t|t-1}}, \tag{C.21}$$

$$b_t^m(0) = \lambda \frac{\left(1 - Pd_t^m Pg_t^m\right)}{Pg_t^m}, \tag{C.22}$$

$$b_t^m(r) = \mathcal{L}^m\left(z_{tr}\right), \quad r = 1, \ldots, n_t. \tag{C.23}$$

If the clutter distribution is a mixture, then the denominator of the likelihood ratio can be replaced by the sum over the mixture components weighted by their relative mixing proportions.

When the clutter distribution is unknown, it must be modelled in some way. One way to do this is to assume a parameterised distribution for the clutter probability density function and then estimate the parameters of the distribution. This is the approach taken in [RCD97, RC98, CD00b], where a mixture model for the clutter probability density function is used.

An alternative is to build up an empirical model. A histogram of the received measurements is accumulated over time, and this histogram is used to estimate the probability density function of clutter measurements for future scans. The IPDA-map algorithm in [ME97] uses an approach similar to this to build a nonparametric model for spatially nonuniform clutter.

C.5 Sophisticated Target Models

The PDAF assumes that the target obeys a linear almost constant velocity motion model. In practice this may be true for sections of a flight path, but targets usually turn (change heading) or change speed. It desirable that the tracker accommodate such behaviour and not diverge from the true target trajectory. To do this, a more complicated model of target behaviour must be used.

A commonly used assumption is that the target behaviour can be well modelled by a jump-Markov process. This means that the dynamic behaviour of the target follows one of a family of models, and may switch between these models at random times, according to set of transition probabilities. An example of this is a model set containing an almost constant velocity model and turn models for left and right turns.

If the sequence of dynamic models is known, then the state estimate can be estimated. However, the model sequence is unknown and must be estimated with the states. Since the number of possible model sequences grows exponentially with time, it is impractical to implement an optimal algorithm for estimating the states and the model sequence. This

is very similar to the problem encountered with the visibility Markov model for track decisions as described in chapter 2. However, whereas the visibility model is a switching model controlling the target observation process, this is a switching model controlling the state evolution process.

Like the visibility model, the problem of switching target dynamics can be solved using an IMM-PDAF [MABSD98]. As described in chapter 2, the IMM-PDAF is an approximation to a generalised pseudo Bayes order 2 approach.

An alternative to the IMM is presented in [Col99]. This filter uses a first order GPB structure. The first order GPB approach is simpler than the IMM structure, but IMM is generally considered to give superior performance for manoeuvring targets [BSL95].

C.6 The Unified PDAF

The Multiple Model Unified PDAF (MM-UPDAF) [CD03] is a nonparametric PDAF that incorporates nearest neighbour validation, multiple nonuniform clutter models, multiple target dynamic models and target visibility. This algorithm is used in the operational JFAS detection and tracking software. The event probabilities for the MM-UPDAF are denoted by $\beta_t^{mp}(r)$, where m is a track index, p is a model index, and r is the measurement index. Since the algorithm is a single target one, $\beta_t^{mp}(r)$ is assumed to be independent of $\beta_t nl(s)$ if $n \neq m$.

The JFAS system uses a preprocessor that tags each measurement as being caused by clutter component 1 or 2. This tagging procedure is assumed to be perfect. The event probabilities for target models $m > 2$ are given by

$$\beta_t^{mp}(r) = \frac{b_t^{mp}(r)}{\sum_{s=1}^{M_p} \sum_{i=1}^{I} b_t^{ms}(i)} \tag{C.24}$$

where

$$b_t^{mp}(-1) = Pm_t^{mp} \frac{1 - Pv_{t|t-1}^m}{Pv_{t|t-1}^m}, \tag{C.25}$$

$$b_t^{mp}(0) = Pm_t^{mp} \left(1 - Pd_t^{mp} Ps_t^{mp}\right), \tag{C.26}$$

$$b_t^{mp}(r) = Pm_t^{mp} Pd_t^{mp} \frac{\tilde{\zeta}_t^{mp}\left(z_{tr}|x_t^{mp}\right) n_t^c A_t^c}{\zeta_t^c\left(z_{tr}|x_t^c\right) N_t^c I}, c = \boldsymbol{z}_{tr}^{(k)}. \tag{C.27}$$

Models 1 and 2 are used to represent clutter. The term Pm_t^{mp} is the a priori probability for model p, Ps_t^{mp} is the probability that the target orientated measurement is within the I nearest, given model p is the correct model, n_t^c is the number of measurements tagged as type c from those within the I nearest for all models, N_t^c is the total number of measurements tagged as type c and A_t^c is the spatial area of the component c (provided by the preprocessor).

The derivation of these expressions is presented in detail in [Col99].

C.7 The Joint UPDAF

The MM-UPDAF makes a single target approximation, namely that the assignment of measurements can be performed independently for different target tracks. This assumption is invalid when targets are closely spaced with respect to the sensor resolution. This

circumstance may also arise in OTHR when multiple propagation paths are closely spaced. In order to improve the tracking performance of the MM-UPDAF, this single target assumption was relaxed and a Multiple Model Unified Joint PDAF was developed [DC01]. The event probabilities for this filter are found using substantially more complicated expressions than those above, and are not presented here.

Bibliography

[Aka74] H Akaike, *A new look at the statistical model identification*, IEEE transcations on Automatic Control **19** (1974), no. 6, 716–723.

[AM79] B D O Anderson and J B Moore, *Optimal filtering*, Prentice Hall, 1979.

[BBS88] H A P Blom and Y Bar-Shalom, *The interacting multiple model algorithm for systems with markovian switching coefficients*, IEEE transcations on Automatic Control **33** (1988), no. 8, 780–783.

[Ben81] V E Benes, *Exact finite-dimensional filters with certain diffusion non linear drift*, Stochastics **5** (1981), 65–92.

[Ber79] D P Bertesakas, *A distributed algorithm for the assignment problem*, Lab. for information and decision systems working paper, MIT, March 1979.

[Ber88] D P Bertsekas, *The auction algorithm: A distributed relaxation method for the assignment problem*, Annals of Operations Research: special issue on parallel optimisation **14** (1988), 105–123.

[Bla86] S S Blackman, *Multiple target tracking with radar applications*, Artec House, Norwood, Massachusetss, USA, 1986.

[Blo84] H A P Blom, *An efficient filter for abruptly changing systems*, 23rd conference on decision and control (Las Vegas, Nevada, USA), December 1984.

[BP99] S S Blackman and R Popoli, *Design and analysis of modern tracking systems*, Artec House, Norwood, Massachusetss, USA, 1999.

[BSCB90] Y Bar-Shalom, K C Chang, and H A P Blom, *Multitarget-multisensor tracking: Applications and advances*, vol. 1, ch. 2: Automatic track formation in clutter with a recursive algorithm, pp. 25–42, Artech House, Norwood, Massachusetss, USA, 1990.

[BSCL90] Y Bar-Shalom, L J Campo, and P B Luh, *From receiver operating characteristic to system operating characteristic: Evaluation of a track formation system*, IEEE transcations on Automatic Control **35** (1990), no. 2, 172–179.

[BSF88] Y Bar-Shalom and T E Fortman, *Tracking and data association*, Academic Press, San Diego, California, USA, 1988.

[BSL91] Y Bar-Shalom and X R Li, *Effectiveness of the likelihood function in logic-based track formation*, IEEE transactions on Aerospace and Electronic Systems **27** (1991), no. 1, 184–187.

[BSL95] ———, *Multitarget-multisensor tracking: principles and techniques*, YBS, Storrs, Conneticut, USA, 1995.

[BST75] Y Bar-Shalom and E Tse, *Tracking in a cluttered environment with probabilistic data association*, Automatica **11** (1975).

[CC99] C K Chui and G Chen, *Kalman filtering with real-time applications*, Springer-Verlag, Berlin, Germany, 1999.

[CC01] S B Colegrove and B Cheung, *A peak detector that picks more than peaks*, Radar, 2001.

[CD00a] S B Colegrove and S J Davey, *On using nearest neighbours with the probabilistic data association filter*, Radar, 2000, pp. 53–58.

[CD00b] ———, *The probabilistic data association filter with multiple nonuniform clutter regions*, Radar, 2000, pp. 65–70.

[CD03] ———, *PDAF with multiple clutter regions and target models*, IEEE transactions on Aerospace and Electronic Systems **39** (2003), no. 1, 110–124.

[CDA86] S B Colegrove, A W Davis, and J K Ayliffe, *Track initiation and nearest neighbours incorporated into probabilistic data association*, Journal of Electrical Electronics Engineering, Australia **6** (1986), no. 3, 191–198.

[CDD96] S B Colegrove, L M Davis, and S J Davey, *Performance assessment of tracking systems*, International Symposium on Signal Processing and its Applications (Gold Coast, Australia), vol. 1, August 1996, pp. 188–192.

[CEW94a] B D Carlson, E D Evans, and S L Wilson, *Search radar detection and track with the hough transform I: System concept*, IEEE transactions on Aerospace and Electronic Systems **30** (1994), no. 1, 102–108.

[CEW94b] ———, *Search radar detection and track with the hough transform II: Detection statistics*, IEEE transactions on Aerospace and Electronic Systems **30** (1994), no. 1, 109–115.

[CEW94c] ———, *Search radar detection and track with the hough transform III: Detection performance with binary integration*, IEEE transactions on Aerospace and Electronic Systems **30** (1994), no. 1, 116–125.

[Cha00] S Challa, *Target tracking incorporating flight envelope information*, Proceedings of the 3rd International Conference on Information Fusion (Paris, France), 2000.

[CKBS01] H Chen, T Kirubarajan, and Y Bar-Shalom, *Multiple low observable ballistic missile track initiation using an ESA radar*, ESP lab technical report, University of Conneticut, March 2001.

[Col87] S B Colegrove, *Multitarget tracking in a cluttered environment*, International Symposium on Signal Processing and its Applications (Brisbane, Autralia), 1987.

[Col89] ———, *Australian developments on tracking in a cluttered environment*, IEEE International Conference on Control and Applications, April 1989.

[Col99] ———, *Advanced Jindalee tracker: probabilistic data association multiple model initiation filter*, Technical report TR-0659, Defence Science and Technology Organisation, Australia, June 1999.

[Col00] ———, *Project jindalee: from bare bones to operational OTHR*, Radar, 2000, pp. 825–830.

[CP01] S Challa and G W Pulford, *Joint target tracking and classification using radar and ESM sensors*, IEEE transactions on Aerospace and Electronic Systems **37** (2001), no. 3, 1039–1055.

[CVW02] S Challa, B Vo, and X Wang, *Bayesian approaches to track existence - IPDA and random sets*, Proceedings of the 5th International Conference on Information Fusion (Annapolis, Maryland, USA), July 2002.

[dC05] M de Cervantes, *Don Quixote de la Mancha*, 1605.

[DC99] S J Davey and S B Colegrove, *Advanced Jindalee tracker: Enhanced peak detector*, Technical report TR-0765, Defence Science and Technology Organisation, Australia, June 1999.

[DC01] ———, *A unified joint probabilistic data association filter with multiple models*, Technical report TR-1184, Defence Science and Technology Organisation, Australia, June 2001.

[DDFG01] A Doucet, N De Freitas, and N Gordon, *Sequential monte carlo methods in practice*, Springer Verlag, April 2001.

[DH72] R O Duda and P E Hart, *Use of the hough transformation to detect lines and curves in pictures*, Communications of the ACM **15** (1972), no. 1, 11–15.

[DH97] D T Dunham and R G Hutchins, *Tracking multiple targets in cluttered environments with a probabilistic multi-hypothesis tracker*, Tracking and Pointing XI (SPIE Annual International Symposium on Aerosense, Orlando, Florida, USA), vol. SPIE 3086, April 1997, pp. 284–295.

[DLR77] 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 **140** (1977), 1–38.

[Dru99] O E Drummond, *On features and attributes in multisensor, multitarget tracking*, Proceedings of the 2nd International Conference on Information Fusion (Sunnyvale, California, USA), 1999, pp. 1045–1053.

[Dru01] ———, *Feature, attribute, and classification aided target tracking*, Signal and Data Processing of Small Targets (San Diego, California, USA), vol. SPIE 4473, July 2001, pp. 542–558.

[FBR02] A Farina, D Benvenuti, and B Ristic, *A comparative study of the Benes filtering problem*, Signal Processing **82** (2002), 133–147.

[FBSS83] T E Fortman, Y Bar-Shalom, and M Scheffe, *Sonar tracking of multiple targets using joint probabilistic data association*, IEEE Journal of Oceanic Engineering **8** (1983), 173–184.

[FF99] L Frenkel and M Feder, *Recursive Expectation-Maximistaion (EM) algorithms for time-varying parameters with applications to multiple target tracking*, IEEE transactions on Aerospace and Electronic Systems **47** (1999), no. 2, 306–320.

[Fis81] M L Fisher, *The Lagrangian relaxation method for solving integer programming problems*, Management Science **27** (1981), no. 1, 1–18.

[FJ73] G D Forney Jr, *The viterbi algorithm*, Proceedings of the IEEE **61** (1973), no. 3, 268–278.

[GMF02] A Gad, F Majdi, and M Farooq, *A comparison of data association techniques for target tracking in clutter*, Proceedings of the 5th International Conference on Information Fusion (Annapolis, Maryland, USA), July 2002, pp. 1126–1133.

[GR01] N Gordon and B R Ristic, *Tracking airborne targets occasionally hidden in the blind doppler*, Signal and Data Processing of Small Targets (San Diego, Calfiornia, USA), vol. SPIE 4473, July 2001, pp. 587–595.

[HD97] R G Hutchins and D T Dunham, *Evaluation of a probabilistic multihypothesis tracking algorithm in cluttered environments*, Thirtieth Asilomar Conference on Signals, Systems and Computers (Los Alamitos, California, USA), vol. 2, 1997, pp. 1260–1264.

[HLB97] Z Hu, H Leung, and M Blanchette, *Statistical performance analysis of track initiation techniques*, IEEE transcations on Signal Processing **45** (1997), no. 2, 445–456.

[HLP01] C Hue, J-P Le Cadre, and P Pérez, *The (MR)MTPF: particle filters to track multiple targets using multiple receivers*, Information Fusion (Montreal, Canada), vol. 2, August 2001.

[HLP02] ———, *Sequential mente carlo methods for multiple target tracking and data fusion*, IEEE transcations on Signal Processing **50** (2002), no. 2, 309–325.

[IK88] J Illingworth and J Kittler, *A survey of the hough transform*, Computer Vision, Graphics, and Image Processing **44** (1988), 87–116.

[JO97] S P Jacobs and J A O'Sullivan, *High resolution radar models for joint tracking and recognition*, IEEE International Radar Conference (Syracuse, New York, USA), May 1997, pp. 99–104.

[JR86] B Jang and L Rabiner, *An introduction to hidden markov models*, IEEE ASSP Magazine (1986).

[JV87] R Jonker and A Volgenant, *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, Journal of Computing **38** (1987), 325–340.

[Kal60] R E Kalman, *A new approach to linear filtering and prediction problems*, Journal of Basic Engineering **82** (1960), 34–45.

[KB61] R E Kalman and R Bucy, *New results in linear filtering and prediction problems*, Journal of Basic Engineering **83** (1961), 95–108.

[KG97a] M L Kreig and D A Gray, *Comparison of probabilistic least squares and probabilistic multi-hypothesis tracking algorithms for multi-sensor tracking*, IEEE International Conference on Acoustics, Speech and Signal Processing (Munich, Germany), vol. 1, April 1997, pp. 515–518.

[KG97b] ———, *Multi-sensor probabilistic multi-hypothesis tracking using dissimilar sensors*, Proceedings of the Conference on Acquisition, Tracking and Pointing XI, SPIE Symposium on Aerosense (Orlando, Florida, USA), vol. SPIE 2086, April 1997, pp. 129–138.

[Kre98] M L Kreig, *Probabilistic association and fusion for multi-sensor tracking applications*, Ph.D. thesis, Department of Electrical and Electronic Engineering, the University of Adelaide, Adelaide, South Australia, 1998.

[Kur90] T Kurien, *Multitarget-multisensor tracking: Applications and advances*, vol. 1, ch. 3: Issues in the Design of Practical Multitarget Tracking Algorithms, pp. 43–?, Artech House, Norwood, Massachusetss, USA, 1990.

[LBS90] D Lerro and Y Bar-Shalom, *Automatic tracking with target amplitude feature*, American Control Conference (San Diego, California, USA), 1990.

[LBS01] ———, *Interacting multiple model tracking with target amplitude feature*, IEEE transactions on Aerospace and Electronic Systems **29** (2001), no. 2, 494–509.

[Lee87] M L Lees, *An overview of signal processing for an over-the-horizon radar in the HF band*, International Symposium on Signal Processing and its Applications (Brisbane, Australia), 1987.

[LHB96] H Leung, Z Hu, and M Blanchette, *Evaluation of multiple target track initiation techniques in real radar tracking environments*, IEE Proceedings on Radar, Sonar Navigation? **143** (1996), no. 4, 246–254.

[LKH97] A Logothesis, V Krishnamurthy, and J Holst, *On maneuvring target tracking via the PMHT*, Thirty-Six Conference on Decision and Control (San Diego, California, USA), December 1997.

[LL01a] N Li and X R Li, *Target perceivability and its application*, IEEE transcations on Signal Processing **49** (2001), no. 11, 2588–2604.

[LL01b] ———, *Tracker design based on target perceivability*, IEEE transactions on Aerospace and Electronic Systems **37** (2001), no. 1, 214–225.

[LSW01] T E Luginbuhl, Y Sun, and P Willett, *A track management system for the PMHT algorithm*, Proceedings of the 4th International Conference on Information Fusion (Montreal, Canada), August 2001.

[Lug01] T E Luginbuhl, Personal Correspondence, 2001.

[MABSD98] E Mazor, A Averbuch, Y Bar-Shalom, and J Dayan, *Interacting multiple model methods in target tracking: a survey*, IEEE transactions on Aerospace and Electronic Systems **34** (1998), no. 1, 102–123.

[ME97] D Mušicki and R Evans, *Data association using clutter map information*, Workshop on Signal Processing Applications (Brisbane, Australia), December 1997, pp. 227–230.

[MES94] D Mušicki, R Evans, and S Stankovic, *Integrated probabilistic data association*, IEEE transcations on Automatic Control **39** (1994), no. 6, 1237–1241.

[MK97] G J McLachlan and T Krishnan, *The em algorithm and extensions*, Wiley Interscience, 1997.

[Mor77] C Morefield, *Application of 0-1 integer programming to multi-target tracking problems*, IEEE transcations on Automatic Control **22** (1977), no. 6, 302–312.

[NSC84] V Nagarajan, R N Sharma, and M R Chidambara, *An algorithm for tracking a manouevering target in clutter*, IEEE transactions on Aerospace and Electronic Systems **20** (1984), 560–573.

[PL97] G W Pulford and A Logothesis, *An expectation-maximisation tracker for multiple observations of a single target in clutter*, Thirty-sixth IEEE Conference on Decision and Control (San Diego, California, USA), December 1997, pp. 4997–5003.

[PPK00] K R Pattipati, R L Popp, and T Kirubarajan, *Multitarget-multisensor tracking: Applications and advances*, vol. 3, ch. 2: Survey of Assignment Techniques for Multitarget Tracking, pp. 77–159, Artech House, Norwood, Massachusetss, USA, October 2000.

[RA00] B Ristic and S Arulampalam, *Multitarget mixture reduction algorithm with incorporated target existence recursions*, Signal and Data Processing of Small Targets, vol. SPIE 4048, 2000, pp. 357–368.

[RC98] B Ristic and Colegrove, *Modelling the distribution of false detections for target tracking with an OTHR*, International Radar Symposium (Munich, Germany), vol. 3, sep 1998, pp. 1147–1154.

[RCD97] B Ristic, S B Colegrove, and S J Davey, *Segmentation of the surveillance area of a skywave OTHR for tracking in amplitude*, Workshop on Signal Processing Applications (Brisbane, Australia), 1997, pp. 243–246.

[Rei77] D B Reid, *A multiple hypothesis filter for tracking multiple targets in a cluttered environment*, Report LMSC,D-560254, Lockhead Missiles and Space Company, September 1977.

[Rei79] _____, *An algorithm for tracking multiple targets*, IEEE transcations on Automatic Control **24** (1979), 843–854.

[Ris78] J Rissanen, *Modeling by shortest data description*, Automatica **14** (1978), 465–471.

[RW01a] Y Ruan and P Willett, *An improved PMHT using an idea from coding*, 2001 Aerospace Conference (Big Sky, Montana, USA), March 2001.

[RW01b] _____, *Manoeuvring PMHTs*, Signal and Data Processing of Small Targets (SPIE Annual International Symposium on Optical Science and Technology, San Diego, California, USA), vol. SPIE 4473, July 2001, pp. 416–427.

[RWS95a] C Rago, P Willett, and R L Streit, *A comparison of the JPDAF and PMHT tracking algorithms*, International Conference on Acoustics, Speech and Signal Processing (Detroit, Michigan, USA), 1995, pp. 3571–3574.

[RWS95b] _____, *Direct data fusion using the PMHT*, American Control Conference (Seattle, Washington, USA), 1995, pp. 1678–1682.

[RWS99] Y Ruan, P Willett, and R L Streit, *A comparison of the PMHT and PDAF tracking algorithms based on their model CRLBs*, SPIE Conference on Acquisition, Tracking and Pointing XIII (Orlando, Florida, USA), vol. SPIE 3692, 1999, pp. 177–188.

[Sal90] D J Salmond, *Mixture reduction algorithms for target tracking in clutter*, Signal and Data Processing of Small Targets, vol. SPIE 1305, 1990, pp. 434–445.

[Sch78] G Schwartz, *Estimating the dimension of a model*, Annals of Statistics **6** (1978), no. 2, 461–464.

[SGW01] R L Streit, M Graham, and M J Walsh, *Multiple target tracking of distributed targets using Histogram-PMHT*, Proceedings of the 4th International Conference on Information Fusion (Montreal, Canada), August 2001.

[Sil86] B W Silverman, *Density estimation for statistics and data analysis*, Chapman and Hall/CRC, 1986.

[Sit64] R W Sittler, *An optimal data association problem in surveillance theory*, IEEE transactions on Military Electronics **MIL-8** (1964), 125–139.

[SL95] R L Streit and T E Luginbuhl, *Probabilistic multi-hypothesis tracking*, Technical report 10428, NUWC, Newport, Rhode Island, USA, February 1995.

[Str95] R L Streit, *Track initialization sensitivity in clutter*, Technical report 10426, NUWC, Newport, Rhode Island, USA, February 1995.

[Str00a] _____, Personal Correspondence, 2000.

[Str00b] _____, *Tracking on intensity-modulated data streams*, Technical report 11221, NUWC, Newport, Rhode Island, USA, May 2000.

[Str01] _____, *Tracking targets with specified spectra using the H-PMHT algorithm*, Technical report 11291, NUWC, Newport, Rhode Island, USA, June 2001.

[TK99] S Theodoridis and Koutroumbas K, *Pattern recognition*, Academic Press, 1999.

[Vit67] A J Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE transcations on Information Theory **13** (1967), no. 2, 260–269.

[VL00] N Vlassis and A Likas, *An EM-VDM algorithm for gaussian mixtures with unknown number of components*, Technical report, Computer Science Institute, University of Amsterdam, Intelligent Autonomous Systems, The Netherlands, April 2000.

[VLK00] N Vlassis, A Likas, and B Krose, *Multivariate gaussian mixture modeling with unknown number of components*, Technical report, Computer Science Institute, University of Amsterdam, Intelligent Autonomous Systems, The Netherlands, April 2000.

[WRS98a] P Willett, Y Ruan, and R L Streit, *The PMHT for manoeuvring targets*, Signal and Data Processing of Small Targets (SPIE Annual International Symposium on Aerosense, Orlando Florida, USA), vol. SPIE 3373, 1998, pp. 416–427.

[WRS98b] ———, *A variety of PMHTs*, Workshop commun GdR ISIS (GT 1) and NUWC, approaches probabilistes pour l'extraction multipistes, November 1998.

[YBSPD95] M Yeddanapudi, Y Bar-Shalom, K R Pattipati, and S Deb, *Ballistic missile track initiation with satellite observations*, IEEE transactions on Aerospace and Electronic Systems **31** (1995), no. 3, 1054–1071.

[ZB95] B Zhou and N K Bose, *An efficient algorithm for data association in multitarget tracking*, IEEE transactions on Aerospace and Electronic Systems **31** (1995), no. 1, 458–468.