

Wright State University CORE Scholar

Browse all Theses and Dissertations

Theses and Dissertations

2008

# Study of Multi-Modal and Non-Gaussian Probability Density Functions in Target Tracking with Applications to Dim Target Tracking

Peter V. Hlinomaz Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd\_all

Part of the Engineering Commons

#### **Repository Citation**

Hlinomaz, Peter V., "Study of Multi-Modal and Non-Gaussian Probability Density Functions in Target Tracking with Applications to Dim Target Tracking" (2008). *Browse all Theses and Dissertations*. 257. https://corescholar.libraries.wright.edu/etd\_all/257

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

# Study of Multi-Modal and Non-Gaussian Probability Density Functions in Target Tracking with Applications to Dim Target Tracking

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

By

# PETER V. HLINOMAZ

B.S.E.E., Northwestern University, 1985

M.S.E.E., University of Michigan-Dearborn, 1986

M.B.A., University of Texas at San Antonio, 1996

2008

Wright State University

#### WRIGHT STATE UNIVERSITY

#### SCHOOL OF GRADUATE STUDIES

October 21, 2008

I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY <u>Peter V. Hlinomaz</u> ENTITLED <u>Study of Multi-Modal and Non-Gaussian Probability Density</u> <u>Functions in Target Tracking with Applications to Dim Target Tracking</u> BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

Lang Hong, Ph.D.

**Dissertation Director** 

Ramana Grandhi, Ph.D.

Director, Engineering Ph.D. Program

Joseph F. Thomas, Jr., Ph.D.

Dean, School of Graduate Studies

Committee on Final Examination

Lang Hong, Ph.D.

Fred Garber, Ph.D.

Arnab Shaw, Ph.D.

Arthur Goshtasby, Ph.D.

Raj K. Bhatnagar, Ph.D.

#### ABSTRACT

Hlinomaz, Peter Vladimir, Ph.D., Engineering Ph.D. Program, Wright State University, 2008, Study of Multi-Modal and Non-Gaussian Probability Density Functions in Target Tracking with Applications to Dim Target Tracking

The majority of deployed target tracking systems use some variant of the Kalman filter for their state estimation algorithm. In order for a Kalman filter to be optimal, the measurement and state equations must be linear and the process and measurement noises must be Gaussian random variables (or vectors). One problem arises when the state or measurement function becomes a multi-modal Gaussian mixture. This typically occurs with the interactive multiple model (IMM) technique and its derivatives and also with probabilistic and joint probabilistic data association (PDA/JPDA) algorithms. Another common problem in target tracking is that the target's signal-to-noise ratio (SNR) at the sensor is often low. This situation is often referred to as the dim target tracking or track-before-detect (TBD) scenario. When this occurs, the probability density function (PDF) of the measurement likelihood function becomes non-Gaussian and often has a Rayleigh or Ricean distribution. In this case, a Kalman filter variant may also perform poorly. The common solution to both of these problems is the particle filter (PF). A key drawback of PF algorithms, however, is that they are computationally expensive. This dissertation, thus, concentrates on developing PF algorithms that provide comparable performance to conventional PFs but at lower particle costs and presents the following four research efforts.

1. A multirate multiple model particle filter (MRMMPF) is presented in Section-3. The MRMMPF tracks a single, high signal-to-noise-ratio, maneuvering target in clutter. It coherently accumulates measurement information over multiple scans via discrete wavelet transforms (DWT) and multirate processing. This provides the MRMMPF with a much stronger data association capability than is possible with a single scan algorithm. In addition, its particle filter nature allows it to better handle multiple modes that arise from multiple target motion models. Consequently, the MRMMPF provides substantially better root-mean-square error

iii

(RMSE) tracking performance than either a full-rate or multirate Kalman filter tracker or fullrate multiple model particle filter (MMPF) with a same particle count.

- 2. A full-rate multiple model particle filter for track-before-detect (MMPF-TBD) and a multirate multiple model particle filter for track-before-detect (MRMMPF-TBD) are presented in Section-4. These algorithms extend the areas mentioned above and track low SNR targets which perform small maneuvers. The MRMMPF-TBD and MMPF-TBD both use a combined probabilistic data association (PDA) and maximum likelihood (ML) approach. The MRMMPF-TBD provides equivalent RMSE performance at substantially lower particle counts than a full-rate MMPF-TBD. In addition, the MRMMPF-TBD tracks very dim constant velocity targets that the MMPF-TBD cannot.
- 3. An extended spatial domain multiresolutional particle filter (E-SD-MRES-PF) is developed in Section-5. The E-SD-MRES-PF modifies and extends a recently developed spatial domain multiresolutional particle filter prototype. The prototype SD-MRES-PF was only demonstrated for one update cycle. In contrast, E-SD-MRES-PF functions over multiple update cycles and provides comparable RMSE performance at a reduced particle cost under a variety of PDF scenarios.
- 4. Two variants of a single-target Gaussian mixture model particle filter (GMMPF) are presented in Section-6. The GMMPF models the particle cloud as a Gaussian finite mixture model (FMM). MATLAB simulations show that the GMMPF provides performance comparable to a particle filter but at a lower particle cost.

#### ACKNOWLEDGEMENTS

There are many people who have helped me complete the long road toward my Ph.D. It would be impossible to mention them all in the brief space allotted. First and foremost, I wish to thank my parents who sacrificed so much time, money, and energy in my upbringing and who strived to instill in me a spirit of lifelong learning.

I would especially like to thank my adviser, Dr. Lang Hong, for his mentoring and wise counsel. Without his patient help it would have been impossible for me to reach my academic goals. I wish to also thank my other committee members, including Dr. Fred Garber, Dr. Arnab Shaw, Dr. Ardeshir Goshtasby, and Dr. Raj K. Bhatnagar, for their time and their sage advice.

I would also like to thank my colleagues and my managers at the MITRE Corporation for their support. Without their encouragement and the continued support of the MITRE Corporation I would not have made it to the finish line.

Finally, I would like to thank my long suffering wife, Joan. She has put up with nine years of my doctoral studies. During this time, she has artfully juggled the role of wife, mother, and registered nurse and has carried much of the load that was rightfully mine. From now on Joan, I promise to do my chores.

# TABLE OF CONTENTS

1	INT	RODUCTION1
	1.1	Problem Definition
	1.2	Summary of Contributions4
2	PRE	EVIOUS WORK6
	2.1	Kinematic State Estimation
	2.2	Kalman Filter: Linear/Gaussian Special Case7
	2.3	Causes of Multimodality9
	2.3.1	Multiple Modeling (MM) Approach9
	2.3.2	Interacting Multi-Pattern Probabilistic Data Association (IMPDA)13
	2.3.3	Joint Probabilistic Data Association (JPDA)18
	2.4	Estimation Techniques to Address Multimodality21
	2.4.1	Approximate Grid-Based Methods22
	2.4.2	2 Gaussian Sum Filter (GSF)23
	2.4.3	Particle Filter (PF)25
	2.5	Multiple Model Particle Filter (MMPF)
	2.6	Initial Multirate Particle Filter Efforts
	2.7	Maintaining Multi-Modality in Particle Filters
	2.8	Measurement Gating With Multi-Modal Likelihood Functions
	2.9	Particle Filter Track Before Detect (TBD-PF)
	2.10	Spatial-Domain Multi-Resolution Particle Filtering (SD-MRES-PF)
3	MUI	LTIRATE - MULTIPLE MODEL PARTICLE FILTER (MRMMPF)37
	3.1	MRMMPF Theoretical Description and Design
	3.1.1	1/3-Rate MMPF Initialization

	3.1.3	1/3-Rate Mixing	42
	3.1.4	1/3-Rate MMPF	43
	3.1.5	Full-Rate MMPF	51
Л	мш	TIRATE MULTIPLE MODEL PARTICLE FULTER TRACK BEFORE	=
-			-
D	DETECT	(MRMMPF-TBD)	54
	4.1	MRMMPF-TBD Algorithm Overview	54
	4.2	Full-Rate Target Models	57
	4.3	Third-Rate Target Models	58
	4.4	Full-Rate Measurement Model	60
	4.5	Third-Rate Measurement Model	61
	4.6	Particle Weight Computation (Full-Rate)	62
	4.7	Particle Weight Computation (1/3-Rate)	66
	4.8	Measurement Gating	68
5	EXT	ENDED SPATIAL DOMAIN MULTI-RESOLUTION PARTICLE	
_			70
		NG (E-3D-MRE3-FF)	/ Z
6	GAL	JSSIAN FINITE MIXTURE MODEL PARTICLE FILTERS (GMMPF)	76
	6.1	Gaussian Finite Mixture Models (FMM)	76
	6.2	GMMPF and K-GMMPF Algorithms	77
7	SCE	NARIOS AND SIMULATION RESULTS	82
	7 4		00
	7.1		02
	7.1.1		δ2
	7.1.2	Scenario-1 (a = +/- 5 m/s <sup>-</sup> ) Results	84
	7.1.3	Scenario-2 (a = +/- 15 m/s <sup>-2</sup> ) Results	93
	7.1.4	Scenario-3 (a = +/- 25 m/s <sup>-2</sup> ) Results	96

	7.1.5	Scenario-4 (a = +/- 40 m/s <sup>-2</sup> ) Results	99
7	.2	MRMMPF-TBD and MMPF-TBD	102
	7.2.1	Scenario-1 Description: Mildly Maneuvering Target	102
	7.2.2	Scenario-2: Non-Maneuvering Target	104
	7.2.3	Scenario-1 results	104
	7.2.4	Scenario-2 Results	
7	.3	E-SD-MRES-PF vs. Standard Uni-Resolutional Bootstrap Filter (BPF)	110
	7.3.1	E-SD-MRES-PF Scenario Description	111
	7.3.2	Results	111
7	.4	GMMPF AND K-GMMPF	116
	7.4.1	GMMPF AND K-GMMPF Scenario	116
	7.4.2	GMMPF and K-GMMPF Results	117
7	.5	Summary and Discussion of Results	117
	7.5.1	MMPF and MRMMPF Summary	118
	7.5.2	MMPF-TBD and MRMMPF-TBD Summary	118
	7.5.3	E-SD-MRES-PF Summary	118
	7.5.4	GMMPF AND K-GMMPF Summary	119
8	CO	NCLUSIONS	120
9	FUT	URE WORK	121
9	.1	MMPF-TBD and MRMMPF-TBD	121
9	.2	E-SD-MRES-PF	
9	.3	GMMPF and K-GMMPF	122
10	REF	ERENCES	123
AP	PEN	DIX-A: LIST OF ABBREVIATIONS	131

# LIST OF FIGURES

Figure 1.1 Actual Gaussian Mixture PDF of Target State	2
Figure 1.2 Moment-Matched Gaussian Representation of PDF Target State	2
Figure 2.1 IMM Algorithm Block Diagram	11
Figure 2.2 Extraction of Patterns from a Sequence	15
Figure 2.3 Pattern Mapping From One Window to Next	15
Figure 2.4 IMPDA Algorithm Block Diagram	18
Figure 2.5 Moment-Matched Gaussian Representation of PDF in JPDA	22
Figure 3.1 MRMMPF Algorithm Block Diagram	39
Figure 4.1 MRMMPF-TBD Block Diagram	56
Figure 4.2 1-R Coarse and Fine Gating Example	71
Figure 5.1 Multiresolutional Decomposition via DWT Filter Bank	73
Figure 5.2 Thresholding Example via the Explicit Method	75
Figure 6.1 Gaussian Finite Mixture Model Particle Filter (GMMPF)	81
Figure 7.1 True Track vs. Noisy Measurements + FA	85
Figure 7.2 IMMPDAF RMS Position Errors (a=+/-5)	86
Figure 7.3 IMPDA RMS Position Errors (a=+/-5)	86
Figure 7.4 MMPF RMS Position Error (a=+/-5)	87
Figure 7.5 MRMMPF RMS Position Error (a=+/-5)	87
Figure 7.6 IMMPDAF RMS Velocity Errors (a=+/-5)	88
Figure 7.7 IMPDA RMS Velocity Errors (a=+/-5)	88
Figure 7.8 MMPF RMS Velocity Errors (a=+/-5)	89
Figure 7.9 MRMMPF RMS Velocity Errors (a=+/-5)	89
Figure 7.10 IMPDA Pattern Probabilities (a=+/-5)	90
Figure 7.11 MRMMPF Pattern Probabilities (a=+/-5)	90
Figure 7.12 IMMPDAF Model Probabilities (a=+/-5)	91
Figure 7.13 IMPDA Model Probabilities (a=+/-5)	91

Figure 7.14 MMPF Model Probabilities (a=+/-5)	92
Figure 7.15 MRMMPF Model Probabilities (a=+/5)	92
Figure 7.16 IMMPDAF RMS Position Errors (a=+/-15)	94
Figure 7.17 IMPDA RMS Position Errors (a=+/-15)	94
Figure 7.18 MMPF RMS Position Errors(a=+/-15)	95
Figure 7.19 MRMMPF RMS Position Errors (a=+/-15)	95
Figure 7.20 IMMPDAF RMS Position Errors (a=+/-25)	97
Figure 7.21 IMPDA RMS Position Errors (a=+/-25)	97
Figure 7.22 MMPF RMS Position Errors (a=+/-25)	98
Figure 7.23 MRMMPF RMS Position Error (a=+/-25)	98
Figure 7.24 IMMPDAF RMS Position Error (a=+/-40)	100
Figure 7.25 IMPDA RMS Position Errors (a=+/-40)	100
Figure 7.26 MMPF RMS Position Errors (a=+/-40)	101
Figure 7.27 MRMMPF RMS Position Errors (a=+/-40)	101
Figure 7.28 X-Position RMS Error Summary for All Scenarios	102
Figure 7.29 MMPF-TBD vs. MRMMPF-TBD Position RMSE (2000 Particles, SNR = 10dB)	107
Figure 7.30 MMPF-TBD vs. MRMMPF-TBD Velocity RMSE (2000 Particles, SNR = 10dB)	107
Figure 7.31 MMPF-TBD Model Probabilities (2000 Particles, SNR = 10dB)	108
Figure 7.32 MRMMPF-TBD Model Probabilities (2000 Particles, SNR = 10dB)	108
Figure 7.33 MRMMPF-TBD Mean Particle Count vs. Time (2000 Particles, SNR = 10dB)	109
Figure 7.34 MMPF-TBD Position RMSE vs. Particle Count Sensitivity	109
Figure 7.35 MRMMPF-TBD Position RMSE vs. Particle Count Sensitivity	110
Figure 7.36 Uni-Res vs. Multi-Res Performance (Scenario-1)	112
Figure 7.37 Uni-Res vs. Multi-Res Performance (Scenario-2)	114
Figure 7.38 Uni-Res vs. Multi-Res Performance (Scenario-3)	115
Figure 7.39 Particle Efficiency Ratio Summary	115

# LIST OF TABLES

Table 6-1 Filter Run Times (Rounded to Nearest Minute)	76
Table 7-1 Tracking Scenarios	83
Table 7-2 RMS Position and Velocity Errors (a=+/-5)	85
Table 7-3 RMS Position and Velocity Errors (a=+/-15)	93
Table 7-4 RMS Position and Velocity Errors (a=+/-25)	96
Table 7-5 RMS Position and Velocity Errors (a=+/-40)	99
Table 7-6 X-Position RMS Error Summary for All Scenarios	102
Table 7-7 MRMMPF-TBD vs. MMPF-TBD Performance Summary: Scenario-1	106
Table 7-8 MRMMPF-TBD vs. MMPF-TBD Performance Summary: Scenario-2	106
Table 7-9 Uni-Res vs. Multi-Res RMSE (Scenario-1)	112
Table 7-10 Uni-Res vs. Multi-Res RMSE (Scenario-2)	113
Table 7-11 Uni-Res vs. Multi-Res RMSE (Scenario-3)	114
Table 7-12 GMMPF/K-GMMPF Results	

## **1** INTRODUCTION

This dissertation studies the impact of non-Gaussian and multi-modal probability density functions in target tracking. The majority of currently deployed target tracking systems use some variant of the Kalman filter for their state estimation algorithm [2,3,4,5,7,8,9,10]. In order for a Kalman filter to be optimal, the measurement and state equations must be linear and the process and measurement noises must be Gaussian random variables (or vectors). In reality, the linearity assumptions often do not hold. When this occurs, standard Kalman filter variants such as the extended Kalman filter (EKF) and unscented Kalman filter (UKF) generally perform well. One problem area arises when the state or measurement function becomes a multi-modal Gaussian mixture. This situation commonly occurs in the following tracking scenarios:

- Interacting Multiple Models (IMM);
- Interacting multi-pattern data association (IMPDA);
- Joint probabilistic data association (JPDA).

In all of these cases, a standard Kalman-filter variant attempts to represent a Gaussian mixture as a single, moment-matched, Gaussian probability density function (PDF). An example of this phenomenon is illustrated in Figure 1.1 and Figure 1.2 below. The PDF in Figure 1.1 is a Parzen estimate [10] of the X-position target state component in a multiple model particle filter (MMPF) while Figure 1.2 is a moment-matched approximation of that PDF. It is evident from the two figures that the single Gaussian poorly represents the actual mixture PDF. For a target tracking algorithm, the end result of this oversimplification is less accurate tracking.

Another common problem in target tracking is that the target's signal-to-noise ratio (SNR) at the sensor is often low. This situation is often referred to as the dim target tracking or track-

before-detect (TBD) scenario. When this occurs, the PDF of the measurement likelihood function becomes non-Gaussian and often has a Rayleigh or Ricean distribution. In this case, Kalman filter derivatives often perform poorly.



Figure 1.1 Actual Gaussian Mixture PDF of Target State



Figure 1.2 Moment-Matched Gaussian Representation of PDF Target State

#### 1.1 **Problem Definition**

The standard technique that has been used in recent years to attack both the multi-modal and dim-target problems is particle filtering. Although standard particle filters perform better in multi-modal/non-Gaussian scenarios than other algorithms, they suffer from several key drawbacks. They do not coherently accumulate information over multiple scans (i.e. all data association hypotheses resolved at each measurement update). Particle filters are also computationally costly with run times that are 2-3 orders of magnitude longer than Kalman filterbased estimators.

In addition, current particle filter TBD algorithms assume constant velocity (CV) motion and full-rate filter updates (i.e. at every measurement scan). Previous work in multirate processing has shown that multirate tracking algorithms can provide comparable performance at a lower computational cost. To date these multirate approaches have not yet been applied to low SNR targets.

Thus, the main goal of this research is to combine:

- Multiple model particle filtering (MMPF);
- Track-before-detect (TBD) techniques;
- Multirate processing in order to track low-SNR targets at a reduced particle cost.

Secondary goals are to:

- Extend current multiresolutional particle filtering techniques in order to provide equivalent RMSE performance at reduced particle counts;
- Investigate the feasibility of combining finite mixture models (FMM) and particle filtering in order to reduce computational costs.

#### 1.2 Summary of Contributions

This dissertation presents four original research efforts that focus on each of the preceding particle filter issues.

- 1. A multirate multiple model particle filter (MRMMPF) is presented in Section-3. The MRMMPF tracks a single, high signal-to-noise-ratio, maneuvering target in clutter. It coherently accumulates measurement information over multiple scans via discrete wavelet transforms (DWT) and multirate processing. This provides the MRMMPF with a much stronger data association capability than is possible with a single scan algorithm. In addition, its particle filter nature allows it to better handle multiple modes that arise from multiple target motion models. As a consequence, the MRMMPF provides much better root-mean-square error (RMSE) tracking performance than either a full-rate or multirate Kalman filter tracker or full-rate MMPF with a same particle count. Note: Due to the large runtimes encountered with the MMPF and the MRMMPF, subsequent efforts were re-focused on reducing runtimes while maintaining RMSE performance rather simply reducing RMSE.
- 2. A full-rate multiple model particle filter for track before detect (MMPF-TBD) and a multirate multiple model particle filter for track-before-detect (MRMMPF-TBD) are presented in Section-4. These algorithms extend the MMPF and MRMMPF so that they can track low SNR targets which perform small maneuvers. The MRMMPF-TBD and MMPF-TBD both use a combined probabilistic data association (PDA) and maximum likelihood (ML) approach. The MRMMPF-TBD provides equivalent RMSE performance at substantially lower particle counts than a full-rate MMPF-TBD. In addition, the MRMMPF-TBD also tracked very dim constant velocity targets that the MMPF-TBD could not.
- An extended spatial domain multiresolutional particle filter (E-SD-MRES-PF) is developed in Section-5. The E-SD-MRES-PF modifies and extends a recently developed spatial domain multiresolutional particle filter prototype [71]. The prototype SD-MRES-PF was only

demonstrated for one update cycle. In contrast, the E-SD-MRES-PF functions over multiple update cycles and provides comparable RMSE performance at a reduced particle cost.

4. Two variants of a single-target Gaussian mixture model particle filter (GMMPF) are presented in Section-6. The GMMPF models the particle cloud as a Gaussian finite mixture model. MATLAB simulations show that the GMMPF provides performance comparable to a standard particle filter but at substantially less particle cost.

### 2 PREVIOUS WORK

#### 2.1 Kinematic State Estimation

In order to understand the role of particle filter-based estimation it is first useful to briefly overview basic estimation concepts and summarize relevant work done to date. All kinematic state estimation algorithms seek to estimate the kinematic state (i.e. position, velocity, and possibly acceleration) of a target from a sequence of measurements that have been corrupted by noise. The target kinematic state at time instant k,  $x_k$ , can be described by the following difference equation:

$$x_k = f_{k-1}(x_{k-1}) + w_{k-1}$$
(2.1)

where:  $f_k(x_k)$  is a target kinematic model and  $w_k$  is an additive process noise term.

The target measurement at time instant k,  $z_k$ , can likewise be defined by an analogous difference equation:

$$z_k = h_k(x_k) + v_k \tag{2.2}$$

where:  $h_k(x_k)$  defines the measurement model and  $V_k$  is an additive measurement noise term.

The probability density function (PDF) of the target state conditioned on the measurement set,  $p(x_k|z_{1:k})$ , can be described via the following recursive Bayesian relationship below:

$$p(x_{k} | z_{1:k}) = p(x_{k} | z_{k}, z_{1:k-1}) = \frac{p(z_{k} | x_{k}, z_{1:k-1}) p(x_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})}$$

$$= \frac{p(z_{k} | x_{k}, z_{1:k-1}) p(x_{k} | z_{1:k-1})}{\int p(z_{k} | x_{k}, z_{1:k-1}) p(x_{k} | z_{1:k-1}) dx_{k}} = \frac{p(z_{k} | x_{k}) p(x_{k} | z_{1:k-1})}{\int p(z_{k} | x_{k}) p(x_{k} | z_{1:k-1}) dx_{k}}$$
(2.3)

The prior probability,  $p(x_k | z_{1:k-1})$ , is defined by the Chapman-Kolmogorov equation as described by Bar Shalom and Li. [6] and can also be derived via the total probability theorem:

$$p(x_{k} | z_{1:k-1}) = \int p(x_{k} | x_{k-1}, z_{1:k-1}) p(x_{k-1} | z_{1:k-1}) dx_{k-1} =$$

$$\int p(x_{k} | x_{k-1}) p(x_{k-1} | z_{1:k-1}) dx_{k-1}$$
(2.4)

The fact that the process evolution is first order Markov allows the conditioning on  $z_{1:k-1}$  to be removed from both the transition prior probability,  $p(x_k | x_{k-1})$ , and the likelihood function,  $p(z_k | x_{k-1})$ .

#### 2.2 Kalman Filter: Linear/Gaussian Special Case

Although (2.3) defines the solution to the optimum estimation problem, it is generally impossible to solve analytically. An exact analytical solution is possible only when (2.1-2.2) describe linear systems and when both the process noise (whose covariance matrix denoted as  $Q_k$ ) and the measurement noise (covariance matrix denoted as  $R_k$ ) are Gaussian. When this occurs, the target state PDF,  $p(x_{k+1} | z_{k+1})$ , can be computed via the standard linear Kalman filter (LKF) equations shown below in (2.5-2.12). Note: since the process is assumed to be Markov, the dependency on measurements prior to time *k* is dropped in the LKF equations.

$$x_{k+1|k} = F_k x_k$$
 (State Mean Prediction) (2.5)

$$P_{k+1|k} = F_k P_{k|k} F'_k + Q_k$$
 (State Covariance Prediction) (2.6)

$$S_{k+1} = H_{k+1} P_{k+1|k} H'_{k+1} + R_{k+1} \qquad (\text{Innovation Covariance}) \qquad (2.7)$$

$$K_{k+1} = P_{k+1|k} H'_{k+1} S_{k+1}^{-1}$$
 (Filter Gain) (2.8)

$$\widetilde{z}_{k+1|k} = H_{k+1} x_{k+1|k}$$
 (Measurement Prediction) (2.9)

 $\boldsymbol{\nu}_{k+1} = \boldsymbol{z}_{k+1} - \widetilde{\boldsymbol{z}}_{k+1|k} \tag{Innovation} \tag{2.10}$ 

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1} V_{k+1}$$
 (State Mean Update) (2.11)

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} S_{k+1} K'_{k+1}$$
 (State Covariance Update) (2.12)

where:

F<sub>k</sub> is the linear state transition matrix

 $H_k$  is the linear measurement matrix

**Note:** Deterministic control inputs,  $\Gamma_k u_k$ , are assumed to be zero without loss of generality.

In realistic target tracking scenarios, the Gaussian and linear assumptions often are not valid because either the system dynamics are nonlinear (due to target maneuvers) or the measurement prediction equation is a non-linear function of the state (i.e. state equations are in Cartesian coordinates while measurements are in polar coordinates). A variety of filters have been developed to deal with these non-linear situations. These include:

- Converted measurements Kalman filter (CMKF); [4]
- Extended Kalman filter (EKF) [7];
- Unscented Kalman filter (UKF) [23,24,25];

- Biscay distribution filter (BDF) [22];
- Gauss-Hermite filter (GHF) [19,20].

Analysis by Cui, Hong, and Layne [46] and Farina et al. [40] indicates that when the only issue is a small-to-moderate nonlinearity, all of these filters provide very similar performance. The real difficulties arise when either the process and/or measurement noises are non-Gaussian or when the state PDF is a multi-modal.

#### 2.3 Causes of Multimodality

The three situations that give rise to multimodality are:

- 1. Multiple modeling (MM) approach [6];
- 2. Interacting multi-pattern data association (IMPDA) [45];
- 3. Joint probabilistic data association (JPDA) [2].

#### 2.3.1 Multiple Modeling (MM) Approach

In the multiple modeling approach with switching models, the state and measurement equations are described via (2.13-2.14).

$$x_{k} = f_{k-1}(x_{k-1}, M_{k}) + w_{k-1}(M_{k})$$
(2.13)

$$z_{k} = h_{k}(x_{k}, M_{k}) + v(M_{k})$$
(2.14)

The variable *M* is the model index parameter that can take on values of M = 1,...r. By applying the total probability theorem, the PDF of the target state at time index *k* can be expressed via (2.15).

$$p(x_{k}|z_{1:k}) = \sum_{j=1}^{r} p(x_{k}|M_{k}^{j}, z_{1:k}) \cdot p(M_{k}^{i}|z_{1:k})$$
(2.15)

If one assumes that the model index depends on a Markov process, the mode transition probability of mode *i* into mode *j* can be defined as:

$$h^{ij} = p\left(M_{k}^{j} \mid M_{k-1}^{i}\right)$$
(2.16)

and mechanized as a pre-defined model transition matrix. As the time index k increases the number of possible model histories increases exponentially with  $r^{k}$ .

Thus, if each one of the model paths is modeled via a Kalman filter, the target state PDF shown below in (2.17) is a Gaussian mixture with an exponentially increasing number of terms.

$$p(x_{k} | z_{1:k}) = \sum_{l=1}^{r^{k}} p(x_{k} | M_{k,l}, z_{1:k}) \cdot p(M_{k,l} | z_{1:k})$$
(2.17)

where: l = model history throughout the trajectory

Although the approach above generates the optimal minimum variance estimate, it is evident that this technique is impractical even for small values of *r*.

#### 2.3.1.1 Interactive Multiple Model (IMM) Algorithm

The most common sub-optimal approach is the Interactive Multiple Model (IMM) [6]. A functional diagram of a two-model IMM is depicted below in Figure 2.1.



Figure 2.1 IMM Algorithm Block Diagram

The mixing probabilities are defined as:

$$u_{k-1|k-1}^{i|j} \equiv p(M_{k-1}^{i} | M_{k}^{j}, z_{1:k-1}) = \frac{p(M_{k}^{j} | M_{k-1}^{i}, z_{1:k-1}) p(M_{k-1}^{i} | z_{1:k-1})}{p(M_{k}^{j} | z_{1:k-1})}$$

$$= \frac{p(M_{k}^{j} | M_{k-1}^{i}, z_{1:k-1}) p(M_{k-1}^{i} | z_{1:k-1})}{\sum_{i=1}^{r} p(M_{k}^{j} | M_{k-1}^{i}, z_{1:k-1}) p(M_{k-1}^{i} | z_{1:k-1})}$$

$$= \frac{p(M_{k}^{j} | M_{k-1}^{i}) p(M_{k-1}^{i} | z_{1:k-1})}{\sum_{i=1}^{r} p(M_{k}^{j} | M_{k-1}^{i}) p(M_{k-1}^{i} | z_{1:k-1})}.$$
(2.18)

The mixing probabilities can now be expressed more compactly via (2.19).

$$u_{k-1|k-1}^{i|j} \equiv \frac{h^{ij}u_{k-1}^{i}}{\sum_{i=1}^{r} h^{ij}u_{k-1}^{i}} \quad i, j = 1, \cdots, r$$
(2.19)

where:  $u_{k-1}^i = \text{prob. of } i^{th} \text{ model at } k-1$ 

In the optimal MM, each mixing operation results in a new set of Gaussian mixtures. The IMM, however, simplifies the PDF by approximating it as a single Gaussian, $N[\hat{x}_{k-1|k-1}^{0j}, P_{k-1|k-1}^{0j}]$ , where the mean is given by:

$$\hat{x}_{k-1|k-1}^{0\,j} = \sum_{i=1}^{r} \hat{x}_{k-1|k-1}^{i} u_{k-1|k-1}^{i|j}, \quad j = 1 \cdots r$$
(2.20)

and the covariance is expressed as:

$$P_{k-1|k-1}^{0j} = \sum_{i=1}^{r} u_{k-1|k-1}^{i|j} \{ P_{k-1|k-1}^{i} + [\hat{x}_{k-1|k-1}^{i} - \hat{x}_{k-1|k-1}^{0j}] \cdot [\hat{x}_{k-1|k-1}^{i} - \hat{x}_{k-1|k-1}^{0j}]' \},$$
  

$$j = 1, \cdots r.$$
(2.21)

The IMM resolves the problem of exponentially increasing model history by maintaining a constant number of model terms. If the individual means of the Gaussian mixture components are close together then the Gaussian approximation (with its single mode) is reasonably accurate. If, however, the means are widely separated then the single Gaussian approximation is a poor representation of the true PDF. After the mixing process, the individual mixed states,  $\hat{x}_{k-1|k-1}^{0,j}$ ,  $P_{k-1|k-1}^{0,j}$ , are processed via a Kalman filter that generates a posterior estimate,  $\hat{x}_{k|k}^{j}$ ,  $P_{k|k}^{j}$ , for each model, and a model likelihood,  $\Lambda_{k}^{j}$ . The model likelihood is computed via (2.22).

$$\Lambda_{k}^{j} = p[z_{k} | M_{k}^{j}, Z^{k-1}] = N\{[z_{k} - h(\hat{x}_{k|k-1}^{0j})]; S_{k}^{j}\}, \ j = 1 \cdots r$$
(2.22)

The next step in the IMM algorithm is to update the mode probabilities,  $u_k^j$ . The mode probabilities are defined as shown in (2.23).

$$u_k^j \equiv p[M_k^j | Z^k] = \frac{\Lambda_k^j \overline{c}_j}{\sum_{j=1}^r \left(\Lambda_k^j \overline{c}_j\right)}$$
(2.23)

The last step in the algorithm is to compute the model conditioned state estimate. The PDF at this point is also a Gaussian sum. The IMM algorithm again approximates the PDF as a single moment-matched Gaussian,  $N[\hat{x}_{k|k}, P_{k|k}]$ , where:

$$\hat{x}_{k|k} = \sum_{j=1}^{r} \hat{x}_{k|k}^{j} u_{k}^{j}, \quad j = 1 \cdots r \quad \text{and}$$

$$P_{k|k} = \sum_{i=1}^{r} u_{k}^{j} \{ P_{k|k}^{j} + [\hat{x}_{k|k}^{j} - \hat{x}_{k|k}] \cdot [\hat{x}_{k|k}^{j} - \hat{x}_{k|k}]' \}, \quad j = 1, \cdots r .$$
(2.24)

In sum, we can see that there are two places (i.e. mixing and output) in the IMM algorithm in which a single moment-matched Gaussian approximates a Gaussian mixture.

#### 2.3.2 Interacting Multi-Pattern Probabilistic Data Association (IMPDA)

The IMPDA, that Hong et al. [45] developed, is a multirate extension of the IMMPDAF that operates both at full rate (1R) and one-third rate (1/3-R). The discussion below briefly summarizes the key features of the IMPDA and identifies the points within the algorithm that give rise to a multi-modal state PDF. A detailed derivation of the algorithm is found in [45].

The IMPDA uses the discrete wavelet transform to extract coherent information from measurements over multiple scans. This allows the IMPDA to accumulate information over several scans and provides better data association performance than single-scan algorithms such as IMMPDAF. While the IMMPDAF uses only distance information for data association, the IMPDA uses multi-patterns containing distance, directional, and maneuver information.

In order to generate these multi-patterns, the IMPDA takes a sequence of three trajectory points and then passes them through a series of two-tap, high-pass and low-pass discrete Haar wavelet transform filters. The output of the filter bank (as depicted in Figure 2.2) is a set of three patterns:

- Location pattern  $f_P$  (analogous to target position);
- Pointing pattern *f<sub>L</sub>* (analogous to velocity);
- Maneuvering pattern  $f_M$  (analogous to acceleration).

Since the multi-patterns are derived from target state vectors and are analogous to position, velocity, and acceleration, it is convenient to define them via (2.25).

$$\begin{bmatrix} f_{L} \\ f_{P} \\ f_{M} \end{bmatrix} \equiv \begin{bmatrix} x_{L} \\ x_{H} \\ x_{H^{2}} \end{bmatrix}$$
(2.25)

For non-maneuvering targets,  $f_P$  and  $f_L$  define the target pattern while for maneuvering targets  $f_P$ ,  $f_L$ , and  $f_M$  are required to define the target's kinematic behavior. The basic IMPDA uses two types of multirate models to represent the target kinematics. These are the Constant High-pass (CH) model, which is analogous to a Constant Velocity (CV) model, and the Constant High-High-pass (CH<sup>2</sup>), which is analogous to a Constant Acceleration (CA) target model. The task of these models is to map target patterns from one, 3-scan wide, time window into the next (Figure 2.3).



Figure 2.2 Extraction of Patterns from a Sequence



Figure 2.3 Pattern Mapping From One Window to Next

The CH model is defined via (2.26) below.

$$X_{k+3}^{1/3R} \equiv F_{k}^{1/3R} X_{k}^{1/3R} + \Gamma_{k}^{1/3R} u_{k}^{1/3R} = \begin{bmatrix} x_{k+3_{L}} \\ x_{k+3_{H}} \end{bmatrix} = \begin{bmatrix} I & 6I \\ 0 & I \end{bmatrix} \begin{bmatrix} x_{k_{L}} \\ x_{k_{H}} \end{bmatrix} + \begin{bmatrix} 5\sqrt{2}I & 3\sqrt{2}I & \sqrt{2}I \\ \sqrt{2}I & \sqrt{2}I & \sqrt{2}I \end{bmatrix} \begin{bmatrix} x_{k+1_{H^{2}}} \\ x_{k+2_{H^{2}}} \\ x_{k+3_{H^{2}}} \end{bmatrix}$$
(2.26)

The high-high-pass components are treated as zero-mean Gaussian disturbances with the following distributions

$$x_{k+1_{H^2}} \sim N(0, Q_{k+1_{H^2}}), \quad x_{k+2_{H^2}} \sim N(0, Q_{k+2_{H^2}}), \quad x_{k+3_{H^2}} \sim N(0, Q_{k+3_{H^2}}).$$
(2.27)

The one-third-rate CH measurements are defined as:

$$\begin{bmatrix} z_{k_L} \\ z_{k_H} \end{bmatrix} = \begin{bmatrix} 0.5z_{k-2} + z_{k-1} + 0.5z_k \\ -0.5z_{k-2} + 0.5z_k \end{bmatrix} + \begin{bmatrix} v_{k_L} \\ v_{k_H} \end{bmatrix} .$$
(2.28)

The equivalent one-third-rate measurement noise is:  $\begin{bmatrix} v_{k_L} \\ v_{k_H} \end{bmatrix} = N(0, R_{k_{LH}})$ , where:

$$R_{k_{LH}} = \begin{bmatrix} 0.25R_{k-2} + R_{k-1} + 0.25R_k & 0\\ 0 & 0.25R_{k-2} + 0.25R_k \end{bmatrix}.$$
 (2.29)

The  $CH^2$  model is defined via (2.30) below.

$$\begin{aligned} X_{k+3}^{1/3R} &\equiv F_{k}^{1/3R} X_{k}^{1/3R} + \Gamma_{k}^{1/3R} u_{k}^{1/3R} = \\ \begin{bmatrix} x_{k+3_{L}} \\ x_{k+3_{H}} \\ x_{k+3_{H^{2}}} \end{bmatrix} = \begin{bmatrix} I & 6I & 9\sqrt{2}I \\ 0 & I & 3\sqrt{2}I \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_{k_{L}} \\ x_{k_{H}} \\ x_{k_{H^{2}}} \end{bmatrix} + \begin{bmatrix} 18I & 8I & 2I \\ 6I & 4I & 2I \\ \sqrt{2}I & \sqrt{2}I & \sqrt{2}I \end{bmatrix} \begin{bmatrix} x_{k+1_{H^{3}}} \\ x_{k+2_{H^{3}}} \\ x_{k+3_{H^{3}}} \end{bmatrix} \end{aligned}$$
(2.30)

The high-high-pass components are treated as zero-mean Gaussian disturbances with the following distributions:

$$x_{k+1_{H^3}} \sim N(0, Q_{k+1_{H^3}}), \quad x_{k+2_{H^3}} \sim N(0, Q_{k+2_{H^3}}), \quad x_{k+3_{H^3}} \sim N(0, Q_{k+3_{H^3}}).$$
 (2.31)

The one-third-rate CH<sup>2</sup> measurement equation is:

$$\begin{bmatrix} z_{k_L} \\ z_{k_H} \end{bmatrix} = H \begin{bmatrix} x_{k_L} \\ x_{k_H} \\ x_{k_{H^2}} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_{k_L} \\ x_{k_H} \\ x_{k_{H^2}} \end{bmatrix} + \begin{bmatrix} v_{k_L} \\ v_{k_H} \end{bmatrix};$$
(2.32)

where 
$$\begin{bmatrix} z_{k_L} \\ z_{k_H} \end{bmatrix}$$
 and  $\begin{bmatrix} v_{k_L} \\ v_{k_H} \end{bmatrix}$  are previously defined in (2.28).

Since the patterns are only updated every three samples, the target positions at sample points between pattern updates are calculated via a standard full-rate Kalman filter.

The IMPDA (Figure 2.4) runs multiple parallel models and has a structure analogous to that of an IMMPDAF. Thus, like the IMM, the IMPDA results in target states that are Gaussian sums both after the mixing process and in the final output state. The IMPDA also models these Gaussian sums via a single moment-matched Gaussian.



Figure 2.4 IMPDA Algorithm Block Diagram

#### 2.3.3 Joint Probabilistic Data Association (JPDA)

The JPDA algorithm [2] is a multi-target extension of the well known probabilistic data association filter (PDAF) [3]. In both the PDAF and the JPDA, the posterior state PDF is a Gaussian mixture that is modeled via a single Gaussian. In the single-target PDAF the multi-modality is caused by non-persistent clutter. This clutter is generally modeled as uniformly distributed throughout the surveillance volume. JPDA, however, assumes multiple targets are present. If two (or more) targets are closely spaced then target measurements from one target may fall within the validation gate of its neighbor, resulting in persistent clutter. Since the JPDA

models multiple modes via a single Gaussian, the individual targets may coalesce into a single target.

The discussion below briefly summarizes key elements of JPDA and identifies where and how multimodality occurs. A detailed JPDA derivation is available in [2, 3, and 4].

JPDA operates under the following set of assumptions:

- There is a known number established targets that are being tracked;
- Tracking occurs in the presence of clutter;
- Measurements from one target may fall into the validation gate of another target over multiple scans and act as persistent interference;
- The targets follow a Markov process, which can be sufficiently described by an approximate conditional mean and covariance for each target;
- Each target has a state and measurement model.

JPDA thus takes the following basic approach to the multi-target tracking problem:

- Measurement to target track probabilities are calculated jointly across the targets;
- The association probabilities are calculated only for the current set of measurements and previous association hypotheses are not considered;
- The state estimates are computed separately for each target.

The key task in the JPDA algorithm is to compute the joint measurement-track association probabilities,  $P(\theta_k|z_{1:k})$ . Once the joint association probabilities are available, the marginal association probabilities are computed by summing over the joint events in which the marginal event occurs as shown in (2.33).

$$\beta_{jt} \equiv P(\theta_{jt} | z_{1:k}) = \sum_{\theta} P(\theta | z_{1:k}) \hat{\omega}_{jt}(\theta),$$

$$j = 1, \cdots m_{k}, \quad t = 0, 1, \cdots T$$
(2.33)

where:

 $\theta_{\rm jt}$  is a measurement-track association event;

 $\hat{\omega}_{jt} = 1$  if a measurement-track association,  $\theta_{jt}$ , for measurement *j* and track *t* is feasible and  $\hat{\omega}_{jt} = 0$  if not;

 $m_k$  = number of measurements at time *k*;

T = number of target tracks.

The values of  $\beta_{ji}$  then become the weighting factors that are used to calculate the combined innovation for each target, *t*.

$$\nu_{t_k} = \sum_{j=1}^{m_k} \beta_{jt_k} \nu_{jt_k} \,. \tag{2.34}$$

The combined innovation,  $v_{t_k}$ , is itself a Gaussian mixture of  $m_k$  Gaussian components having PDFs of  $N[v_{jt_k}, S_{t_k}]$ . Although (2.34) is a Gaussian mixture, the JPDA approximates the posterior state estimate PDF as a single moment matched Gaussian. The posterior mean of the state estimate of each target,  $t_i$  is thus computed via the standard Kalman filter equation:

$$\hat{x}_{t_{k|k}} = \hat{x}_{t_{k|k-1}} + K_{t_k} \, V_{t_k} \quad . \tag{2.35}$$

The posterior covariance for track t is composed of three covariance components:

$$P_{t_{kk}} = \beta_{t_{0_k}} P_{t_{kk-1}} + [1 - \beta_{t_{0_k}}] P_{t_{kk}}^c + \widetilde{P}_{t_k}$$
(2.36)

The first covariance component,  $\beta_{t_{0_k}} P_{t_{k|k-1}}$ , is due the fact that with probability  $\beta_{t_{0_k}}$ , none of the measurements are correct. The term  $P_{t_{k|k}}^c$  in the second covariance component is the covariance of the state updated with the correct measurement and is described by the standard Kalman filter covariance update equation:

$$P_{t_{k|k}}^{c} = P_{t_{k|k-1}} - K_{t_{k}} S_{t_{k}} K_{t_{k}}' .$$
(2.37)

The third component,  $\tilde{P}_{t_k}$ , is the spread of the innovations (analogous to the spread of the means discussed previously) and is defined by (2.38).

$$\widetilde{P}_{t_k} \equiv K_{t_k} \left[ \sum_{i=1}^{m_k} \beta_{t_{i_{k,k}}} \, \nu_{i_{t_k}} \, \nu'_{i_{t_k}} - \nu_{t_k} \, \nu'_{t_k} \, \right] K'_{t_k} \tag{2.38}$$

In the single-target PDAF, clutter is uniform over the surveillance region and is nonpersistent. Thus, the net contribution of the clutter to the state estimate mean is zero and a single, moment-matched Gaussian is a reasonable approximation of the true PDF. In the JPDA, however, if there is persistent clutter from another target then a single moment-matched Gaussian poorly represents the true PDF. This concept is illustrated in Figure 2.5. A consequence of using a single moment-matched Gaussian rather than the true multi-modal PDF is the JPDA track-coalescence phenomenon (when two closely spaced parallel tracks merge into a single track). [7,11]

In summary, we see that multi-modality occurs in the IMM, IMPDA, and JPDA tracking algorithms. All of these algorithms model a Gaussian mixture as a single, moment-matched Gaussian. This "simplification" often results in significantly greater tracking errors. In order to reduce these errors and obtain more accurate tracking, it is necessary to better model the actual multi-modal PDF.

#### 2.4 Estimation Techniques to Address Multimodality

The three standard techniques that address non-Gaussian PDFs (including Gaussian mixtures) are:

- Gaussian sum filters (GSF) [13,14];
- Grid-based methods [36, 65];
- Particle filters (PF) [36, 65].

As will be shown in upcoming sections, the GSF and grid-based methods suffer from several shortcomings that make them impractical for our purposes. Consequently, the particle filter is the technique of choice for this type of problem.



Figure 2.5 Moment-Matched Gaussian Representation of PDF in JPDA

#### 2.4.1 Approximate Grid-Based Methods

Grid-based methods use a discrete version of the Bayesian update equation (2.3). They can approximate the posterior density,  $p(x_k \mid z_{1:k})$ , if the state space is continuous but can be divided into a finite number,  $N_s$ , of discrete states  $\{x_k^i: i=1,...,N_s\}$ . The posterior density is computed via the method shown below.

Assume that the posterior PDF at time *k*-1 is defined as:

$$p(x_{k-1}|z_{1:k-1}) = \sum_{i=1}^{N_s} w_{k-1|k-1}^i \,\delta(x_{k-1} - x_{k-1}^i) \,. \tag{2.39}$$

The discrete prediction and update equations can then be expressed via (2.40-2.43).

$$p(x_k | z_{1:k-1}) = \sum_{i=1}^{N_s} w_{k|k-1}^i \,\delta(x_k - x_k^i) \qquad \text{(State Prediction)} \tag{2.40}$$

$$p(x_k | z_{1:k}) = \sum_{i=1}^{N_s} w_{k|k}^i \,\delta(x_k - x_k^i) \qquad (\text{State Update}) \qquad (2.41)$$

Where the weights are defined as:

$$w_{k|k-1}^{i} \equiv \sum_{j=1}^{N_{s}} w_{k-1|k-1}^{j} p\left(x_{k}^{i} | x_{k-1}^{j}\right)$$
(2.42)

$$w_{k|k}^{i} = \frac{w_{k|k-1}^{i} p(z_{k}|x_{k}^{i})}{\sum_{j=1}^{N_{s}} w_{k|k-1}^{j} p(z_{k}|x_{k}^{j})}$$
(2.43)

Thus, (2.40) is a discrete form of the Chapman-Kolmogorov equation while (2.41) is a discrete Bayesian update equation.

The approximate grid-based method suffers from two key drawbacks. First, the grid must be sufficiently dense in order to get an accurate representation of a continuous state space. This is computationally expensive because it requires a very large number of grid points as the dimension of the state space increases. The second drawback is that the state space must be predefined. Thus, the grid points cannot be concentrated so as to provide better resolution in high probability regions.

#### 2.4.2 Gaussian Sum Filter (GSF)

Sorenson and Alspach [13,14] developed the concept of the GSF to deal with nonlinear/non-Gaussian situations. The GSF makes use of the Gaussian sum approximation lemma, which states that any PDF, p(x), can be approximated as closely as desired by a weighted sum of Gaussian PDFs as shown below in (2.44).

$$p_{GS}(x) = \sum_{i=1}^{m} \alpha_i \ N(x - \mu_i, P_i) \cong p(x)$$
(2.44)

where:

$$lpha_i$$
 is a scalar weighting factor with  $\sum_{i=1}^m lpha_i = 1$  ;

and  $\mu_i$  and  $P_i$  are the mean and covariance, respectively, of the *i*th Gaussian term.

The parameters  $\alpha_i$ ,  $\mu_i$ , and  $P_i$  are chosen so that they minimize the  $L^k$  norm (*k* generally is equal to 2) between the actual density function, p(x), and the Gaussian sum approximation,  $p_{gs}(x)$ . This approximation can be made very accurate by choosing a large value for m, the number of Gaussian terms. Thus, a bank of parallel Kalman filters can represent a nonlinear/non-Gaussian system. A key drawback of the Gaussian sum approach is that the number of Gaussian terms, and hence the number of Kalman filters, increases at each time iteration and grows exponentially (referred to as the growing memory problem). This growth, if left unchecked, makes the GSF too expensive computationally.

Caputi [15,16] developed a modified Gaussian sum estimation technique that uses a fixed number of Gaussian sum terms and avoids the growing memory problem. Caputi's method is designed for systems with linear state and measurement equations but non-Gaussian measurement and process noise. His technique models the non-Gaussian noises as the sum of a zero mean Gaussian component and a semi-Markov bias term.

Tam and Hatzinakos [17,20] developed an adaptive Gaussian sum tracking algorithm for radar tracking. Their approach assumes that both process and measurement noises are Gaussian and state equations are linear. As was the case in the CMKF, their main goal is to deal with the effects of non-linear polar-Cartesian measurement transformation. In order to accomplish this, they use a GS approximation to compute the value of  $p(z_k | x_k)$ . The growing memory problem is dealt with by disregarding density functions with small  $\alpha_i$  coefficients and by combining densities that are statistically close (i.e. small Bhattacharyya distance). Since state equations are assumed to be linear, the Chapman-Kolmogorov equation in numerator in (2.4) is replaced by a Gaussian density whose mean is obtained by the Kalman filter state prediction (2.5) equation and whose covariance is defined by the Kalman covariance prediction equation (2.6).
The drawback of all of these approaches, however, is that they retain only a fixed number of Gaussian mixture components. Thus, they are not well suited for modeling a target state with a multi-modal PDF that potentially has a large number of modes.

## 2.4.3 Particle Filter (PF)

Although, Monte-Carlo methods for state estimation have been available for over 30 years, Gordon, et al. presented the first true particle filter in 1993 [26]. The PF is a sequential Monte-Carlo technique that produces, at each time instant k, a cloud of  $N_P$  particles that approximates estimates the probability density function of the posterior target state,  $p(x_k | z_{1:k})$ . Thus, by drawing appropriately weighted samples from this "cloud" one can solve the Bayesian estimation equation (2.3) and obtain the state estimate. As  $N_P$  becomes very large, the density approximate a multi-modal PDF.

Another PF benefit is that non-linear states/measurements and non-Gaussian noises can be handled without resorting to linearization and/or partial derivatives (i.e. Jacobians). The major drawback of PF methods is that a very large number of particles may be required in order to accurately represent the target state PDF.

The particle filter solves the Bayesian estimation equation by approximating the posterior PDF via the discrete weighted sum in (2.45).

$$p(x_{0:k} \mid z_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \, \delta(x_{0:k} - x_{0:k}^i).$$
(2.45)

The individual weights,  $w_k^i$ , are computed by applying the principle of importance sampling. Since it is difficult or impossible to directly sample  $p(x_{0:k} | z_{1:k})$ , we define a density  $\pi(x)$  that can be evaluated and that is chosen such that  $p(x) \propto \pi(x)$ . Additionally, let  $x^{i} \sim q(x), i=1,...N_{p}$  be samples that are drawn from a proposal q(x) that is referred to as the importance density.

Since  $p(x) \propto \pi(x)$ , the individual weights of each normalized particle are then defined as:

$$w_k^i \propto \frac{\pi(x^i)}{q(x^i)} \Longrightarrow w_k^i \propto \frac{p(x_{0:k}^i | z_{1:k})}{q(x_{0:k}^i | z_{1:k})}.$$
 (2.46)

The importance density is chosen so that it can be factorized as:

$$q(x_{0:k}|z_{1:k}) \equiv q(x_k|x_{0:k-1}, z_{1:k}) q(x_{0:k-1}|z_{1:k-1}).$$
(2.47)

This allows us to obtain samples from the current state by augmenting samples from the previous state.

To obtain the weight update equation, we first express the posterior PDF,  $p(x_{0:k} | z_{1:k})$ , in terms of  $p(x_{0:k-1} | z_{1:k-1})$ ,  $p(z_k | x_k)$ , and  $p(x_k | x_{k-1})$  to obtain:

$$p(x_{0:k} | z_{1:k}) = p(x_{0:k} | z_{k}, z_{1:k-1}) = \frac{p(z_{k} | x_{0:k}, z_{1:k-1})p(x_{0:k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} = \frac{p(z_{k} | x_{0:k}, z_{1:k-1})p(x_{k}, x_{0:k-1} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} = \frac{p(z_{k} | x_{0:k}, z_{1:k-1})p(x_{k} | x_{0:k-1}, z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | x_{k})p(x_{k} | x_{k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | x_{k})p(x_{k} | x_{k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | x_{k})p(x_{k} | x_{k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | x_{k})p(x_{k} | x_{k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | x_{k})p(x_{k} | x_{k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(x_{0:k-1} | z_{1:k-1}) = \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}) + \frac{p(z_{k} | z_{1:k-1})}{p(z_{k} | z_{1:k-1})} \cdot p(z_{k} | z_{1:k-1}$$

State evolution is assumed to be a first order Markov process. Consequently, the conditioning term,  $z_{1:k-1}$ , can be dropped from the likelihood function,  $p(z_k | x_k)$ , and the transition prior PDF,  $p(x_k | x_{k-1})$ . Since  $p(z_k | z_{1:k-1})$  is simply a normalizing constant,  $p(x_{0:k} | z_{1:k})$  is proportional to the quantity in (2.49).

$$p(x_{0:k} \mid z_{1:k}) \propto p(z_k \mid x_k) p(x_k \mid x_{k-1}) p(x_{0:k-1} \mid z_{1:k-1})$$
(2.49)

We now note that If we now substitute (2.47) and (2.49) into (2.46) and simplify, we obtain the recursive particle weight update equation (2.50).

$$w_{k}^{i} \propto \frac{p(z_{k} | x_{k}^{i}) p(x_{k}^{i} | x_{k-1}^{i}) p(x_{0:k-1}^{i} | z_{1:k-1})}{q(x_{k}^{i} | x_{0:k-1}^{i}, z_{1:k}) q(x_{0:k-1}^{i} | z_{1:k-1})}$$

$$= w_{k-1}^{i} \frac{p(z_{k} | x_{k}^{i}) p(x_{k}^{i} | x_{k-1}^{i})}{q(x_{k}^{i} | x_{0:k-1}^{i}, z_{1:k})}$$

$$(2.50)$$

If we assume that the importance density, q(x), also describes a first order Markov process, then the importance density depends only on the previous state,  $X_{k-1}$ , and the current measurement,  $Z_k$ . In most tracking scenarios, only the current filtered state estimate,  $X_k$ , is required. We can therefore discard the target path,  $X_{0:k-1}$ , and the observation history,  $Z_{1:k-1}$ . The particle weight update equation can then be expressed via (2.51).

$$w_{k}^{i} \propto w_{k-1}^{i} \frac{p(z_{k} \mid x_{k}^{i}) p(x_{k}^{i} \mid x_{k-1}^{i})}{q(x_{k}^{i} \mid x_{k-1}^{i}, z_{k})}$$
(2.51)

The weights are then normalized by dividing each particle weight by the sum of the particle weights at a given sample time k.

$$w_k^i = \frac{w_k^i}{\sum_{\forall i} w_k^i}$$
(2.52)

The posterior filtered density,  $p(x_k \mid z_{1:k})$ , is now calculated as:

$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N_P} w_k^i \, \delta(x_k - x_k^i).$$
(2.53)

The particle filtering technique described above is known as the sequential importance sampling (SIS). Although, the SIS is simple to implement, it suffers from the "Degeneracy Phenomenon". Over time, the variance of the particle weights increases. This eventually results in a situation in which all but one particle has negligible weight. A common technique to reduce this degeneracy is to resample when the effective sample size,  $N_{eff}$ , falls below a predefined threshold (such as  $N_{eff} < 0.5 N_P$ ). Although  $N_{eff}$  cannot be directly computed, it can be approximated as:

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N_{p}} (w_{k}^{i})^{2}}.$$
(2.54)

The other key issue in particle filtering is choosing an appropriate importance density. The simplest choice of importance density is to use the transition prior state density,  $p(x_k^i | x_{k-1}^i)$ . When the prior is used as the importance density, the particle update equation (un-normalized) can be expressed as:

$$w_{k}^{i} \propto w_{k-1}^{i} \frac{p(z_{k} \mid x_{k}^{i}) p(x_{k}^{i} \mid x_{k-1}^{i})}{p(x_{k}^{i} \mid x_{k-1}^{i})} = w_{k-1}^{i} p(z_{k} \mid x_{k}^{i}).$$
(2.55)

Gordon's particle filter, which is known as the bootstrap particle filter (BPF) or Sampling Importance Resampling (SIR) filter uses the prior as the importance density. In addition, the SIR resamples at every time increment and sets the resampled particle weight to  $1/N_p$ . This removes the dependency of the current particle weight to the previous particle weight. Thus, the unnormalized particle weight is simply the value of the measurement likelihood function, evaluated at the predicted particle,  $x_k^i$ , yielding:  $w_k^i \propto = p(z_k | x_k^i)$ . With this in mind, the SIR algorithm can be summarized as follows:

- Initialization: Assume that the initial state PDF, measurement and process noise PDFs, and the measurement likelihood function are known.
- Sampling and Prediction: Obtain  $N_P$  samples from the posterior density available at time *k-1*:  $p(x_{k-1} | z_{k-1})$  and propagate these points through the system model,  $f_{k-1}(x_{k-1}) + w_{k-1}$ , and obtain a collection of "predicted points",  $x_k^i$ .

• Importance Weight Calculation: Upon receipt of a measurement  $z_k$ , evaluate the likelihood of each prior sample point and thus obtain a normalized weight,  $w_k^i$ , for each

sample: 
$$w_{k}^{i} = \frac{p(z_{k} | x_{k}^{i})}{\sum_{j=1}^{N_{P}} p(z_{k} | x_{k}^{j})};$$
 (2.56)

where:  $p(z_k | x_k^i)$  is the likelihood function of the current measurement, conditioned on the "predicted" particle.

 Resampling: The posterior state density function is then obtained by sampling (with replacement) from the set of points defined by the right hand side (RHS) of the equation below.

$$p(x_{k} | z_{k}) \approx \sum_{i=1}^{N_{p}} w_{k}^{i} \, \delta(x_{k} - x_{k}^{i})$$
(2.57)

After resampling, all of the particle weights are set to  $1/N_{p}$ .

• **Filter output:** The state estimate is typically chosen to be the mean value of the particle states. Since the particle weights are now equal after resampling, the state mean is:

$$\hat{x}_{k|k} = \frac{1}{N_P} \sum_{i=1}^{N_P} x_k^i \quad .$$
(2.58)

The SIR PF is popular because it is easy to implement. Thus, it has been used in numerous non-linear/non-Gaussian filtering applications. It does, however, sometimes require a very large number of particles in order to work well. This situation occurs when the prior density and likelihood function have only a small region of overlap [36]. A significant amount of research has been done on particle filtering since the introduction of the SIR PF. The bulk of this research has focused on improving the performance or reducing the computational cost of the basic SIR filter and identifying new applications for the PF.

Djuric, et al. [30] combined the PF with a Gaussian sum approach to develop a hybrid Gaussian sum particle filter (GSPF) that used a small, fixed number of Gaussian sum terms (6 to 16) and relatively small number of particles ( $N_p$  =100). Their GSPF implementation was applied against a one-dimensional system that had highly non-linear state and measurement equations. Additionally, process and measurement noises were non-Gaussian. Their results indicated that the hybrid GSPF offered much lower mean squared errors (MSE) than a GS-only filter with the same number of GS terms.

Arulampalam, et al. [36] presented several PF algorithm variants that offer some advantages over the traditional SIR PF. These PFs, which include the Auxiliary Sampling Importance Resampling Filter (ASIR), Regularized Particle Filter (RPF), and the Likelihood Particle Filter (LPF), sometimes offer better RMSE performance than the conventional SIR. Hue, et al. [33] have recently addressed the multi-target tracking via the PF and have developed the Multi-target Particle Filter (MTPF) that incorporates a Markov-chain Monte-Carlo (MCMC) technique known as Gibbs sampling. Blom et al. [41], Frank et al. [43], Schultz et al. [50], and Vermaak et al. [59] have also focused on developing multi-target PF implementations.

Farina, et al. [40] compared the performance and computational costs of the EKF, UKF, CADET (Covariance Analysis Describing Function Technique), and SIR particle filter against the theoretical Cramer-Rao lower bounds (CRLB) of estimation error. Their example used non-linear measurement and process models with Gaussian process and measurement noises. All of the estimation methods were consistent and produced good estimates. The particle filter, however, (and also the CADET algorithm) required over two orders of magnitude of computations than did the EKF or UKF.

# 2.5 Multiple Model Particle Filter (MMPF)

Another area of PF research is in the tracking of maneuvering targets via multiple switching process models [6]. As was mentioned previously, the Kalman-based IMM approximates a multi-modal state PDF via a single moment-matched Gaussian. The particle

30

filter, however, is not restricted to Gaussian densities. McGinnity and Irwin adapted the multiple model concept to particle filtering and developed the first MMPF [28, 29]. Their MMPF uses an alternate form of the Bayesian estimator in which branched prior densities are merged into *r* model conditioned densities:

$$p(x_{k}|M_{k+1}^{i}, z_{1:k}) = \sum_{j=1}^{r} p(x_{k}|M_{k}^{j}, z_{1:k}) \cdot p(M_{k}^{j}|M_{k+1}^{i}, z_{1:k}) .$$
(2.59)

The second term of the right hand side (LHS) is expanded out by using Bayes' rule to obtain the following model probability in (2.60).

$$\frac{p(M_{k}^{j}|M_{k+1}^{i}, z_{1:k}) = \frac{p(M_{k+1}^{i}|M_{k}^{j}, z_{1:k}) p(M_{k}^{j}|z_{1:k})}{p(M_{k+1}^{i}|z_{1:k})} = \frac{h^{ij} p(M_{k}^{j}|z_{1:k})}{p(M_{k+1}^{i}|z_{1:k})}$$
(2.60)

The denominator,  $p(M_{k+1}^{i} | z_{1:k})$ , is a normalizing term and is simply the sum of the numerator over all values of *j*. The posterior state PDF at time *k*+1 is the given by the sum of *r*, model conditioned, posterior PDFs as shown below in (2.61).

$$p(x_{k+1}|z_{1:k+1}) = \sum_{i=1}^{r} p(x_{k+1}|M_{k+1}^{i}, z_{1:k+1}) \cdot p(M_{k+1}^{i}|, z_{1:k+1})$$
(2.61)

A key difference between the MMPF and the standard bootstrap PF is that each particle is an ordered pair that consists of the state,  $x_k$ , and a mode index  $M_k^j$ , j = 1, ..., r.

The MMPF includes a mode mixing step in which particle modes transition from one mode to another according to a Markov transition matrix,  $h^{ij}$ . This Markov transition is implemented via a "roulette wheel" sampling method in which the "size" of each pattern on the wheel is proportional to its probability.

The predicted state for each particle,  $[x_k^i, M_k^j]$ , is obtained by applying the process model that corresponds to the model indicated for particle *i*. The importance weight calculation, resampling, and computation of the posterior mean are the same as in the SIR PF. The posterior

model probabilities are automatically calculated during the resampling process since the particles for each mode are resampled according to their posterior probability.

#### 2.6 Initial Multirate Particle Filter Efforts

Hong and Cui [52] further extended the multirate estimation concept to multiple-model particle filtering techniques and developed the multirate interacting multiple model particle filter (MRIMM-PF). The basic idea behind MRIMM-PF is that targets spend most of their time in CV motion and that target maneuvers are relatively infrequent. The MRIMM-PF exploits this fact by developing a multirate algorithm which consists of a non-maneuvering third-rate model that is updated every three scans while the maneuvering full-rate models are updated at every scan.

A typical target's trajectory is CV for most of the track life. Thus, on average, most of the particles will be assigned to the non-maneuvering model. Since this non-maneuvering model is updated once every three scans, the average number of particles in the MRIMM-PF is substantially less (approximately 46%) than that required for a full-rate MMPF for a comparable level of RMSE performance. This results in less computational cost, since cost is O(N) in particle filters.

# 2.7 Maintaining Multi-Modality in Particle Filters

Particle filter-based algorithms are theoretically well suited for dealing with multi-modal PDFs. In reality, however, low-weight particles are seldom resampled. Weak modes are, thus, often lost after a few iterations. This presents a significant problem if the weak mode is due to the presence of another target that we wish to track. Vermaak, et al. [44] have developed a technique to maintain multimodality by modeling the target distribution as a non-parametric mixture model. Each mixture is modeled via a separate particle filter that interacts with the other particle filters only during the computation of mixture weights. Their algorithm uses K-means clustering to recompute the mixture representation during the tracking scenario as targets appear and disappear.

## 2.8 Measurement Gating With Multi-Modal Likelihood Functions

The issue of "how to define a measurement validation gate?" arises in particle filters because there is no direct analog to the validation gate found in Kalman filter-based trackers. In single target tracking scenarios where false alarms are present, measurement gating is necessary to reduce the possible measurement-track association hypotheses to a manageable level since the number of hypotheses equals the number of measurements plus one (i.e. an additional hypothesis is required for the null target case). Gating becomes even more critical in multi-target scenarios because the number of possible association hypotheses grows exponentially as the number of targets and false alarms increases.

A conventional Kalman tracker uses the Gaussian innovation covariance,  $S_{k+1}$ , to define a validation gate around the predicted measurement. Typically, the gate excludes measurements that fall outside the 3-4 sigma range. Since innovation covariance is not available in particle filters, some other gating scheme is required. Marrs, et al. [47] developed a non-parametric efficient score function by computing the expected log-likelihood from known measurement and clutter statistics. Vermak et al. [59] also developed a gating mechanism that models the prior particle set as a Gaussian and then incorporates a particle filter analog of the innovation covariance matrix from this Gaussian.

## 2.9 Particle Filter Track Before Detect (TBD-PF)

Conventional target detection schemes set a detection threshold to determine if a sensor return represents a potential target or is the result of noise. The dilemma of this method is that if the threshold is set too high then a target may not be detected. Conversely, if the threshold is set too low then many false alarms will be generated. Thus, the detection threshold is often set as a practical compromise between a high probability of detection ( $P_D$ ) and an acceptable probability of false alarm ( $P_{FA}$ ). In a low-SNR environment, achieving a practical compromise is problematic. TBD techniques eliminate the detection threshold and simultaneously track and

detect targets. This allows tracking of targets having much lower SNR values than is possible with standard detection-then-track schemes.

In recent years, particle filtering techniques have been applied to the TBD problem. Particle filters are an attractive choice because measurement likelihood functions often have Rayleigh or Ricean PDFs at low SNR levels. A basic single target TBD-PF algorithm was initially proposed by Salmond et al. [54]. Rollason and Salmond [55] then developed a TBD-PF for targets with unknown amplitude. Boers and Driessen further extended TBD-PF concepts and developed a multi-target TBD-PF [63]. Musick et al. [32] implemented a bootstrap TBD-PF algorithm for an electro-optical (EO) sensor with a Rayleigh likelihood function. Oii et al. [57] adapted Musick's algorithm by deriving an optimal proposal density which used Rao Blackwellization. Ristic [56] designed a TBD-PF tracker that used an EO sensor with Gaussian likelihood function. His algorithm tracked targets down to an SNR of 5 dB and contained the following elements:

- Explicitly probability of track computation;
- Particle existence state determined via Markov transition (existence states = newborn, existing, and dead);
- A single PF was used for all existence states.

Rutten et al. [58] built upon and improved Ristic's algorithm. Rutten's TBD-PF algorithm modeled a radar sensor that used a Ricean-Rayleigh measurement model (target plus noise PDF is Ricean while noise-only PDF is Rayleigh). His algorithm differed from Ristic's in that it explicitly included the track existence probability in the target state vector and used separate particle filters to compute the newborn and existing densities. Although Rutten's TBD-PF implementation was more complex, it could track CV targets down to an SNR of 3 dB.

# 2.10 Spatial-Domain Multi-Resolution Particle Filtering (SD-MRES-PF)

SD-MRES-PF is a data compression and particle count reduction technique that Hong and Wicker [71] recently developed. SD-MRES-PF (like multirate particle filtering) uses a DWT to

34

decompose a data sequence into LP and HP components. Unlike multirate particle filters, MRES-PF works at full-rate and decomposes the sampled uni-resolution (uni-res) PDF into LP and HP PDF components. The HP PDF components are then compared against a pre-defined minimum threshold. This process is illustrated in Figure 2.6 and Figure 2.7. Component samples that fall below this threshold are then removed. In practice, many of the data points in the HP components have relatively small values and are "noise-like" in nature. Thus, removing these small "noise-like" components allows us to reconstruct the uni-res PDF with fewer particles without significantly degrading particle filter RMSE performance.

The PDF components are then transformed with an appropriate IDWT algorithm in order to reconstruct a "data compressed" uni-res PDF that has fewer particles than the original. The amount of "data compression" varies according to the size of the threshold. A larger threshold results in more compression and fewer particles. Conversely, a smaller threshold produces the opposite effect. The new, reduced, particle set is then propagated and updated via a SIR-PF.

The SD-MRES-PF features two methods to implement multiresolutional particle filtering. These are termed as the implicit and explicit methods. The implicit method embeds the wavelet transformation into a complicated variable structure but does not require an inverse transform to reconstruct the uni-res density. In contrast, the explicit method uses a simple variable structure but requires an inverse transform for uni-res density reconstruction.

The Hong and Wicker SD-MRES-PF was a proof-of-concept model that only operated over one update cycle. The original uni-res PDF in the SD-MRES-PF was generated as histogram PDF that required 5000 samples to generate 1000 sampled PDF points. Consequently, it is not suitable as a multiple update particle filtering algorithm because the PDF generation process would negate any particle savings obtained from the multi-resolution processing. An extended SD-MRES-PF that operates over multiple time increments is presented in Section-5.

35



Figure 2-6 Original Uni-Res PDF and Level-1/2 Multi-Res Decompositions (No Thresholding)



Figure 2-7 Level-1/2 Decompositions (Thresholded)

# 3 MULTIRATE - MULTIPLE MODEL PARTICLE FILTER (MRMMPF)

The MRMMPF algorithm was first introduced in the initial proposal for this dissertation and forms the building block of the MRMMPF-TBD algorithm that is described in the Section-4. The MRMMPF described in the following paragraphs was thus intended as a "proof of concept" in order to demonstrate the advantages of multirate particle filtering vs. full-rate particle filtering and Kalman-based tracking algorithms.

The MRMMPF combines elements of the MMPF and the IMPDA. It uses a multi-pattern multiple model particle filter to compute state estimates at 1/3-rate (1/3-R) and conventional MMPFs to compute state estimates at full-rate (1-R). Within the MRMMPF, the 1/3-R MMPF and the 1-R MMPF are run in parallel. The 1/3-R MMPF computes estimates at every third sample increment (i.e. k = 3,6, 9,...) while two cascaded 1-R MMPFs compute estimates at the intermediate points (k = 1,2,4,5,...).

The original MMPF algorithm was designed for single target tracking in a zero-clutter environment (i.e. zero false alarms). The MRMMPF, however, is intended to function in the presence of false alarms. Thus, the measurement likelihood functions in both the 1/3-R MMPF and the 1-R MMPF components of the MRMMPF were modified to use a PDA-type likelihood function that will be described later in this section.

Both multirate tracking (via Kalman filtering) and multiple model particle filtering have been addressed in previous research. These two techniques have yet, however, to be combined into an integrated tracking algorithm that tracks targets in the presence of clutter. Hence, the rationale behind the MRMMPF is to combine the strengths of the aforementioned algorithms. These strengths are:

- The IMPDA's ability extract coherent information from measurements over multiple scans;
- The ability of the MMPF to handle non-linear/non-Gaussian PDFs.

It will be shown that a bootstrap (i.e. SIR) PF implementation of MRMMPF outperforms the IMMPDAF, IMPDA, and the MMPF.

#### 3.1 MRMMPF Theoretical Description and Design

A four-pattern/four model bootstrap version of the MRMMPF algorithm was implemented according to the block diagram shown in Figure 3.1. The basic components of the MRMMPF algorithm are:

- 1. 1/3-Rate MMPF Initialization;
- 2. Full-rate MMPF Initialization;
- 3. 1/3-Rate Mixing;
- 4. 1/3-Rate MMPF;
- 5. Full-rate MMPF;
- 6. Full-rate state vector output.

Note: In the remainder of this dissertation, one-third-rate variables will be denoted by the "1/3R" superscript (e.g.  $\hat{x}_{k/k}^{1/3R}$ ). Variables without the "1/3R" superscript are assumed to be full-rate (e.g.  $\hat{x}_{k/k}$ ). Additionally, particles will be annotated with a subscript to indicate whether they are predicted ( $x_{k/k-1}^i$ ) or posterior ( $x_{k/k}^i$ ) particles.



Figure 3.1 MRMMPF Algorithm Block Diagram

#### 3.1.1 1/3-Rate MMPF Initialization

The proof-of-concept MRMMPF does not include a track initiation function and assumes that the initial 1/3-rate state PDF is known. This initial PDF is assumed to be Gaussian with a mean vector,  $\mu_0^{1/3R}$ , and covariance matrix  $P_0^{1/3R}$ . No information (i.e. diffuse prior) is assumed to be available regarding initial pattern probabilities,  $p(M_0^{j_{1/3R}}), j=1,\dots,4$ . Thus, each pattern probability is set to:  $p(M_0^{j_{1/3R}}) = \frac{1}{4}$ .

The 1/3-R measurement noise PDF is also assumed to be Gaussian and is the same as in the IMPDA:  $\begin{bmatrix} v_{k_L} \\ v_{k_H} \end{bmatrix} = N(0, R_{k_{LH}})$ , where:

$$R_{k_{LH}} = \begin{bmatrix} 0.25R_{k-2} + R_{k-1} + 0.25R_k & 0\\ 0 & 0.25R_{k-2} + 0.25R_k \end{bmatrix};$$
(3.1)

and where:  $R_k$  is the 1-R measurement covariance of the sensor.

Based on the initial state PDF and pattern probabilities, we first generate an initial set of  $N_P$  particles:  $[x_0^{i_{1/3R}}, M_0^{j_{1/3R}}], j=1,...,4$ . As was the case with the MMPF, each 1/3-R particle is an ordered pair that consists of a state vector,  $x_0^{i_{1/3R}}$ , and its associated pattern index,  $M_0^{j_{1/3R}}$ . Since the probability of each pattern is 0.25,  $\frac{N_P}{4}$  particles are assigned to each pattern. The state vectors of each particle,  $x_0^{i_{1/3R}}$ , are obtained by drawing  $N_P$  values from the following random vector:

$$x_0^{i_{1/3R}} = \mu_0^{1/3R} + w_0^{1/3R}, \ i = 1, \dots, N_P$$
(3.2)

where  $w_0^{1/3R}$  is the 1/3 rate process noise vector.

Since the initial covariance matrix,  $P_0^{1/3R}$ , is assumed diagonal, the process noise vector can be obtained taking the square roots of the variance components (i.e. the main diagonal) and multiplying the resulting matrix by a zero-mean, unity variance random vector:

$$w_0^{1/3R} = \begin{bmatrix} \sqrt{p_{0_{1,1}}^{1/3R}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{p_{0_{2,2}}^{1/3R}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{p_{0_{3,3}}^{1/3R}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{p_{0_{4,4}}^{1/3R}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{p_{0_{5,5}}^{1/3R}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{p_{0_{6,6}}^{1/3R}} \end{bmatrix} \cdot V_N$$
(3.3)

where

 $p_{0_{i,j}}^{1/3R}$  = *i*th row, *j*th column entry in the  $P_0^{1/3R}$  matrix;

 $V_{\rm N}$  = 6x1 random vector whose elements are random variables distributed ~ N[0,1].

## 3.1.2 Full-Rate MMPF Initialization

The full-rate MMPF is initialized by transforming the 1/3-R MMPF particles to 1-R. This is accomplished via a set of inverse discrete wavelet transform (IDWT) matrices. The IDWT matrices used are the same ones found in the IMPDA and are designated as  $T_{CV}^{-1}$  and  $T_{CA}^{-1}$ . The  $T_{CV}^{-1}$  IDWT converts 1/3-R particles with constant-high-pass (CH) model indices (i.e. CH<sup>2</sup> component = 0) into 1-R constant velocity (CV) particles in which the acceleration components are zero, as shown in (3.4).

$$x_{k}^{i} = T_{CV}^{-1} x_{k}^{i_{1/3R}} = \begin{bmatrix} 2I & -2TI & 0\\ 0 & TI & 0\\ 0 & 0 & I \end{bmatrix}^{-1} x_{k}^{i_{1/3R}}$$
(3.4)

where:

T = scan period; I = 2x2 identity matrix; 0 = 2x2 matrix of zeros.

Correspondingly,  $T_{CA}^{-1}$  transforms 1/3-R particles with constant-high-high-pass (CH<sup>2</sup>) model indices (i.e. non-zero CH<sup>2</sup> component) into 1-R constant acceleration (CA) particles:

$$x_{k}^{i} = T_{CA}^{-1} x_{k}^{i_{1/3R}} = \begin{bmatrix} 2I & -2TI & 1.5T^{2}I \\ 0 & TI & -T^{2}I \\ 0 & 0 & \frac{\sqrt{2}}{2}T^{2}I \end{bmatrix}^{-1} x_{k}^{i_{1/3R}}.$$
(3.5)

#### 3.1.3 1/3-Rate Mixing

The MRMMPF algorithm works on a three scan update cycle. Thus, when describing the algorithm, we will assume that the update cycle starts at t = k-3. For each 1/3-R particle,  $[x_{k-3|k-3}^{i_{1/3R}}, M_{k-3|k-3}^{j_{1/3R}}]$ , we generate a new particle,  $[x_{k-3|k-3}^{i_{1/3R}}, M_{k|k-3}^{j_{1/3R}}]$ , with particle number *i* and pattern index *j* (Note: The particle number, *i*, is different from pattern index *i*). As with the IMMPDAF, IMPDA, and MMPF, we assume that the mode jump is a Markov process with known probability transition matrix  $h^{ij}$ . The "post-mixing" model index,  $M_{k|k-3}^{j_{1/3R}}$ , is then obtained by applying the switching Markov chain with transition probability  $h^{ij}$  to  $M_{k-3|k-3}^{1/3R}$ . If  $M_{k-3|k-3}^{1/3R} = i$ , then  $M_{k|k-3}^{1/3R}$  will be set to *j* with a probability  $h^{ij}$ .

This Markov transition is implemented via a "roulette wheel" sampling method in which the "size" of each pattern on the wheel is proportional to its probability. Thus, if  $M_{k-3|k-3}^{1/3R} = i$  and  $u_n$  is a uniformly distributed number from (0,1], then  $M_{k|k-3}^{1/3R}$  is chosen as the value of *s* where:

$$\sum_{j=1}^{s-1} h^{ij} < u_n \le \sum_{j=1}^{s} h^{ij} \text{ and } \sum_{j=1}^{0} h^{ij} \equiv 0.$$
(3.6)

This concept is somewhat difficult to visualize and can best be described by the following example. Assume that we have the following Markov transition matrix:

$$h^{ij} = \begin{bmatrix} 0.91 & 0.04 & 0.04 & 0.01 \\ 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.85 & 0.05 \\ 0.01 & 0.04 & 0.04 & 0.91 \end{bmatrix}$$

The previous mode, *i*, at *k*-3 is denoted by the matrix rows while the new mode, *j*, at time *k* is denoted by the matrix columns. Assume that at *k*-3, the old mode is: i=1. We now generate a

uniformly distributed random number:  $\mu_n \sim (0,1]$  and obtain the following potential mode transition scenarios

- If  $\mu_n \leq 0.91$ , then mode j=1.
- If  $0.91 < \mu_n \le (0.91 + 0.04)$ , then mode *j*=2.
- If  $0.95 < \mu_n \le (0.91 + 0.04 + 0.04)$ , then mode *j*=3.
- Finally, If  $0.99 < \mu_n \le (0.91 + 0.04 + 0.04 + 0.01)$ , then mode *j=4*.

A similar argument applies if mode i = 2,3 or 4.

#### 3.1.4 1/3-Rate MMPF

# 1/3-R State Propagation: The 1/3-R MMPF functions in an analogous manner to the 1-R

MMPF. After the mixing process, each particle,  $[x_{k-3|k-3}^{i_{1/3R}}, M_{k|k-3}^{j_{1/3R}}]$ , is propagated through a dynamic system model,  $f_{M_{k|k-3}}(x_{k-3}^{1/3R}) + w_{M_{k|k-3}}^{1/3R}$ , that is based on its pattern index,  $M_{k|k-3}^{j_{1/3R}}$ . The 1/3-R state transition matrix,  $f_{M_{k|k-3}}(x_{k-3}^{1/3R})$ , operates over three time steps and depends on the specific pattern and model index. Patterns 1 and 4 both use the CH<sup>2</sup> model and will thus use the CH<sup>2</sup> 1/3-R state transition matrix:

$$f_{M_{k|k-3}}^{1/3R^{CH^2}}(x_{k-3}^{1/3R}) = \begin{bmatrix} I & 6I & 9\sqrt{2}I \\ 0 & I & 3\sqrt{2}I \\ 0 & 0 & I \end{bmatrix}.$$
(3.7)

Conversely, Patterns 2 and 3 both use the CH model and will thus use the CH 1/3-R state transition matrix:

$$f_{M_{k|k-3}}^{1/3R^{CH}}(x_{k-3}^{1/3R}) = \begin{bmatrix} I & 6I & 0\\ 0 & I & 0\\ 0 & 0 & 0 \end{bmatrix}.$$
(3.8)

The process noise vector is obtained from the 1/3-R process noise covariance matrix,  $Q_{M_{kk-3}}^{1/3R}$  that corresponds the model (either CH or CH<sup>2</sup>) associated with a given pattern. The value of  $Q_{M_{kk-3}}^{1/3R}$  is the same as that developed for the IMPDA. Thus, for the CH case:

$$Q_{M_{k|k-3}}^{1/3R^{CH}} = \Gamma_{k-3}^{1/3R^{CH}} \sigma_{W_{1/3R}}^{2} \Gamma_{k-3}^{\prime 1/3R^{CH}}$$
(3.9)

where:

$$\Gamma_{k-3}^{1/3R^{CH}}$$
 = 1/3 rate CH noise gain, previously defined in (2.26);  
 $\sigma_{W_{1/3R}}^{2}$  = 1/3 rate process noise variance.

The 1/3 rate process noise variance,  $\sigma_{_{W_{1/3R}}^{CH}}^2$ , in turn derived from the 1-R process noise variance,  $\sigma_{_w}^2$  via the following transformation:

$$\sigma_{w_{1/3R}}^{2} = T_{\sigma^{CH}} \sigma_{w}^{2} T_{\sigma^{CH}}^{\prime}$$
(3.10)

where:

$$T_{\sigma^{CH}} = \begin{bmatrix} \frac{T^2}{2\sqrt{2}}I & 0 & 0\\ 0 & \frac{T^2}{2\sqrt{2}}I & 0\\ 0 & 0 & \frac{T^2}{2\sqrt{2}}I \end{bmatrix}$$
(3.11)

and:

T = scan time

I = 2x2 identity matrix.

For the CH<sup>2</sup> case,  $\Gamma_{k-3}^{1/3R^{CH^2}}$  is the 1/3 rate CH<sup>2</sup> noise gain shown in (2.30). The 1/3 rate process noise variance,  $\sigma_{w_{1/3R}^{CH^2}}^2$ , is then expressed as:

$$\sigma_{W_{1/3R}}^{2} = T_{\sigma^{CH^{2}}} \sigma_{W}^{2} T_{\sigma^{CH^{2}}}^{\prime}$$
(3.12)

where:

$$T_{\sigma^{CH^2}} = \begin{bmatrix} \frac{T^2}{4}I & 0 & 0\\ 0 & \frac{T^2}{4}I & 0\\ 0 & 0 & \frac{T^2}{4}I \end{bmatrix}.$$
(3.13)

Once  $Q_{M_{k|k-3}}^{1/3R}$  is known,  $w_{M_{k|k-3}}^{1/3R}$  is obtained via eigendecomposition as follows:

$$w_{M_{k|k-3}}^{1/3R} = \mathbf{X}^{1/3R} D^{1/3R} V_N^{1/3R} = \mathbf{X}^{1/3R} \begin{bmatrix} \sqrt{\Lambda_1^{1/3R}} & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \sqrt{\Lambda_6^{1/3R}} \end{bmatrix} V_N^{1/3R}$$
(3.14)

where:

$$\begin{split} D^{1/3R} &= \text{A diagonal matrix whose entries, } \Lambda_1^{1/3R} \cdots \Lambda_6^{1/3R} \text{ , are the eigenvalues of } \mathcal{Q}_{M_{kk-3}}^{1/3R} \text{ ;} \\ \mathbf{X}^{1/3R} &= \text{A matrix whose columns are the corresponding eigenvectors such} \\ \text{that: } \mathcal{Q}_{M_{kk-3}}^{1/3R} \mathbf{X}^{1/3R} = \mathbf{X}^{1/3R} D^{1/3R} \text{ ;} \end{split}$$

 $V_N^{1/3R}$  = 6x1 random vector whose elements are random variables distributed ~ N[0,1].

For CH<sup>2</sup> patterns, the state vectors of "predicted" particles,  $[x_{k|k-3}^{i_{1/3R}}, M_{k|k-3}^{j_{1/3R}}]$  are obtained by setting the CH<sup>2</sup> component of the 1/3-R state vector to the value indicated in the pattern index,  $M_{k|k-3}$ , passing it through the CH<sup>2</sup> state transition matrix and then adding a random CH<sup>2</sup> process noise vector:

$$\begin{bmatrix} x_{k_L} \\ x_{k_H} \\ x_{k_{H^2}} \end{bmatrix} = \begin{bmatrix} I & 6I & 9\sqrt{2}I \\ 0 & I & 3\sqrt{2}I \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_{k-3_L} \\ x_{k-3_H} \\ f_m \end{bmatrix} + w_{M_{kk-3}}^{1/3R^{CH^2}}$$
(3.15)

where:  $f_m = x_{k-3_{H^2}}$  = the maneuver pattern for pattern index  $M_{k/k-3}$ .

For example, if the CH<sup>2</sup> pattern was:  $f_m = x_{k_{H^2}} = 10\sqrt{2}$  , then (3.15) would become:

$$\begin{bmatrix} x_{k_L} \\ x_{k_H} \\ x_{k_{H^2}} \end{bmatrix} = \begin{bmatrix} I & 6I & 9\sqrt{2}I \\ 0 & I & 3\sqrt{2}I \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_{k-3_L} \\ x_{k-3_H} \\ 10\sqrt{2} \end{bmatrix} + w_{M_{k|k-3}}^{1/3R^{CH^2}}.$$
(3.16)

Correspondingly, for particles with CH patterns, the state vector is:

$$\begin{bmatrix} x_{k_L} \\ x_{k_H} \\ x_{k_{H^2}} \end{bmatrix} = \begin{bmatrix} I & 6I & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{k-3_L} \\ f_p \\ 0 \end{bmatrix} + w_{M_{k|k-3}}^{1/3R^{CH}} .$$
(3.17)

where:

 $f_p = x_{k_H}$  = the pointing pattern for pattern index  $M_{k|k-3}$ .

1/3-R Likelihood Function and Particle Weights : At this point, one can now compute a 1/3-R likelihood function,  $p(z_k^{1/3R} | x_{k|k-3}^{i_{1/3R}})$ , and the corresponding particle weights,  $w_k^{i_{1/3R}}$ . We construct a 1/3-R measurement vector from the measurements  $[z_{k-2}, z_{k-1}, z_k]$  via the method shown in (2.28). Since there are measurement false alarms, the measurement vector actually becomes a measurement matrix in which each column represents a 1/3-R measurement vector. If we assume that there is only one true target in the scenario, then the number of columns is equal to the number of measurement combinations,  $m_k$ , available from  $[z_{k-2}, z_{k-1}, z_k]$  and is defined as:

$$m_{k} = (1 + Nf_{k-2}) \cdot (1 + Nf_{k-1}) \cdot (1 + Nf_{k})$$
(3.19)

where:

 $Nf_k$  = the number of false alarms at time *k*.

Thus, the measurement matrix is defined as

$$z_k^{1/3R} = \left[ z_k^{1/3R_1}, \cdots, z_k^{1/3R_{m_k}} \right].$$
(3.19)

We assume that the false alarms obey the Poisson clutter model. Therefore, the probability of observing  $m_k$  false measurements at scan k is:

$$\mu_F(m_k) = e^{-\lambda V_k} \frac{(\lambda V_k)^{m_k}}{m_k!}$$
(3.20)

where:

 $\lambda_{k}$  = the false alarm rate per scan and  $V_{k}$  = measurement volume of the validation gate.

The 1/3-R likelihood function in (3.21) below is then obtained in an analogous manner to that of the parametric PDAF [7] and PDA particle filter [48]. We first note that the measurements are independent. Thus, by summing over the association hypotheses the aggregate 1/3-R likelihood function can be expressed as the sum of individual likelihood functions generated by  $m_{k+}$  1 hypotheses:

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}\right) = \sum_{n=0}^{m_{k}} p\left(z_{k}^{1/3R}, \theta_{k}^{n} \mid x_{k|k-3}^{i_{1/3R}}\right).$$
(3.21)

We now factor the LHS of (3.21) to obtain:

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}\right) = \sum_{n=0}^{m_{k}} p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}, \theta_{k}^{n}\right) p\left(\theta_{k}^{n} \mid x_{k|k-3}^{i_{1/3R}}\right)$$
(3.22)

where:

 $\theta_k^n$  = Feasible association hypothesis for measurement *n*;

 $p(\theta_k^n | x_{k|k-3}^{i_{l/3R}})$  = the probability of hypothesis  $\theta_k^n$ ;

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}, \theta_{k}^{n}\right)$$
 = the likelihood of hypothesis  $\theta_{k}^{n}$ .

The measurements are assumed to be independent. Thus, the overall likelihood function of a hypothesis becomes a product of the component likelihoods:

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}, \boldsymbol{\theta}_{k}^{n}\right) = \prod_{n=0}^{m_{k}} p\left(z_{k}^{1/3R_{n}} \mid x_{k|k-3}^{i_{1/3R}}, \boldsymbol{\theta}_{k}^{n}\right).$$
(3.23)

The component likelihoods,  $p(z_k^{1/3R_n} | x_{k|k-3}^{i_{1/3R}}, \theta_k^n)$ , can be expressed as:

$$p\left(z_{k}^{1/3R_{n}} \mid x_{k|k-3}^{i_{1/3R}}, \theta_{k}^{n}\right) = N\left(\left[z_{k}^{1/3R_{n}} - H x_{k|k-3}^{i_{1/3R}}\right], R_{k_{LH}}\right), \text{ if } z_{k}^{1/3R_{n}} \text{ is from a target;}$$
(3.24)

$$p(z_k^{1/3R_n} | x_{k|k-3}^{i_{1/3R}}, \theta_k^n) = \frac{1}{V_k}$$
, if  $z_k^{1/3R_n}$  is from a false alarm; (3.25)

where:

 $V_k$  = Volume of measurement space.

The likelihood of a given hypothesis can now be obtained using the previous results and applying them to the following cases:

- $\theta_0$ : None of the measurements are valid
- $\theta_n$ : Association hypotheses 1... $m_k$ , that each feature a single valid target.

This results in:

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}, \theta_{k}^{n}\right) = \frac{1}{V_{k}^{m_{k}-1}} N\left(\left[z_{k}^{1/3R_{n}} - H x_{k|k-3}^{i_{1/3R}}\right], R_{k_{LH}}\right), n = 1, 2, \cdots m_{k}$$
(3.26)

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}, \theta_{k}^{n}\right) = \frac{1}{V_{k}^{m_{k}}}, n = 0.$$
(3.27)

The next step is to compute,  $p(\theta_k^n | x_{k|k-3}^{i_{1/3R}})$ . Applying the PDA derivations in [7 and 48], we obtain:

$$p(\theta_k^n | x_{k|k-3}^{i_{1/3R}}) = \frac{1}{m_k} P_D P_G \left[ P_D P_G + (1 - P_D P_G) \frac{\mu_F(m_k)}{\mu_F(m_k-1)} \right]^{-1}, \ n = 1 \cdots m_k$$
(3.28)

$$p(\theta_k^n | x_{k|k-3}^{i_{1/3R}}) = (1 - P_D P_G) \frac{\mu_F(m_k)}{\mu_F(m_k - 1)} \left[ P_D P_G + (1 - P_D P_G) \frac{\mu_F(m_k)}{\mu_F(m_k - 1)} \right]^{-1}, \ n = 0$$
(3.29)

where:

 $P_D$  = probability of detection and  $P_G$  = probability that measurement falls in the measurement gate.

We now simplify the previous two equations by noting that:

$$\frac{\mu_F(m_k)}{\mu_F(m_k-1)} = \frac{\left[e^{-\lambda V_k} \frac{(\lambda V_k)^{m_k}}{m_k!}\right]}{\left[e^{-\lambda V_k} \frac{(\lambda V_k)^{m_k-1}}{(m_k-1)!}\right]} = \frac{\left[(m_k-1)!(\lambda V_k)^{m_k}\right]}{m_k!(\lambda V_k)^{m_k-1}} = \frac{\lambda V_k}{m_k}.$$
(3.30)

Applying (3.30) into (3.28) and (3.29), one obtains:

$$p(\theta_{k}^{n} | x_{k|k-3}^{i_{1/3R}}) = \frac{1}{m_{k}} P_{D} P_{G} \left[ P_{D} P_{G} + (1 - P_{D} P_{G}) \frac{\lambda V_{k}}{m_{k}} \right]^{-1} =$$

$$P_{D} P_{G} \left[ m_{k} P_{D} P_{G} + (1 - P_{D} P_{G}) \lambda V_{k} \right]^{-1}, n = 1 \cdots m_{k}$$

$$p(\theta_{k}^{n} | x_{k|k-3}^{i_{1/3R}}) = (1 - P_{D} P_{G}) \frac{\lambda V_{k}}{m_{k}} \left[ P_{D} P_{G} + (1 - P_{D} P_{G}) \frac{\lambda V_{k}}{m_{k}} \right]^{-1} =$$

$$(3.32)$$

$$(1 - P_D P_G) \frac{\lambda V_k}{m_k} \left[ m_k P_D P_G + (1 - P_D P_G) \lambda V_k \right]^{-1}, \ n = 0$$

The previous results are now substituted into (3.22) to obtain (3.33)

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}\right) = \sum_{n=0}^{m_{k}} p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}, \theta_{k}^{n}\right) p\left(\theta_{k}^{n} \mid x_{k|k-3}^{i_{1/3R}}\right) = \frac{1}{V_{k}^{m_{k}}} \left(1 - P_{D} P_{G}\right) \frac{\lambda V_{k}}{m_{k}} \left[m_{k} P_{D} P_{G} + \left(1 - P_{D} P_{G}\right) \lambda V_{k}\right]^{-1} + \frac{P_{D} P_{G}}{V_{k}^{m_{k}-1}} \left[m_{k} P_{D} P_{G} + \left(1 - P_{D} P_{G}\right) \lambda V_{k}\right]^{-1} \sum_{r=1}^{m_{k}} N\left(\left[z_{k}^{1/3R_{n}} - H x_{k|k-3}^{i_{1/3R}}\right], R_{k_{LH}}\right)$$
(3.33)

In the MRMMPF presented here, the SNR is assumed to be large and gating is not used. Consequently  $P_D$  is set to unity (i.e. target is always detected) and  $P_G = 1$ . The likelihood function then becomes:

$$p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i_{1/3R}}\right) = \frac{1}{m_{k} V_{k}^{m_{k}-1}} \sum_{n=1}^{m_{k}} N\left(\left[z_{k}^{1/3R_{n}} - H x_{k|k-3}^{i_{1/3R}}\right], R_{k_{LH}}\right).$$
(3.34)

The importance weights are then calculated using the same methodology as in the standard bootstrap PF. Since the particle weights are normalized, the constant term,  $\frac{1}{m_k V_k^{m_k-1}}$ , drops out

and we are left with (3.35) below.

$$w_{k}^{i_{1/3R}} = \frac{p\left(z_{k}^{1/3R} \mid x_{k/k-3}^{i_{1/3R}}\right)}{\sum_{nn=1}^{N_{p}} p\left(z_{k}^{1/3R} \mid x_{k/k-3}^{nn_{1/3R}}\right)} = \frac{\sum_{n=1}^{M_{k}} N\left(\left[z_{k}^{1/3R_{n}} - H x_{k|k-3}^{i_{1/3R}}\right], R_{k_{LH}}\right)}{\sum_{nn=1}^{N_{p}} \left(\sum_{n=1}^{M_{k}} N\left(\left[z_{k}^{1/3R_{n}} - H x_{k|k-3}^{nn_{1/3R}}\right], R_{k_{LH}}\right)\right)\right)}$$
(3.35)

The particles are then resampled via the importance sampling method to obtain  $x_{k|k}^{i_{1/3R}}$ . Following resampling, all particle importance weights are set to  $\frac{1}{N_P}$ . The 1/3-R target state in each particle is then transformed via a 1/3-R-to-1-R inverse DWT, to obtain the 1-R target state,  $x_{k|k}^{i}$ . If the particle in question has a CH model index, then  $T_{CV}^{-1}$  is applied while  $T_{CA}^{-1}$  is used for a CH<sup>2</sup> particle.

The model indices in the transformed particles are unchanged. Thus, a 1/3-R particle with an index that corresponds to a negative  $x_{k_{\mu^2}}$  value will be transformed into a 1-R particle with a negative acceleration while 1/3-R CH particles are transformed into constant velocity (acceleration = 0) 1-R particles. The full-rate output of the MRMMPF at time *k* is then obtained by taking the mean of the 1-R particle states as shown in (3.36).

$$\hat{x}_{k|k} = \frac{1}{N_P} \sum_{i=1}^{N_P} x_{k|k}^i$$
(3.36)

#### 3.1.5 Full-Rate MMPF

**1-R State Propagation:** The 1/3-R MMPF only generates target state outputs at every third sample point (*k*-3, *k*, *k*+3, ...). In order to obtain 1-R target states at the interim two sample points (*k*-2 and *k*-1, *k*+1 and *k*+2, ...), we use a modified version the MMPF described previously. To do this, at every the sample (*k*-3, *k*, *k*+3, ...) the state from each 1/3-R particle is fed through the 1/3R-1R inverse DWT described previously. This generates a 1-R particle set, [ $x_{k-3|k-3}^i$ ,  $M_{k-3|k-3}^i$ ]. This particle set is filtered via a 2-stage/3-model, constant acceleration (CA) MMPF. The prior PDF is obtained via the following kinematic model:

$$x_{k-2|k-3} = f_{M_{k-2|k-3}}(x_{k-3}) + w_{M_{k-2|k-3}} = \begin{bmatrix} I & TI & \frac{T^2}{2}I \\ 0 & I & TI \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x \\ x' \\ x'' = a_{M_{k-2|k-3}} \end{bmatrix} + w_{M_{k-2|k-3}}.$$
 (3.37)

The MMPF includes models for  $a_{M_{k-2k-3}} = -a_{\max}$ ,  $a_{M_{k-2k-3}} = 0$ ,  $a_{M_{k-2k-3}} = a_{\max}$ . The  $a_{M_{k-2k-3}} = 0$  model processes particles with model indices corresponding to the CH patterns while the  $a_{M_{k-2k-3}} = \pm a_{\max}$  models process particles with model indices corresponding to equivalent CH<sup>2</sup> patterns (i.e.  $x_{k_{H^2}} = \pm \frac{\sqrt{2}}{2} T^2 a_{\max}$ ). The 1-R process noise vector is derived by using a

similar technique to the one described for the 1/3-R case. First, the full-rate process noise covariance matrix,  $Q_{M_{k-2k-3}}$ , is derived for both the  $a_{M_{k-2k-3}} = \pm a_{\max}$  models and the  $a_{M_{k-2k-3}} = 0$  model (which is actually a constant velocity (CV) model).

The process noise matrix for the  $a_{M_{k-2k-3}} = \pm a_{\max}$  models is then expressed as:

$$Q_{M_{k-2|k-3}}^{CA} = \Gamma_{k-3}^{CA} \sigma_{w^{CA}}^2 \Gamma_{k-3}^{\prime CA}$$
(3.38)

where:

$$\Gamma_{k-3}^{CA} = \begin{bmatrix} \frac{T^2}{2} I \\ T \\ I \end{bmatrix}$$
 is the full-rate CA noise gain;

 $\sigma_{_{\rm wC4}}^2$  is the full-rate constant acceleration process noise variance.

The process noise matrix for the  $a_{M_{k-2k-3}} = 0$  model,  $Q_{M_{k-2k-3}}^{CV}$ , is developed in a similar fashion

except that the process noise gain terms becomes:  $\Gamma_{k-3}^{CV} = \begin{bmatrix} \frac{T^2}{2} I \\ T \\ 0 \end{bmatrix}$ . Theoretically, the process

noise variance,  $\sigma_{w^{CA}}^2$ , would be set to zero in a CV model. In a particle filter, however, additional process noise must be added in order to prevent the "degeneracy phenomenon". Note: A very small value (~10<sup>-8</sup>) is also added to the last element of the  $\Gamma_{k-3}^{CV}$  vector so that the process noise covariance matrix stays non-singular. Once the process noise matrices are computed, the process noise vector is computed the same way as in the 1/3 rate case via (3.39).

$$w_{M_{k-2|k-3}} = \mathbf{X} D \quad V_N = \mathbf{X} \begin{bmatrix} \sqrt{\Lambda_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sqrt{\Lambda_6} \end{bmatrix} V_N$$
(3.39)

**1-R Likelihood Function:** The first stage of the MMPF uses the measurement set  $z_{k-2}$ . As was the case with the 1/3-R measurements, this measurement set includes false alarms. Thus, the number of measurements is:

$$m_{k-2} = (1 + Nf_{k-2}) \tag{3.40}$$

where:  $Nf_{k-2}$  = the number of false alarms at *k*-2.

We now derive a full-rate PDAF-type likelihood function using exactly the same methodology as in the 1/3-R case and obtain (3.41-3.42).

$$p\left(z_{k-2} | x_{k-2|k-3}^{i}\right) = \frac{1}{m_{k-2} V_{k-2}^{m_{k}-1}} \sum_{n=1}^{m_{k-2}} N\left(z_{k-2}^{n} - H x_{k-2|k-3}^{i}\right), R_{k}$$
(3.41)

$$w_{k-2}^{i} = \frac{p(z_{k-2} \mid x_{k-2/k-3}^{i}))}{\sum_{n=1}^{N_{p}} p(z_{k-2} \mid x_{k-2/k-3}^{nn})} = \frac{\sum_{n=1}^{m_{k-2}} N([z_{k-2}^{n} - H x_{k-2/k-3}^{i}], R_{k-2}))}{\sum_{n=1}^{N_{p}} (\sum_{n=1}^{m_{k-2}} N([z_{k-2}^{n} - H x_{k-2/k-3}^{nn}], R_{k-2}))}$$
(3.42)

This produces a full-rate particle set  $[x_{k-2|k-2}^{i}, M_{k-2|k-2}^{i}]$ . The full-rate output at

 $\hat{x}_{k-2|k-2}$  is then obtained by computing the sample mean. Since we are already operating at the full-rate, an inverse DWT is not used. The second stage of the MMPF repeats the process with  $z_{k-1}$  and generates  $\hat{x}_{k-1|k-1}$ .

# 4 MULTIRATE MULTIPLE MODEL PARTICLE FILTER TRACK BEFORE DETECT (MRMMPF-TBD)

This section presents a full-rate multiple model particle filter for track before detect (MMPF-TBD) and a multirate multiple model particle filter for track-before-detect (MRMMPF-TBD). It extends the previously developed MMPF and MRMMPF so that they can track low SNR targets which perform small maneuvers. Current particle filter track before detect (PF-TBD) algorithms assume constant velocity (CV) motion and filter updates at a full-rate (i.e. at every measurement scan). Previous work in multirate processing, via a discrete wavelet transform (DWT), has shown that multirate tracking algorithms can provide comparable performance at a lower computational cost.

To date, these multirate approaches have not yet been applied to low signal-to-noise ratio (SNR) targets. Consequently, the goal of the MRMMPF-TBD is to combine the MMPF, TBD techniques, and multirate processing in order track low-SNR targets at a reduced particle cost.

#### 4.1 MRMMPF-TBD Algorithm Overview

The MRMMPF-TBD and MMPF-TBD both use a combined probabilistic data association (PDA) and maximum likelihood (ML) approach. The MRMMPF-TBD (top-level block diagram shown in Figure 4.1) consists of a 3-model full-rate MMPF run in parallel with third-rate, 3-model, MMPF. The full-rate MMPF uses a CV model and two constant acceleration (CA) models for positive or negative accelerations. The third-rate MMPF employs a constant high-pass (CH) model, which is analogous to the full-rate CV model.

A third-rate model is used instead of a half-rate because the third-rate model only requires one update per three scans (versus one update per two scans for half-rate), resulting in a lower particle count. Additionally, at least three scans are required to obtain CH<sup>2</sup> state and measurement vectors (which are analogous to acceleration components). Both the full-rate and

third-rate MMPFs use the bootstrap method and incorporate ML-PDA likelihood functions for data association and particle weighting. The basic operation of the MRMMPF-TBD is summarized as follows:

**1. Initialization:** Begin with a set of 1/3-R particles at *k*-3. Each particle consists of an ordered pair,  $[x_{k-3|k-3}^{i^{1/3R}}, M_{k-3|k-3}^{i^{1/3R}}]$ , that consists of a target state and a mode index.

 Mode Mixing: Perform mode mixing according to the Markov state transition probabilities.

3. Third-rate and Full-rate Separation: Compute the probability of maneuver by determining the ratio of CH<sup>2</sup> mode particles,  $N_{CH^2}$ , to the total number of particles  $N_P$ :

$$P_{man} = \frac{N_{CH^2}}{N_P}$$
,  $P_{nonman} = 1 - P_{man}$ . Divide the particle set into two portions such that

 $P_{man} N_P$  particles are assigned to the full-rate set and then convert the states of these particles from 1/3-R $\rightarrow$ 1-R via the inverse discrete wavelet transform (IDWT) matrix (Defined in section 3). The mode indices for each particle remain unchanged. Thus, CH particles are mapped to CV, and CH<sup>2</sup> particles are mapped to their appropriate CA model. The remaining fraction (i.e.  $P_{nonman} N_P$ ) of the particle set is left unchanged. The key point is that as the maneuver probability increases more particles are processed via the full-rate model in order to quickly respond to maneuvers.

**4. MMPF:** Process the third-rate CH particle set by the third-rate MMPF, using the 1/3-R measurement vector (Described in Section 3), and compute the posterior 1/3-R partial particle set,  $[\tilde{x}_{k|k}^{i^{1/3R}}, \tilde{M}_{k|k}^{i^{1/3R}}]$ . Process the full-rate particle CV/CA set by the three full-rate MMPFs, using the 1-R measurements at k-2, k-1, k, and compute the posterior 1-R partial particle set,  $[\tilde{x}_{k|k}^{i}, \tilde{M}_{k|k}^{i}]$ . Convert the 1-R particle set to 1/3-R via the DWT and then merge with the partial

posterior 1/3-R set, resulting in the complete posterior 1/3-R set,  $[x_{k|k}^{i^{1/3R}}, M_{k|k}^{i^{1/3R}}]$ .

5. State Estimation: Convert the 1/3-R particle set,  $[x_{k|k}^{i^{1/3R}}, M_{k|k}^{i^{1/3R}}]$ , to 1-R via the inverse DWT and then compute the conditional mean of the particle states to obtain the state estimate,  $\hat{x}_{k|k}$ . Real time outputs, if required, can be obtained at k-2 and k-1 from the 1-R MMPF particle sets. Otherwise, target states for k-2 and k-1 can be obtained by smoothing  $\hat{x}_{k|k}$ . Smoothed, non-real-time outputs will generally be more accurate since they incorporate the information from measurements at k-2, k-1, k and are based on a larger particle set.



Figure 4.1 MRMMPF-TBD Block Diagram

# 4.2 Full-Rate Target Models

The full-rate (1-R) target models used are the standard CV and CA models with the state vectors defined as shown in (4.1)-(4.3). In both the CV and CA cases, the control input,  $u_k$ , is modeled as a Gaussian noise process,  $w_k$ , that is zero mean and has variance  $\sigma_{w_k}^2$ .

$$X_{k+1} = F_k X_k + \Gamma_k u_k \tag{4.1}$$

For the CA Model: 
$$\begin{bmatrix} x \\ v_x \\ a_x \\ y \\ v_y \\ a_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ v_x \\ a_x \\ y \\ v_y \\ a_y \end{bmatrix}_k + \begin{bmatrix} \frac{T^2}{2} \\ T \\ \frac{T^2}{2} \\ T \\ 1 \end{bmatrix} w_k$$
(4.3)

The 1-R process noise covariance for the CV case is therefore defined as:

Correspondingly, the 1-R process noise covariance for the CA case is:

$$Q_{k} = \Gamma_{k} \sigma_{w_{k}}^{2} \Gamma_{k}' = \begin{bmatrix} \frac{T^{2}}{2} \\ \frac{T}{2} \\ \frac{1}{2} \\ \frac{T^{2}}{2} \\ \frac{T}{2} \\ 1 \end{bmatrix}' \sigma_{w_{k}}' \begin{bmatrix} \frac{T^{2}}{2} \\ \frac{T}{2} \\ \frac{T^{2}}{2} \\ \frac{T}{2} \\ 1 \end{bmatrix}' = \sigma_{w_{k}}^{2} \begin{bmatrix} \frac{T^{4}}{4} & \frac{T^{3}}{2} & \frac{T^{2}}{2} \\ \frac{T^{3}}{2} & T^{2} & T & 0 & 0 & 0 \\ \frac{T^{2}}{2} & T & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{T^{4}}{4} & \frac{T^{3}}{2} & \frac{T^{2}}{2} \\ 0 & 0 & 0 & \frac{T^{3}}{2} & T^{2} & T \\ 0 & 0 & 0 & \frac{T^{2}}{2} & T & 1 \end{bmatrix}$$
(4.5)

# 4.3 Third-Rate Target Models

The third rate models include a constant high-pass (CH) model for tracking during nonmaneuvering segments. The CH model is analogous to the CV model in the full-rate case while the CH<sup>2</sup> model (and its associated patterns) is analogous to the full-rate constant acceleration (CA) models. Note: The third-rate MMPF does not include constant high-high pass (CH<sup>2</sup>) model. Since the MRMMPF-TBD does, however, require CA $\leftarrow \rightarrow$ CH<sup>2</sup> conversions, a derivation of a 2pattern CH<sup>2</sup> model is included for completeness.

The vectors are then stacked to produce a 1/3-Rate state vector:

• Low-Pass  $X_L$  (analogous to position)

- High-Pass  $X_H$  (analogous to velocity)
- High-High Pass  $X_{H}^{2}$  (analogous to acceleration)

Where: 
$$X_{k+3}^{1/3R} = F_k^{1/3R} X_k^{1/3R} + \Gamma_k^{1/3R} u_k^{1/3R}$$
 (4.6)

The 1/3-R state transition matrices  $F_{CH}$  and  $F_{CH^2}$  for the equation above are obtained from the CH and CH<sup>2</sup> model definition that was derived in [45].

CH<sup>2</sup> Model: 
$$\begin{bmatrix} x_L \\ x_H \\ x_{H^2} \\ y_L \\ y_H \\ y_{H^2} \end{bmatrix}_{k+3} = F_{CH^2} X_k^{1/3R} = \begin{bmatrix} 1 & 6 & 9\sqrt{2} & 0 & 0 & 0 \\ 0 & 1 & 3\sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 6 & 9\sqrt{2} \\ 0 & 0 & 0 & 0 & 1 & 3\sqrt{2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L \\ x_H \\ x_{H^2} \\ y_L \\ y_H \\ y_{H^2} \end{bmatrix}_k$$
(4.8)

In the CH model, the control inputs are modeled as  $CH^2$  Gaussian noise disturbances with:  $x_{k+1_{H^2}} \sim N(0, Q_{k+1_{H^2}}), \quad x_{k+2_{H^2}} \sim N(0, Q_{k+2_{H^2}}), \quad x_{k+3_{H^2}} \sim N(0, Q_{k+3_{H^2}})$ . Correspondingly, in the CH<sup>2</sup> Model the control inputs are modeled as CH<sup>3</sup> Gaussian noise disturbances with:

 $x_{k+1_{H^3}} \sim N(0, Q_{k+1_{H^3}}), \quad x_{k+2_{H^3}} \sim N(0, Q_{k+2_{H^3}}), \quad x_{k+3_{H^3}} \sim N(0, Q_{k+3_{H^3}}) \,.$ 

Full-rate state vectors are converted into third-rate state vectors (i.e.  $1-R \rightarrow 1/3-R$ ) via an invertible linear transformation matrix. In the case of a constant velocity model, the CV-CH transform matrix is  $T_{CV}$  while the CA-CH<sup>2</sup> matrix is  $T_{CA}$ .

$$CV \to CH: x_k^{1/3R} = T_{CV} x_k = \begin{bmatrix} 2 & -2T & 0 & 0 & 0 & 0 \\ 0 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -2T & 0 \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_k$$
(4.9)

$$CA \to CH^{2}: x_{k}^{1/3R} = T_{CA} x_{k} = \begin{bmatrix} 2 & -2T & 1.5T^{2} & 0 & 0 & 0 \\ 0 & T & -T^{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2}T^{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -2T & 1.5T^{2} \\ 0 & 0 & 0 & 0 & T & -T^{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2}T^{2} \end{bmatrix} x_{k}$$
(4.10)

The reverse transformations (i.e. 1/3-R $\rightarrow$ 1-R) are accomplished by multiplying the 1/3-R state vectors via the inverse DWT matrices,  $T_{_{CV}}^{-1}$  and  $T_{_{CA}}^{-1}$ .

#### 4.4 Full-Rate Measurement Model

A typical radar sensor provides range, range rate, and angle information for a target. The sensor used in this simulation is a simplified radar that outputs a matrix of 2-dimensional x-y target position bins along with a target intensity reading for each bin. Thus, the 1-R measurement vector is defined as:
$$z_{k} = \begin{bmatrix} z_{x} \\ z_{y} \\ z_{I} \end{bmatrix}_{k} = \begin{bmatrix} H_{k} X_{k} \\ z_{I} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} X_{k}$$
(4.11)

Targets are modeled as point masses and target position smearing due to FFT windowing effects or target extent are not modeled. The target amplitude probability density function (PDF) for bin (x,y) is modeled as Rayleigh random variable (RV) via the method described in [7] and [32], where *SNR* is the defined as the minimum expected SNR.

Target + Noise PDF: 
$$p_{_{1}}(z_{I_{k}}) = \frac{z_{I_{k}}}{(1+SNR)} \exp\left(-\frac{z_{_{I_{k}}}^{2}}{2(1+SNR)}\right)$$
 (4.12)

Noise-only PDF: 
$$p_0(z_{I_k}) = z_{I_k} \exp\left(-\frac{z_{I_k}^2}{2}\right)$$
 (4.13)

#### 4.5 Third-Rate Measurement Model

In order to convert the full-rate 1-R position measurements into 1/3-R low-pass and highpass components, we apply a 1-R-to-1/3-R measurement transformation [45]:

$$\begin{bmatrix} z_{k_L} \\ z_{k_H} \end{bmatrix} = \begin{bmatrix} 0.5z_{k-2} + z_{k-1} + 0.5z_k \\ -0.5z_{k-2} + 0.5z_k \end{bmatrix}.$$
(4.14)

The target intensity is assumed to change slowly. It is thus modeled as a low-pass process with no high-pass components:

$$z_{I_k}^{1/3R} = 0.5 \, z_{I_{k-2}} + z_{I_{k-1}} + 0.5 \, z_{I_k} \quad . \tag{4.15}$$

The complete 1/3-R measurement vector is then defined via (4.16).

$$\begin{bmatrix} z_{x_L} \\ z_{y_L} \\ z_{x_H} \\ z_{y_H} \\ z_{I_L} \end{bmatrix}_k = \begin{bmatrix} 0.5 z_x \\ 0.5 z_y \\ -0.5 z_x \\ -0.5 z_y \\ 0.5 z_I \end{bmatrix}_{k-2} + \begin{bmatrix} z_x \\ z_y \\ 0 \\ 0 \\ z_I \end{bmatrix}_{k-1} + \begin{bmatrix} 0.5 z_x \\ 0.5 z_y \\ 0.5 z_y \\ 0.5 z_I \end{bmatrix}_k$$

(4.16)

$$= \begin{bmatrix} H_k^{1/3R} X_k^{1/3R} \\ z_{I_{L_k}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ & & z_{I_{L_k}} \end{bmatrix} X_k^{1/3R}$$

## 4.6 Particle Weight Computation (Full-Rate)

The MRMMPF-TBD algorithm uses the bootstrap method in which the un-normalized particle weights are proportional to the value of the measurement likelihood function:

$$w_k^i \propto p(z_k \mid x_k^i). \tag{4.17}$$

The ML-PDA likelihood function for the full-rate particle filter is presented below and is obtained by applying the methods described in non-parametric PDAF [7], PDA particle filter [48], and the non-parametric IMMPDAFAI [7]. It is first assumed that measurements are statistically independent and that position and target amplitude measurements within the validation gate are also independent. The measurement likelihood function can be expressed as the sum of joint likelihood functions generated by  $m_k + 1$  hypotheses:

$$p(z_k \mid x_k^i) = \sum_{n=0}^{m_k} p(z_k, \theta_k^n \mid x_k^i)$$
(4.18)

We now factor the RHS of (4.18) to obtain:

$$p(z_k \mid x_k^i) = \sum_{n=0}^{m_k} p(z_k \mid x_k^i, \theta_k^n) p(\theta_k^n \mid x_k^i)$$
(4.19)

where:

 $\theta_k^n$  = Feasible association hypothesis for measurement *n*;

 $p(\theta_k^n | x_k^i) = \text{the probability of hypothesis } \theta_k^n;$  $p(z_k | x_k^i, \theta_k^n) = \text{the likelihood of hypothesis } \theta_k^n.$ 

Since the measurements are assumed to be independent the overall hypothesis likelihood becomes a product of the component likelihoods:

$$p\left(z_{k} \mid x_{k}^{i}, \boldsymbol{\theta}_{k}^{n}\right) = \prod_{n=0}^{m_{k}} p\left(z_{k}^{n} \mid x_{k}^{i}, \boldsymbol{\theta}_{k}^{n}\right).$$

$$(4.20)$$

The likelihood of a given hypothesis can now be obtained using the previous results and applying them to the following cases:

- $\theta_0$ : None of the measurements are valid;
- $\theta_n$ : Association hypotheses 1...n, each feature a single valid target.

In order to compute  $p(z_k | x_k^i, \theta_k^n)$ , we also assume that position and target amplitude measurements within the validation gate are independent. The likelihood of hypothesis  $\theta_k^n$  can then be decomposed as a product of individual position and amplitude likelihoods:

$$p(z_{k} | x_{k}^{i}, \theta_{k}^{n}) = p(z_{I_{k}}, z_{P_{k}} | x_{k}^{i}, \theta_{k}^{n})$$
  
=  $p(z_{I_{k}} | x_{k}^{i}, \theta_{k}^{n}) \cdot p(z_{P_{k}} | x_{k}^{i}, \theta_{k}^{n})$ , (4.21)

Where:  $Z_{P_k}$  is the position component of the measurement and  $Z_{I_k}$  is the target intensity.

The amplitude likelihood function,  $p(z_{I_k} | x_k^i, \theta_k^n)$ , if  $z_k$  is from a target is:

$$p(z_{I_k} \mid x_k^i, \theta_k^n) = p_1(z_{I_k}^n) \cdot \prod_{j \neq n}^{m_k} p_0(z_{I_k}^j) = \frac{p_1(z_{I_k}^n)}{p_0(z_{I_k}^n)} \cdot \prod_{j=1}^{m_k} p_0(z_{I_k}^j).$$
(4.22)

Substituting the PDFs for  $p_0(z_{i_k}^n)$  and  $p_1(z_{i_k}^n)$  that were previously computed, we obtain:

$$p(z_{I_{k}} | x_{k}^{i}, \theta_{k}^{n}) = \frac{\frac{z_{I_{k}}^{n}}{(1 + SNR)} \exp\left(-\frac{(z_{I_{k}}^{n})^{2}}{2(1 + SNR)}\right)}{z_{I_{k}}^{n} \exp\left(-\frac{(z_{I_{k}}^{n})^{2}}{2}\right)} \cdot \prod_{j=1}^{m_{k}} p_{0}(z_{I_{k}}^{j})$$

$$= \frac{1}{(1 + SNR)} \exp\left(\frac{(z_{I_{k}}^{n})^{2} SNR}{2(1 + SNR)}\right) \cdot \prod_{j=1}^{m_{k}} p_{0}(z_{I_{k}}^{j})$$
(4.23)

The equation above is now expressed more compactly as:

$$p(z_{I_k} \mid x_k^i, \theta_k^n) = L(z_{I_k}^n) \cdot \prod_{j=1}^{m_k} p_0(z_{I_k}^j).$$
(4.24)

The amplitude likelihood function,  $p(z_{I_k} | x_k^i, \theta_k^n)$ , if  $z_k$  is not from a target is:

$$p(z_{I_k} \mid x_k^i, \theta_k^n) = \prod_{j=1}^{m_k} p_0(z_{I_k}^j).$$
(4.25)

The next step is to compute the position likelihood. If  $z_{P_k}^n$  is from a target, then the position likelihood is defined as a Gaussian:

$$p\left(z_{P_{k}}^{n} \mid x_{k}^{i}, \theta_{k}^{n}\right) = \frac{P_{G}^{-1}}{V_{k}^{m_{k}-1}} N\left(\left[z_{P_{k}}^{n} - \operatorname{H} x_{k|k-1}^{i}\right], R_{k}\right), n = 1, 2, \cdots m_{k} .$$
(4.26)

If  $z_{P_k}^n$  is not from a target:

$$p\left(z_{P_{k}}^{n} \mid x_{k}^{i}, \theta_{k}^{n}\right) = \frac{1}{V_{k}^{m_{k}}}, n = 0$$
(4.27)

where:

 $V_k$  = Volume of measurement space;

 $P_{G}$  = Probability that the correct measurement is inside the gate volume;

 $R_k$  = Position measurement covariance matrix.

The covariance matrix  $R_k$  is obtained by applying the standard radar measurement accuracy formula [4]:

$$R_{k} = \begin{bmatrix} r_{k}^{2} & 0\\ 0 & r_{k}^{2} \end{bmatrix}, where : r_{k} = \frac{sensor \ resolution}{2\sqrt{SNR}}$$
(4.28)

Since the actual SNR of the target will be unknown, the SNR value in the equation above is defined as the minimum SNR at which the tracker is designed to operate. The hypothesis probability,  $p(\theta_k^n | x_k^i)$ , is now calculated by using the non-parametric (i.e. diffuse prior) PDAF model shown in (4.29).

$$p\left(\theta_{k}^{n} \mid x_{k}^{i}\right) = \frac{P_{D}P_{G}}{m_{k}}, \quad i = 1, \cdots, m_{k}$$

$$p\left(\theta_{k}^{n} \mid x_{k}^{i}\right) = 1 - P_{D}P_{G}, \quad i = 0$$
(4.29)

The value of  $P_D$  is computed by integrating the target+noise PDF from the detection threshold  $\tau$  to infinity:

$$P_{D} = \int_{\tau}^{\infty} p_{1}(z_{I_{k}}) dz_{I_{k}} \quad .$$
(4.30)

Combining (4.26-4.29) and then substituting this into (4.18) we obtain the complete full-rate measurement likelihood function in (4.31).

$$p(z_{k} | x_{k}^{i}) = \prod_{j=1}^{m_{k}} p_{0}(z_{i_{k}}^{j}) \cdot \left\{ \frac{1 - P_{D}P_{G}}{V_{k}^{m_{k}}} + \frac{P_{D}}{m_{k}V_{k}^{m_{k}-1}} \sum_{n=1}^{m_{k}} \left\{ N\left[ \left[ z_{P_{k}}^{n} - \mathbf{H}_{k} x_{k|k-1}^{i} \right], R_{k} \right] \cdot L\left( z_{i_{k}}^{n} \right] \right\} \right\} = \left[ V_{k}^{-m_{k}} \cdot \prod_{j=1}^{m_{k}} p_{0}\left( z_{i_{k}}^{j} \right) \right] \cdot \left\{ \left( 1 - P_{D}P_{G} \right) + \left( \frac{P_{D}V_{k}}{m_{k}} \right) \sum_{n=1}^{m_{k}} \left\{ N\left[ \left[ z_{P_{k}}^{n} - \mathbf{H}_{k} x_{k|k-1}^{i} \right], R_{k} \right] \cdot L\left( z_{i_{k}}^{n} \right) \right\} \right\}$$
(4.31)

The normalized particle weights for the full-rate particle filter are shown in (4.32).

$$w_{k}^{i} = \frac{p(z_{k} \mid x_{k|k-1}^{i})}{\sum_{j=1}^{N_{p}} p(z_{k} \mid x_{k|k-1}^{j})}$$

$$= \frac{\left\{ \left(1 - P_D P_G\right) + \left(\frac{P_D V_k}{m_k}\right) \sum_{n=1}^{m_k} \left\{ N\left(\left[z_{P_k}^n - H_k x_{k|k-1}^i\right], R_k\right) \cdot L\left(z_{I_k}^n\right) \right\} \right\}}{\sum_{j=1}^{N_P} \left\{ \left(1 - P_D P_G\right) + \left(\frac{P_D V_k}{m_k}\right) \sum_{n=1}^{m_k} \left\{ N\left(\left[z_{P_k}^n - H_k x_{k|k-1}^j\right], R_k\right) \cdot L\left(z_{I_k}^n\right) \right\} \right\}}$$

$$(4.32)$$

The term  $\left[ V_k^{-m_k} \cdot \prod_{n=1}^{m_k} p_0(z_{r_k}^n) \right]$  is common to both numerator and denominator and thus drops out.

## 4.7 Particle Weight Computation (1/3-Rate)

The 1/3-R position likelihood is computed in an analogous manner to the full-rate case. The key difference is that 1/3-R measurements and a 1/3-R measurement covariance matrix,  $R_{k_{III}}$ , are used where:

$$R_{k_{LH}} = \begin{bmatrix} 1.5R_k & 0_{2x2} \\ 0_{2x2} & 0.5R_k \end{bmatrix}.$$
 (4.33)

In order to compute the 1/3-R amplitude likelihood, the 1/3-R noise and target+noise PDFs must first be computed. The 1/3-R amplitude measurement is sum of three 1-R Rayleigh random

variables. Therefore, 1/3-R PDFs can theoretically be obtained via a transformation of variables and three-way convolution of the 1-R PDF:

$$p_{1}^{1/3R}\left(z_{I_{k}}^{1/3R}\right) = \frac{p_{1}\left(\frac{z_{I_{k-2}}}{0.5}\right)}{|0.5|} * p_{1}\left(z_{I_{k-1}}\right) * \frac{p_{1}\left(\frac{z_{I_{k}}}{0.5}\right)}{|0.5|}.$$
(4.34).

In practice, however, this PDF derivation has no closed form solution. Instead, the 1/3-R PDFs were approximated via moment-matched Gaussian PDFs. In order to generate this moment-matched approximation, we note that the Rayleigh PDF has the following form:

$$p(z)_{ray} = \frac{z}{\sigma_{\omega}^2} \exp\left(-\frac{z^2}{2\sigma_{\omega}^2}\right)$$
(4.35)

with the following first and second moments:

$$Mean: \mu_{ray} = \sqrt{\frac{\pi}{2}} \sigma_{\omega} = 1.253314 \sigma_{\omega} \quad \text{and } Variance: \sigma_{ray}^2 = 2\sigma_{\omega}^2 - \mu_{ray}^2 \quad . \tag{4.36}$$

The 1/3-R PDF is then approximated by a Gaussian  $N(z_{I_L} - \mu_{ray}^{1/3R}, \sigma_{ray}^{2^{1/3R}})$  with:

$$\mu_{ray}^{1/3R} = 0.5 \,\mu_{ray} + \mu_{ray} + 0.5 \,\mu_{ray} = 2 \,\mu_{ray} \tag{4.37}$$

$$\sigma_{ray}^{2^{1/3R}} = 0.25 \sigma_{ray}^{2} + \sigma_{ray}^{2} + 0.25 \sigma_{ray}^{2} = 1.5 \sigma_{ray}^{2}.$$
(4.38)

For the noise-only case, the PDF is normalized so that  $\sigma_{\omega}^2 = 1$  while for the target+noise case,  $\sigma_{\omega}^2 = 1 + SNR$ .

The 1/3-R noise-only PDF is therefore approximated as  $N(z_{I_L} - \mu_0^{1/3R}, \sigma_0^{2^{1/3R}})$  while the 1/3-R target+noise PDF, is approximated by a Gaussian  $N(z_{I_L} - \mu_1^{1/3R}, \sigma_1^{2^{1/3R}})$ . The values of

 $\left[\mu_0^{1/3R}, \sigma_0^{2^{1/3R}}\right]$  are obtained from (4.36-4.38) by setting  $\sigma_\omega^2 = 1$  while  $\left[\mu_1^{1/3R}, \sigma_1^{2^{1/3R}}\right]$  are obtained by setting  $\sigma_\omega^2 = 1 + SNR$ .

The 1/3-R amplitude likelihood function, conditioned on hypothesis  $\theta_k^n$  then becomes:

$$p\left(z_{l_{k}}^{1/3R} \mid x_{k}^{i^{1/3R}}, \theta_{k}^{n}\right) = \frac{p_{1}^{1/3R}\left(z_{l_{k}}^{n^{1/3R}}\right)}{p_{0}^{1/3R}\left(z_{l_{k}}^{n^{1/3R}}\right)} \cdot \prod_{j=1}^{m_{k}^{1/3R}} p_{0}^{1/3R}\left(z_{l_{k}}^{j^{1/3R}}\right)$$

$$= \frac{N\left(z_{l_{k}}^{n^{1/3R}} - \mu_{1}^{1/3R}, \sigma_{1}^{2^{1/3R}}\right)}{N\left(z_{l_{k}}^{n^{1/3R}} - \mu_{0}^{1/3R}, \sigma_{0}^{2^{1/3R}}\right)} \cdot \prod_{j=1}^{m_{k}^{1/3R}} p_{0}^{1/3R}\left(z_{l_{k}}^{j^{1/3R}}\right) = L\left(z_{l_{k}}^{n^{1/3R}}\right) \cdot \prod_{j=1}^{m_{k}^{1/3R}} p_{0}^{1/3R}\left(z_{l_{k}}^{j^{1/3R}}\right).$$

$$(4.39)$$

The complete 1/3-R likelihood function (i.e. position-amplitude) and the 1/3-R particle weights can now be computed in the same manner as the 1-R case via (4.40).

$$w_{k}^{i_{l/3R}} = \frac{p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{i^{1/3R}}\right)}{\sum_{j=1}^{N_{p}} p\left(z_{k}^{1/3R} \mid x_{k|k-3}^{j^{1/3R}}\right)}$$

$$= \frac{\left\{\left(1 - P_{D}P_{G}\right) + \left(\frac{P_{D}V_{k}^{1/3R}}{m_{k}^{1/3R}}\right)^{m_{k}^{1/3R}} \left\{N\left(\left[z_{P_{k}}^{n^{1/3R}} - \mathbf{H}_{k}^{1/3R} x_{k|k-3}^{i^{1/3R}}\right], R_{k_{LH}}\right) \cdot L\left(z_{I_{k}}^{n^{1/3R}}\right)\right\}\right\}}{\sum_{j=1}^{N_{p}} \left\{\left(1 - P_{D}P_{G}\right) + \left(\frac{P_{D}V_{k}^{1/3R}}{m_{k}^{1/3R}}\right)^{m_{k}^{1/3R}} \sum_{n=1}^{N_{p}} \left\{N\left(\left[z_{P_{k}}^{n^{1/3R}} - \mathbf{H}_{k}^{1/3R} x_{k|k-3}^{j^{1/3R}}\right], R_{k_{LH}}\right) \cdot L\left(z_{I_{k}}^{n^{1/3R}}\right)\right\}\right\}}$$

$$(4.40)$$

#### 4.8 Measurement Gating

Dim target tracking generates numerous false alarms. Measurement gating is therefore required. The algorithms presented here use a heuristic three-level system of gating. The first level of gating, referred to as coarse gating, places a square gate around the predicted 1-R measurement point location:

$$z_{pred} = \overline{H}_k \ \hat{x}_{k|k-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \hat{x}_{k|k-1}; \text{ where } \hat{x}_{k|k} = \frac{1}{N_p} \sum_{i=1}^{N_p} x_{k|k-1}^i, \text{ with } N_p \text{ being the set of the set of$$

number of particles. (Note: The measurement matrix  $\overline{H}_k$  shown here uses only position information and not intensity.) The objective of the coarse gating is to remove unlikely measurements without incurring much computational cost. The length of a gate side was set at heuristically at 1X the sensor resolution level.

Measurements that passed the coarse gate were then gated via a fine gate in a fashion similar to that outlined by Vermak et al. [59] via (4.41-4.42). Fine gating is used with both the 1-R and 1/3-R measurements. This gating process is illustrated for the 1-R case in Figure 4.2. Note: The 1/3-R fine gates are 4-dimensional and cannot be easily depicted.

$$\gamma^{2} = \left[ z - \overline{H}_{k} \ \hat{x}_{k|k-1} \right]' \left( S_{k} \ \right)^{-1} \left[ z - \overline{H}_{k} \ \hat{x}_{k|k-1} \right] \leq \chi^{2} \ threshold \tag{4.41}$$

where:

$$S_{k} \equiv \overline{H}_{k} P_{k/k-1} \overline{H}'_{k} \quad where : P_{k/k-1} \equiv \frac{1}{N_{P} - 1} \sum_{i=1}^{N_{P}} \left[ x_{k/k-1}^{i} - \hat{x}_{k|k-1} \right] \left[ x_{k/k-1}^{i} - \hat{x}_{k|k-1} \right]' \quad (4.42)$$

The third level of gating is based on target amplitude and is only applied to 1/3-R measurements that pass 1/3-R fine gate. The measurements are sorted by target amplitude (highest-to-lowest) and only the  $N_{\rm max}$  highest ones are selected. **Note:** The value of  $N_{\rm max}$  is heuristically determined from simulation. A low value of  $N_{\rm max}$  reduces run time because fewer false measurement sets are generated. Unfortunately, if this value is set too low, the real measurement may sometimes be inadvertently eliminated, resulting in error. Conversely, if  $N_{\rm max}$  is too large then runtime will be excessive. During simulations,  $N_{\rm max} = 8$  produced good results and fast runtimes.

We construct a 1/3-R measurement vector from the measurements in three consecutive scans:  $[z_{k-2}, z_{k-1}, z_k]$ . The 1-R measurements at each time increment pass through the 1-R coarse gate. Since there are false alarms, the measurement vector actually becomes a measurement matrix in which each column represents a 1/3-R measurement vector. If we assume that there is only one true target in the scenario, then the number of columns is equal to the number of measurement combinations,  $m_k^{1/3R}$ . This quantity is obtained from  $[z_{k-2}, z_{k-1}, z_k]$  and is defined as:

$$m_{k}^{1/3R} = m_{k-2} m_{k-1} m_{k} = (1 + Nf_{k-2}) \cdot (1 + Nf_{k-1}) \cdot (1 + Nf_{k});$$
(4.43)

where  $Nf_k$  = the number of false alarms in the 1-R coarse gate at time *k*.

The 1/3-R measurement matrix is defined as  $z_k^{1/3R} = \left[z_k^{1^{1/3R}}, \dots, z_k^{m_k^{1/3R}}\right]$ . The 1/3-R measurement matrix then passes through a 1/3-R fine gate. Next, amplitude gating is applied to the surviving 1/3-R measurements. Amplitude gating becomes especially important for 1/3-R measurements because of the large number of spurious 1/3-R measurement combinations.



Figure 4.2 1-R Coarse and Fine Gating Example

# 5 EXTENDED SPATIAL DOMAIN MULTI-RESOLUTION PARTICLE FILTERING (E-SD-MRES-PF)

The Hong and Wicker SD-MRES-PF [71] was a proof-of-concept model that only operated over one update cycle. In its current form it is not suitable as a multiple update particle filtering algorithm because of the large number of samples required to generate the histogram PDF that the SD-MRES-PF employs. This section presents an extended SD-MRES-PF(E-SD-MRES-PF) that tracks the evolution of a non-linear/non-Gaussian state over multiple time increments. A detailed derivation of the SD-MRES-PF will not be included in this section since it already presented in [71]. Instead, this section will focus on the modifications required to implement the E-SD-MRES-PF.

The algorithm for the E-SD-MRES-PF is described below.

1. Generate initial sampled PDF of  $N_{P0}$  particles:  $p(x_0) = \sum_{i=1}^{N_{P0}} \omega_0^i \delta(x - x_o^i)$ . This

PDF will typically be non-Gaussian.

2. Propagate the particle set through the non-linear process equation, f(x), and

generate a the propagated sampled PDF:  $p(x_k | z_{k-1}) = \sum_{i=1}^{N_p} \omega_{k-1}^i \delta(x_k - x_{k|k-1}^i).$ 

- 3. Compute a likelihood function for each particle:  $p(z_k | x_{k/k-1}^i)$ .
- 4. Calculate the new normalized weight for each particle,

$$\omega_{k}^{i} = \frac{p(z_{k} | x_{k/k-1}^{i})\omega_{k-1}^{i}}{\sum_{i=1}^{N_{p}} p(z_{k} | x_{k/k-1}^{i})\omega_{k-1}^{i}} .$$

5. Sort the particles according to increasing  $x_{k/k}^i$  values in order to generate a

sampled PDF: 
$$p(x_k | z_k) = \sum_{i=1}^{N_{P0}} \omega_k^i \, \delta(x_k - x_k^i).$$

6. Transform the posterior sampled PDF via the "Explicit Method". To accomplish this, the weights of the sampled PDF are divided into eight-sample blocks. The

sum of the weights of each block,  $w_{B0} = \sum_{j=1}^{8} \omega^{j}$ , is then computed and saved (it

will be used later to re-normalize the weight of the block). Each block is then fed into a 3-level DWT Haar-wavelet filter bank that generates the Level 1-3 filter coefficients in the manner shown in Figure 5.1 below.



Figure 5.1 Multiresolutional Decomposition via DWT Filter Bank

7. Mechanize the filter bank via the linear transformation depicted in (5.1).

$$\boldsymbol{\omega}_{lh} = \begin{bmatrix} \boldsymbol{\omega}_{lh}^{1} \\ \boldsymbol{\omega}_{lh}^{2} \\ \boldsymbol{\omega}_{lh}^{3} \\ \boldsymbol{\omega}_{lh}^{3} \\ \boldsymbol{\omega}_{lh}^{6} \\ \boldsymbol{\omega}_{lh}^{6} \\ \boldsymbol{\omega}_{lh}^{6} \\ \boldsymbol{\omega}_{lh}^{6} \\ \boldsymbol{\omega}_{lh}^{8} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_{l3}^{1} \\ \boldsymbol{\omega}_{h3}^{1} \\ \boldsymbol{\omega}_{h2}^{2} \\ \boldsymbol{\omega}_{h2}^{2} \\ \boldsymbol{\omega}_{h1}^{1} \\ \boldsymbol{\omega}_{h1}^{2} \\ \boldsymbol{\omega}_{h1}^{3} \\ \boldsymbol{\omega}_{h1}^{3} \\ \boldsymbol{\omega}_{h1}^{8} \end{bmatrix} = T \begin{bmatrix} \boldsymbol{\omega}_{l}^{1} \\ \boldsymbol{\omega}_{l}^{2} \\ \boldsymbol{\omega}_{h2}^{3} \\ \boldsymbol{\omega}_{h1}^{6} \\ \boldsymbol{\omega}_{h1}^{5} \\ \boldsymbol{\omega}_{h1}^{6} \\ \boldsymbol{\omega}_{h1}^{7} \\ \boldsymbol{\omega}_{h1}^{8} \end{bmatrix}$$

(5.1)

where:

$$T = \begin{bmatrix} h^3 & h^3 \\ -h^3 & -h^3 & -h^3 & -h^3 & h^3 & h^3 & h^3 & h^3 \\ -h^2 & -h^2 & h^2 & h^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -h^2 & -h^2 & h^2 & h^2 \\ -h & h & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -h & h & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -h & h & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -h & h \end{bmatrix}, and h = \frac{\sqrt{2}}{2}$$

- 8. Compare the elements of  $\omega_{lh}$  against the predetermined threshold,  $t_s$ , and set to zero any element that is below this threshold. The result of this thresholding process is a new transformed block  $\tilde{\omega}_{lh}$ .
- 9. Apply inverse DWT,  $\tilde{\omega} = T^{-1} \tilde{\omega}_{lh} = T' \tilde{\omega}_{lh}$  in order to obtain a new block of uniresolution weights. Note: Some of the elements of the new block will have repeated elements. Larger thresholds will result in greater data compression and more repeated elements.
- Examine the zero elements of the new uni-resolution block in order to determine which elements are repeated and remove those so that only distinctive elements will be propagated.
- 11. Compute the sum of weights of the new block,  $w_{B1}$ . Multiply the individual

weights by the ratio of  $\frac{w_{B0}}{w_{B1}}$ . This ensures that the PDF segment represented by the new block will have the same total weight as the original block in Step-5. Once all of the eight-sample blocks are processed, we will have a reduced particle set with  $N_{P1}$  particles. An example of this thresholding and reconstruction process is depicted in Figure 5.2. 12. Resample (with replacement) the new reduced particle set and set the new

particle weight such that:.  $w_k^i = \frac{1}{N_{P1}}$ .

13. Generate the state estimate: 
$$E[x_k] = \frac{1}{N_{P1}} \sum_{i=1}^{N_{P1}} \delta(x_k - x_k^i)$$
.

14. Draw  $N_{P2} = (1+r)N_{P1}$  particles from the resampled particle set and reset the

particle weights to: 
$$w_k^i = \frac{1}{N_{P2}}$$
. The quantity *r* is a heuristically determined

value that prevents the reduced particle set from going to zero as the number of iterations increases. (A range of 0.1 ≤ r ≤ 0.3 produced good simulation results).
15. Go to Step-2 and repeat Steps 2-15.



Figure 5.2 Thresholding Example via the Explicit Method

# 6 GAUSSIAN FINITE MIXTURE MODEL PARTICLE FILTERS (GMMPF)

Two key challenges of particle filters are maintaining multimodality and reducing computational costs. Although particle filtering techniques outperform Kalman-based methods, their computational costs are between 2-3 orders of magnitude greater than Kalman filter-based estimators. This fact is illustrated in Table 4.1 below.

Filter Type

**Table 6-1** Filter Run Times (Rounded to Nearest Minute)

	Filter Type				
	IMM-PDAF	IMPDA	MMPF	MRMMPF	
Run Time (Minutes)	1	1	416	580	

The main culprits responsible for the large MMPF and MRMMPF runtimes are computing the likelihood functions for each particle and the resampling process. These large runtimes make particle filter-based trackers impractical for most "real-world" tracking scenarios. In addition, particle filters often cannot maintain multimodality over an extended period of time (i.e. weaker modes are suppressed). This section presents two Gaussian finite mixture model particle filter variants that address the multimodality and computational cost issues. During the design and testing of the MRMMPF, it became apparent that Matlab coding implementation (i.e. vectorizing code vs. loops) significantly impacted runtime. Since particle filter computational cost is a function of  $O(N_p)$ , particle count was used as the metric of computational cost.

#### 6.1 Gaussian Finite Mixture Models (FMM)

The GMMPF makes use of the Gaussian sum approximation lemma, which states that any PDF, f(x), can be approximated as closely as desired by a weighted sum of Gaussian PDFs,  $f_i(x)$ . In a Gaussian FMM, the multi-modal PDF, f(x), is approximated as a sum of kGaussians:

$$f(x) = \pi_1 f_1(x) + \pi_2 f_2(x) + \dots + \pi_k f_k(x)$$
(6.1)

where: 
$$\sum_{i=1}^{k} \pi_i = 1$$
.

The key benefit of the Gaussian FMM approach is that each individual component of f(x) can be completely described by only two parameters: a mean vector,  $\mu_i$ , and a covariance matrix,  $\Sigma_i$ . If  $f_i(x)$  represents the prior PDF and if the measurement likelihood function is Gaussian then each Gaussian component can be updated via a Kalman filter variant (analogous to that of a Gaussian sum filter bank). In the event that the measurement equations are non-linear but the likelihood is Gaussian, an EKF or UKF may be used instead. This GMMPF variant is designated as a Kalman GMMPF (K-GMMPF). If the measurement likelihood function is non-Gaussian then each component of f(x) is updated via a particle filter. These concepts are applied in the GMMPF and K-GMMPF algorithms described below.

The GMMPF/K-GMMPF used the K-means algorithm to divide the particle sets for each of the *r* models into *m* cluster components. Next, a mean and covariance,  $[\mu_{r,m}, \Sigma_{r,m}]$ , was computed for each cluster component. Component weights were computed by dividing the number of samples in each cluster component by the total number of particles,  $\pi_{r,m} = \frac{N_{r,m}}{N_P}$ .

Note: The expectation maximization (EM) algorithm [49] was initially used to generate the mixture parameters but proved unsatisfactory. It was slow and often produced ill conditioned covariance matrices. The K-means based parameter extraction algorithm was much faster than EM and also proved to be numerically stable.

#### 6.2 GMMPF and K-GMMPF Algorithms

The GMMPF/K-GMMPF algorithms both use three kinematic models: CV, CA-positive acceleration, and CA-negative acceleration. The GMMPF algorithm is summarized below and depicted in Figure 6.1.

- 1. Begin with a set of  $N_p$  "posterior" particles at time k-1:  $p(x_{k-1}|z_{k-1}, r_{k-1})$ . Model index indicator  $r_{k-1}$  is defined (for a 3-model filter) as:
  - a. r = 1 corresponds to a negative constant acceleration;
  - b. r = 2 corresponds to a zero acceleration (constant velocity);
  - C. r = 3 corresponds to a positive constant acceleration.
- 2. Perform model mixing according to Markov state transition matrix,  $P_T$ , with transition probabilities  $h^{ij}$ .
- 3. Run each particle through a process model whose kinematics are based on the particle model index and obtain a set of "prior" particles that represents:  $p(x_k|z_{k-1}, r_{k-1})$
- 4. Partition the particle set into three subsets such that each subset contains particles having the same model index, *r*.
- 5. Run each particle subset through an FMM parameter extraction algorithm and generate a mean vector,  $\mu_{r,m_{k-1}}$ , a covariance matrix  $\Sigma_{r,m_{k-1}}$ , and a mode probability  $\pi_{r,m_{k-1}}$ . Then model the prior particle density as a finite Gaussian mixture model: GMM<sub>1</sub>, GMM<sub>2</sub>, or GMM<sub>3</sub>. Each GMM is, in turn, composed of *m* Gaussians (*m* = 3 for the prototype algorithm). Note: the *k*-1 subscript is omitted in the RHS of 6.2-6-4 in order to reduce symbol clutter.
  - a. GMM<sub>1</sub> (i.e. the GMM for r = 1 particles):

$$p(x_k|z_{k-1}, r_{k-1}=1) = \pi_{1,1} N[\mu_{1,1}, \Sigma_{1,1}] + \pi_{1,2} N[\mu_{1,2}, \Sigma_{1,2}] + \dots + \pi_{1,m} N[\mu_{1,m}, \Sigma_{1,m}]$$
(6.2)

b.  $GMM_2$  (i.e. the GMM for r = 2 particles):

$$p(x_k|z_{k-1}, r_{k-1}=2) = \pi_{2,1} N[\mu_{2,1}, \Sigma_{2,1}] + \pi_{2,2} N[\mu_{2,2}, \Sigma_{2,2}] + \dots + \pi_{2,m} N[\mu_{2,m}, \Sigma_{2,m}]$$
(6.3)

C. GMM<sub>3</sub> (i.e. the GMM for r = 3 particles):

$$p(x_k|z_{k-1}, r_{k-1}=3) = \pi_{3,1}N[\mu_{3,1}, \Sigma_{3,1}] + \pi_{3,2}N[\mu_{3,2}, \Sigma_{3,2}] + \dots + \pi_{3,m}N[\mu_{3,m}, \Sigma_{3,m}]$$
(6.4)

- 6. Draw  $\pi_{r,m_{k-1}}N_P$  samples from each GMM mode,  $N[\mu_{r,m}, \Sigma_{r,m}]$ , and assign an appropriate model index, *r*, to each particle.
  - a. Particle model indices are assigned based on parent GMM model index.
  - b. Samples from each Gaussian mode processed via a separate particle filter,  $PF_{r,m}$ , where r = process model index and m = Gaussian mode index.
  - C. Compute particle weights,  $W_{r,m_k}^i$ , for each model *r* and mixture *m* via the SIR algorithm. Also compute the sum of the particle weights for each model/mode:

$$\widetilde{w}_{r,m_k} = \sum_{\forall i} w_{r,m_k}^i .$$
(6.5)

d. Compute the new mixture weights according to the method outlined by Vermaak [44]:

$$\pi_{r,m_{k}} = \frac{\pi_{r,m_{k-1}} \widetilde{W}_{r,m_{k}}}{\sum_{r=1}^{3} \sum_{m=1}^{M} \pi_{r,m_{k-1}} \widetilde{W}_{r,m_{k}}} .$$
(6.6)

7. Resample each of the particle filters. Draw  $N_{r,m_k}$  particles from each particle filter, where  $N_{r,m_k} = N_P \pi_{r,m_k}$ . Thus, the particles are drawn from each particle filter according to their new mixture weights and are combined into a single aggregated particle set. The aggregate particle set now approximates the posterior PDF,  $p(x_k|z_k, r_k)$ .

a. Filter output at time k is the mean the mean of the particle states:

$$\hat{x}_{k|k} = \frac{1}{N_P} \sum_{i=1}^{N_P} x_{k|k}^i$$
.

8. Go back to Step-2 and repeat process for next time increment.

The K-GMMPF algorithm is summarized below:

1-5. Same as GMMPF.

6. Process each GMM component mode,  $N[\mu_{r,m}, \Sigma_{r,m}]_{k|k-1}$ , by a Kalman filter (EKF or UKF) and compute the posterior PDF,  $N[\mu_{r,m}, \Sigma_{r,m}]_{k|k}$  for each component.

7. Same as GMMPF except that  $\tilde{w}_{r,m_k}$  is replaced by  $\Lambda_{r,m_k}$ , which is the measurement likelihood function for each Kalman filter.

8. Sample  $N_{r,m_k}$  particles from  $N[\mu_{r,m}, \Sigma_{r,m}]_{k|k}$  where  $N_{r,m_k} = N_P \pi_{r,m_k}$ . Thus, the particles are drawn from each particle filter according to their new mixture weights and are combined into a single aggregated particle set. The aggregate particle set now approximates the posterior PDF,  $p(x_k|z_k, r_k)$ .

a. Filter output at time k is the mean the mean of the particle states:

$$\hat{x}_{k|k} = \frac{1}{N_P} \sum_{i=1}^{N_P} x_{k|k}^i$$
.

9. Same as GMMPF.



Figure 6.1 Gaussian Finite Mixture Model Particle Filter (GMMPF)

# 7 SCENARIOS AND SIMULATION RESULTS

This section describes the simulation scenarios and provides the modeling results for following algorithms:

- MRMMPF vs. MMPF, IMPDA, and IMMPDAF;
- MRMMPF-TBD vs. MMPF-TBD;
- Extended Spatial Domain Spatial-Domain Multi-Resolution Particle Filtering (E-SD-MRES-PF);
- Gaussian Finite Mixture Model Particle Filters (GMMPF).

# 7.1 MRMMPF vs. MMPF, IMPDA, and IMMPDAF

# 7.1.1 Scenario Description

The simulation results below compare the performance of the following algorithms:

- 3-model IMMPDAF: 1-CV and 2-CA models;
- 4-Pattern IMPDA: Same patterns as MRMMPF;
- 3-Model MMPF (10,000 particles): 1-CV and 2-CA models (+accel./-accel.);
- Prototype 4-Pattern MRMMPF (10,000 particles): 2-CH patterns and 2-CH<sup>2</sup> patterns.

Each algorithm was tested against four different target acceleration scenarios:

a = +/-5; a = +/-15, a = +/-25, and a = +/-40 msec<sup>-2</sup>. The performance metrics were average x/y position root mean square (RMS) errors and average  $v_x/v_y$  velocity RMS errors. Of these metrics, the position RMS errors were the key ones. Each target trajectory lasted for 240 sample times, and consisted of five segments:

• Constant velocity segment-1;

- Constant acceleration segment-1 (positive acceleration);
- Constant velocity segment-2;
- Constant acceleration segment-2 (negative acceleration);
- Constant velocity segment-3.

The start-stop sample increments and acceleration levels of each track segment are summarized

in Table 7-1.

	Track	Scena	rio-1	Track Scenario-2		Track Scenario-3			Track Scenario-4			
Track Segment	Start Smpl	Stop Smpl	Acc	Start Smpl	Stop Smpl	Acc	Start Smpl	Stop Smpl	Acc	Start Smpl	Stop Smpl	Acc
CV-1	1	60	0	1	60	0	1	60	0	1	60	0
CA-1	61	90	5	61	90	15	61	73	25	61	69	40
CV-2	91	150	0	91	150	0	74	150	0	70	150	0
CA-2	151	180	-5	151	180	-15	151	163	-25	151	159	-40
CV-3	181	240	0	181	240	0	164	240	0	160	240	0

Table 7-1	Tracking	Scenarios
-----------	----------	-----------

The sampling period for each scenario was 2 seconds and the maximum number of false alarms was 3 per scan. The initial target state for each scenario was:

- $(x_0, y_0) = (15, 100 \text{ m}, 15, 100 \text{ m});$
- (v<sub>x0</sub>,v<sub>y0</sub>) = (100 m/sec, 100 m/s);
- $(a_{x0}, a_{y0}) = (0.0 \text{ m/sec}^2, 0.0 \text{ m/s}^2).$

The full-rate measurement noise covariance was  $R_k = 10,000 I$ . The number of Monte-Carlo runs for each simulation was 50. The mode/pattern Markov transition matrices for the various algorithms were:

$$h_{IMM-PDAF} = \begin{bmatrix} 0.92 & 0.03 & 0.05 \\ 0.02 & 0.9 & 0.08 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}; \quad h_{mnpf} = \begin{bmatrix} 0.7 & 0.3 & 0.0 \\ 0.025 & 0.95 & 0.025 \\ 0.0 & 0.3 & 0.7 \end{bmatrix};$$
$$h_{MR-MMPF} = \begin{bmatrix} 0.85 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.85 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.85 \end{bmatrix}; \quad h_{IMP-PDA} = \begin{bmatrix} 0.92 & 0.01 & 0.0 & 0.07 \\ 0.01 & 0.92 & 0.07 & 0.0 \\ 0.2 & 0.0 & 0.8 & 0.0 \\ 0.0 & 0.2 & 0.0 & 0.8 \end{bmatrix}.$$

## 7.1.2 Scenario-1 (a = $+/-5 \text{ m/s}^{-2}$ ) Results

Scenario-1 has relatively small target maneuvers. The patterns for the original IMPDA developed by Hong and Cui [45] are defined as:

$$p^{1} = \begin{bmatrix} 200\\ 200\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 0\\ 0\\ -10\sqrt{2}\\ -10\sqrt{2} \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 800\\ 800\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0\\ 0\\ 10\sqrt{2}\\ 10\sqrt{2} \end{bmatrix}.$$

The MRMMPF patterns are the same as the IMPDA except the order has been changed:

$$p^{1} = \begin{bmatrix} 0\\0\\-10\sqrt{2}\\-10\sqrt{2}\\-10\sqrt{2} \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 200\\200\\0\\0\\0 \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 800\\800\\0\\0\\0 \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0\\0\\10\sqrt{2}\\10\sqrt{2} \end{bmatrix}$$

This rearrangement of patterns was done for simple bookkeeping purposes and does not impact the MRMMPF algorithm. Thus, in the MRMMPF,  $p^1$  corresponds to a negative acceleration,  $p^2$ and  $p^3$  correspond to straight line motion, while  $p^4$  corresponds to a positive acceleration.

The x-y RMS position and  $v_x$ - $v_y$  RMS velocity errors for each filter are shown in Figure 7.6 through Figure 7.9. The true trajectory overlaid with measurements and false alarms is shown in Figure 7.2. Figure 7.3, Figure 7.4, Figure 7.4, and Figure 7.5 depict the RMS position errors vs. sample increment for the IMMPDAF, IMPDA, MMPF, and the MRMMPF, respectively. Figure 7.6 - Figure 7.9 depict the RMS velocity errors for each filter type. Figure 7.10 - Figure 7.11 show the pattern probabilities for the IMPDA and MRMMPF while Figure 7.13 - Figure 7.14 depict the model probabilities for each of the algorithms (i.e. CV-CA for IMMPDAF/MMPF and CH-CH<sup>2</sup> for IMPDA/MRMMPF).

Track Scenario-1: a=+/-5								
	Acceleration Scenario							
Filter Type	X RMSY RMSV_X RMSV_Y RMSErrorErrorErrorError							
IMMPDAF	57.1	57.8	9.2	9.2				
IMP-PDA	35.0	35.7	5.1	5.2				
MMPF	45.6	44.2	7.9	7.9				
MR-MMPF	30.9 31.7 5.2 5.2							

Table 7-2 RMS Position and Velocity Errors (a=+/-5)



Figure 7.1 True Track vs. Noisy Measurements + FA



Figure 7.2 IMMPDAF RMS Position Errors (a=+/-5)



Figure 7.3 IMPDA RMS Position Errors (a=+/-5)



**Figure 7.4** MMPF RMS Position Error (a=+/-5)



Figure 7.5 MRMMPF RMS Position Error (a=+/-5)



Figure 7.6 IMMPDAF RMS Velocity Errors (a=+/-5)



Figure 7.7 IMPDA RMS Velocity Errors (a=+/-5)



Figure 7.8 MMPF RMS Velocity Errors (a=+/-5)



Figure 7.9 MRMMPF RMS Velocity Errors (a=+/-5)



Figure 7.10 IMPDA Pattern Probabilities (a=+/-5)



Figure 7.11 MRMMPF Pattern Probabilities (a=+/-5)



Figure 7.12 IMMPDAF Model Probabilities (a=+/-5)



Figure 7.13 IMPDA Model Probabilities (a=+/-5)



Figure 7.14 MMPF Model Probabilities (a=+/-5)



Figure 7.15 MRMMPF Model Probabilities (a=+/5)

# 7.1.3 Scenario-2 (a = $+/-15 \text{ m/s}^{-2}$ ) Results

Scenario-2 has moderate target maneuvers. The patterns for the MRMMPF are defined as:

$$p^{1} = \begin{bmatrix} 0 \\ 0 \\ -30\sqrt{2} \\ -30\sqrt{2} \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 200 \\ 200 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 2000 \\ 2000 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0 \\ 0 \\ 30\sqrt{2} \\ 30\sqrt{2} \end{bmatrix};$$

The IMPDA patterns are defined as:

$$p^{1} = \begin{bmatrix} 200\\ 200\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 2000\\ 2000\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 0\\ 0\\ -30\sqrt{2}\\ -30\sqrt{2} \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0\\ 0\\ 30\sqrt{2}\\ 30\sqrt{2} \end{bmatrix}.$$

The *x-y* RMS position and RMS velocity errors for each filter are shown in Table 7-3. In order to save space, only the RMS position errors will be displayed for this and subsequent scenarios (since RMS position error is the key metric). Thus, Figure 7.16 - Figure 7.19 depict the RMS position errors vs. sample increment for the IMMPDAF, IMPDA, MMPF, and the MRMMPF, respectively.

Track Scenario-2: a=+/-15								
	Acceleration Scenario							
Filter Type	X RMSY RMSV_x RMSV_y RMSErrorErrorErrorError							
IMMPDAF	60.8	61.6	13.0	13.1				
IMP-PDA	49.8	49.6	13.1	13.0				
MMPF	51.4	51.0	13.2	13.2				
MR-MMPF	34.8 34.2 9.5 9.3							

 Table 7-3 RMS Position and Velocity Errors (a=+/-15)



Figure 7.16 IMMPDAF RMS Position Errors (a=+/-15)



Figure 7.17 IMPDA RMS Position Errors (a=+/-15)



Figure 7.18 MMPF RMS Position Errors(a=+/-15)



Figure 7.19 MRMMPF RMS Position Errors (a=+/-15)

# 7.1.4 Scenario-3 (a = $\pm - 25 \text{ m/s}^{-2}$ ) Results

as:

Scenario-3 has moderate target maneuvers. The patterns for the MRMMPF are defined

$$p^{1} = \begin{bmatrix} 0 \\ 0 \\ -50\sqrt{2} \\ -50\sqrt{2} \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 200 \\ 200 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 1500 \\ 1500 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0 \\ 0 \\ 50\sqrt{2} \\ 50\sqrt{2} \\ 50\sqrt{2} \end{bmatrix}.$$

The patterns for the IMPDA are defined as:

$$p^{1} = \begin{bmatrix} 200\\ 200\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 1500\\ 1500\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 0\\ 0\\ -50\sqrt{2}\\ -50\sqrt{2} \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0\\ 0\\ 50\sqrt{2}\\ 50\sqrt{2} \end{bmatrix}.$$

The *x-y* RMS position and RMS velocity errors for each filter are shown in Table 7-4. Figure 7.20 - Figure 7.23, depict the RMS position errors vs. sample increment for the IMMPDAF, IMPDA, MMPF, and the MRMMPF, respectively.

Track Scenario-3: a=+/-25								
	Acceleration Scenario							
Filtor Type	X RMS	Y RMS	V <sub>X</sub> RMS	V <sub>Y</sub> RMS				
гшег туре	Error	Error	Error	Error				
IMMPDAF	61.2	60.6	13.4	13.3				
IMP-PDA	48.0	48.0	18.9	19.0				
MMPF	48.3	48.7	12.3	12.2				
MR-MMPF	30.2 29.6 10.9 10.9							

 Table 7-4 RMS Position and Velocity Errors (a=+/-25)


Figure 7.20 IMMPDAF RMS Position Errors (a=+/-25)



Figure 7.21 IMPDA RMS Position Errors (a=+/-25)



Figure 7.22 MMPF RMS Position Errors (a=+/-25)



Figure 7.23 MRMMPF RMS Position Error (a=+/-25)

# 7.1.5 Scenario-4 (a = $+/-40 \text{ m/s}^{-2}$ ) Results

Scenario-4 has large target maneuvers. The patterns for the MRMMPF are defined as:

$$p^{1} = \begin{bmatrix} 0\\0\\-80\sqrt{2}\\-80\sqrt{2} \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 200\\200\\0\\0\\0 \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 1640\\1640\\0\\0\\0 \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0\\0\\80\sqrt{2}\\80\sqrt{2} \end{bmatrix}.$$

The patterns for the IMPDA are defined as:

$$p^{1} = \begin{bmatrix} 200\\ 200\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{2} = \begin{bmatrix} 1640\\ 1640\\ 0\\ 0\\ 0 \end{bmatrix}; \quad p^{3} = \begin{bmatrix} 0\\ 0\\ -80\sqrt{2}\\ -80\sqrt{2} \end{bmatrix}; \quad p^{4} = \begin{bmatrix} 0\\ 0\\ 80\sqrt{2}\\ 80\sqrt{2} \end{bmatrix}$$

The *x-y* RMS position and RMS velocity errors for each filter are shown in Table 7-5. Figure 7.24 - Figure 7.27 depict the RMS position errors vs. sample increment for the IMMPDAF, IMPDA, MMPF, and the MRMMPF, respectively.

Track Scenario-4: a=+/-40					
		Acceleratio	on Scenaric	)	
Filter Type	X RMSY RMSV_x RMSV_y RMSErrorErrorErrorError				
IMMPDAF	63.1	62.1	17.4	17.3	
IMP-PDA	60.9	61.6	28.3	28.3	
MMPF	56.7	57.8	16.5	16.5	
MR-MMPF	29.8	30.1	13.2	13.2	

 Table 7-5 RMS Position and Velocity Errors (a=+/-40)



Figure 7.24 IMMPDAF RMS Position Error (a=+/-40)



Figure 7.25 IMPDA RMS Position Errors (a=+/-40)



Figure 7.26 MMPF RMS Position Errors (a=+/-40)



Figure 7.27 MRMMPF RMS Position Errors (a=+/-40)

The track position RMS errors for all of the scenarios are summarized in Table 7-6 and displayed in Figure 7.28 below. Since the x-position and y-position RMS errors are nearly identical, only the x-position errors are listed.

X-Position RMS ERROR					
		Acceleration Scenario			
Filter Type	a = +\-5 a = +\- 15 a = +\- 25 a = +\- 40				
IMMPDAF	57.1	60.8	61.2	63.1	
IMP-PDA	35.0	49.8	48.0	60.9	
MMPF	45.6	51.4	48.3	56.7	
MR-MMPF	30.9	34.8	30.2	29.8	

Table 7-6 X-Position RMS Error Summary for All Scenarios



Figure 7.28 X-Position RMS Error Summary for All Scenarios

# 7.2 MRMMPF-TBD and MMPF-TBD.

# 7.2.1 Scenario-1 Description: Mildly Maneuvering Target

Scenario-1 used a mildly maneuvering target with the following simulation parameters:

- Scan Period: T = 1;
- Track Length: 123 scans with target visible from T = 4 120;
  - 1<sup>st</sup> Maneuver:  $a = 2.5 m \cdot \sec^{-2}$  (a.k.a mode CA-2) from T=25-30;
  - $2^{nd}$  Maneuver:  $a = -2.5 m \cdot \sec^{-2}$  (a.k.a mode CA-1) from T=55-60;
  - o Rest of track is CV motion;
- Initial State:  $X_0 = [8000, 200, 0, 8000, 0, 0]'$ ;
- Detection threshold:  $\tau = 0 dB SNR$ ;
- Amplitude Gating Setting:  $N_{\text{max}} = 8$  measurements;
- Sensor resolution = 100m x 100m;
- Sensor accuracy: Computed as a function of resolution and target SNR via (5.28);

• Markov mode transition matrix: 
$$P_T = \begin{bmatrix} 0.80 & 0.10 & 0.10 \\ 0.25 & 0.75 & 0.00 \\ 0.25 & 0.00 & 0.75 \end{bmatrix};$$

• Number of Monte-Carlo runs/simulation: 50.

MRMMPF-TBD performance was compared against MMPF-TBD performance for nominal particle counts of 2000, 1000, and 500 particles and for target SNR values of 10dB, and 7dB. The SNR value for each target amplitude measurement was obtained by drawing a random number from either the noise-only Rayleigh PDF in (4.13) if the target is not visible or the target + noise Rayleigh PDF in (4.12) if the target is visible. For the latter case, it should be noted that the

linear, not logarithmic form of SNR must be used in the PDF, where  $SNR = 10^{\left(\frac{SNR_{dB}}{10}\right)}$ . The algorithm for this Rayleigh random number generator can be found in Leonov and Leonov [67]. The performance metrics for this analysis were mean position root-mean-square error (RMSE), mean velocity RMSE, and mean particle count.

#### 7.2.2 Scenario-2: Non-Maneuvering Target

A second non-maneuvering scenario was also run for target SNR values of 5dB and 4dB. The goal here was to compare performance at very low SNR values and to map out the bottom end of the performance envelopes of the MRMMPF-TBD and MMPF-TBD algorithms. The simulation parameters are the same as in Scenario-1 except for the following:

•  $a = 0 m \cdot \sec^{-2}$ ;

• 
$$N_{\rm max} = 15$$
;

• 
$$P_T = \begin{bmatrix} 0.94 & 0.03 & 0.03 \\ 0.25 & 0.75 & 0.00 \\ 0.25 & 0.00 & 0.75 \end{bmatrix}$$
.

#### 7.2.3 Scenario-1 results

The MRMMPF-TBD position and velocity RMSE performance for Scenario-1 was comparable the MMPF-TBD. These results are summarized in Table 7-7. The first column of Table 7-7 lists the full-rate particle count used by the MMPF-TBD while the second column indicates the mean multirate particle count. The multirate count is lower in all cases because CH (i.e. non-maneuvering) particles are updated once in every three scans. The third column lists the ratio of multirate to full particle counts and provides a metric of the relative computational cost. The remaining columns summarize the position and velocity RMSE performance of the two algorithms.

The full-rate MMPF Position and velocity RMSE plots for the 2000 nominal particle case at SNR = 10dB are shown below in Figure 7.29 and Figure 7.30. Since these results were representative, plots for the 1000 and 500 particle cases are omitted. The main tradeoff with the MRMMPF-TBD was that although error during CV motion was lower, peak error during maneuver period was higher than for the MMPF-TBD. Both the MMPF-TBD and MRMMPF-TBD were also tested at for a 6dB SNR. At this low SNR, the performance of both algorithms was erratic and they often diverged. Consequently 6dB SNR results are not included in Table 7-7. The model probability plots for the MMPF and MRMMPF-TBD at SNR= 10dB are shown in Figure 7.31 and Figure 7.32 below.

Inspection of Table 7-7 shows that the actual particle cost for the MRMMPF-TBD was approximately 59% of the MMPF-TBD. The mean particle count vs. time plot is shown in Figure 7.33. The plot shows that the full 2000 particles are used at the third-rate (i.e. N(k+3) seconds) update points while approximately 700 particles are used at the full-rate updates (i.e. N(k+1), N(k+2) seconds) during CV motion.

During the maneuvers, more of the particles migrate from the non-maneuvering mode the maneuvering modes (i.e. CA-1 or CA-2) and the full-rate particle count increases. Since the modes of the particles are governed by Markov transition probabilities, the actual numbers will vary slightly between different runs. Thus, a mean particle count is computed and displayed as a function of time in Figure 7.33. At every third sample point, a third-rate update occurs in which all of the 2000 particles are used. The grand mean particle count is then computed and summarized in Table 7-7. A sensitivity analysis plot of position RMSE vs. particle count for both the MMPF-TBD and the MRMMPF-TBD are shown in Figure 7.34 and Figure 7.35, respectively.

#### 7.2.4 Scenario-2 Results

The results of the Scenario-2 (i.e. non-maneuvering target) are summarized in Table 7-7. MRMMPF-TBD performance is clearly superior to the MMPF-TBD for all cases except the 2000 particle/SNR=5dB case, in which case the performance is approximately equivalent. It is also evident that except for the 2000 particle/SNR=5dB case, the MMPF-TBD tracker diverged. In contrast, the MRMMPF-TBD algorithm successfully tracked the target except for the worst case (i.e. 500 particle/SNR=4dB). Since there were no target maneuvers, the MRMMPF-TBD in Scenario-2 used approximately 40% as many particles as the MMPF-TBD. This occurred because the dominant mode was CH, which is only updated once every three scans. Since there were no maneuvers in Scenario-2, model probabilities and particle counts remained nearly constant throughout the run and are therefore not displayed.

	Mean Number of Particles		Mean Time Avg. Position			Mean Time Avg. Velocity			
SNR (dB)	Full Rate	Multi Rate	Particle Ratio (Multi/Full)	RMSE Full Rate	RMSE Multi Rate	Pos. RMSE Ratio (Multi/Full)	RMSE Full Rate	RMSE Multi Rate	Vel. RMSE Ratio (Multi/Full)
	2000	1166	0.583	13.419	12.790	0.953	4.023	3.813	0.948
10	1000	585	0.585	13.631	13.396	0.983	4.162	3.914	0.940
	500	293	0.586	13.720	14.438	1.052	4.345	4.102	0.944
	2000	1188	0.594	18.973	18.128	0.955	5.130	4.856	0.947
7	1000	596	0.596	18.882	19.669	1.042	5.141	5.091	0.990
	500	299	0.599	19.716	19.257	0.977	5.449	4.926	0.904

Table 7-7 MRMMPF-TBD vs. MMPF-TBD Performance Summary: Scenario-1

Table 7-8 MRMMPF-TBD vs. MMPF-TBD Performance Summary: Scenario-2

	Mean I	Number	of Particles	Mean Time Avg. Position RMSE			Mean Time Avg. Velocity		
SNR (dB)	Full Rate	Multi Rate	Particle Ratio (Multi/Full)	RMSE Full Rate	RMSE Multi Rate	Pos. RMSE Ratio (Multi/Full)	RMSE Full Rate	RMSE Multi Rate	Vel. RMSE Ratio (Multi/Full)
	2000	796	0.398	25.384	26.119	1.029	3.838	3.810	0.993
5	1000	400	0.400	261.896	29.774	0.114	5.960	3.504	0.588
	500	199	0.398	641.476	26.502	0.041	11.458	3.921	0.342
	2000	807	0.403	388.808	41.205	0.106	10.117	4.848	0.479
4	1000	402	0.402	396.842	31.355	0.079	12.567	4.283	0.341
	500	203	0.406	703.138	420.259	0.598	16.050	10.745	0.670



Figure 7.29 MMPF-TBD vs. MRMMPF-TBD Position RMSE (2000 Particles, SNR = 10dB)



Figure 7.30 MMPF-TBD vs. MRMMPF-TBD Velocity RMSE (2000 Particles, SNR = 10dB)



Figure 7.31 MMPF-TBD Model Probabilities (2000 Particles, SNR = 10dB)



Figure 7.32 MRMMPF-TBD Model Probabilities (2000 Particles, SNR = 10dB)



Figure 7.33 MRMMPF-TBD Mean Particle Count vs. Time (2000 Particles, SNR = 10dB)



Figure 7.34 MMPF-TBD Position RMSE vs. Particle Count Sensitivity



Figure 7.35 MRMMPF-TBD Position RMSE vs. Particle Count Sensitivity

## 7.3 E-SD-MRES-PF vs. Standard Uni-Resolutional Bootstrap Filter (BPF)

The E-SD-MRES-PF was compared against a standard BPF for the three scenarios described below. The key difference between the scenarios was the type and complexity of the initial PDF. The simple scenario featured an initial PDF that was a single Gaussian. The more complex scenarios featured initial PDFs that were Gaussian sums composed of widely dispersed modes and different variances. The key performance metric was the particle efficiency ratio,  $R_{PE}$ , which is defined as the number of uni-resolution particles to multiresolutional particles for a given RMSE

performance level: 
$$R_{PE} = \frac{N_{P\_uni}}{N_{P\_mres}}$$
.

#### 7.3.1 E-SD-MRES-PF Scenario Description

#### 7.3.1.1 Scenario-1 (Complex 5-Modal Initial PDF)

Scenario-1 parameters are listed below:

- Highly non-linear state equation:  $x_{k+1/k} = 0.6x_{k/k} + \frac{5x_{k/k}}{(1+x_{k/k}^2)} + \log(x_{k/k}^2);$
- Highly non-linear measurement equation:  $z_x = 15 \tan^{-1} \left( \frac{5 x_{k/k}}{1 + x_{k/k}^2} \right);$
- Number of time steps: 30;
- Number of Monte-Carlo iterations per simulation:100;
- Initial particle count (prior to thresholding):  $N_{P0} = 1000$ ;
- Multi-Resolution thresholds varied from 0 to 10<sup>-3</sup>;
- Initial PDF: Complex Gaussian mixture with 5 widely spaced modes and widely different variances:

 $p_0 = 0.3N[x,10,4] + 0.1N[x,25,0.2] + 0.1N[x,30,0.1] + 0.2N[x,55,0.01] + 0.3N[x,95,0.3].$ 

#### 7.3.1.2 Scenario-2 (Bi-Modal Initial PDF)

All parameters are the same as Scenario-1 except for initial PDF:

• Initial PDF: Gaussian mixture with 2 widely spaced modes and different variances:

$$p_0 = 0.6N[x,10,2] + 0.4N[x,20,4]$$
.

## 7.3.1.3 Scenario-3 (Gaussian Initial PDF)

All parameters are the same as Scenario-1 except for initial PDF:

• Initial PDF: Single Gaussian:  $p_0 = N[x, 15, 4]$ .

## 7.3.2 Results

The results for Scenario-1 are summarized in Table 7-9. As is evident from Figure 7.36, the RMSE of the E-SD-MRES-PF remained nearly constant except at very low particle counts.

Scenario-1: Uni-Res vs. Multi-Res RMSE					
Multi-Res Threshold	Particle Count	Uni-Res RMSE (For Same # Particles)	Multi-Res RMSE	Delta: Multi- Res vs. Uni- Res	% RMSE Delta
1.0E-03	25	0.786	0.764	-0.023	-3.00%
1.0E-04	56	0.743	0.680	-0.063	-9.25%
1.0E-05	145	0.725	0.676	-0.050	-7.34%
1.0E-06	405	0.695	0.677	-0.018	-2.60%
0.0E+00	1008	0.680	0.655	-0.025	-3.85%

Table 7-9 Uni-Res vs. Multi-Res RMSE (Scenario-1)



Figure 7.36 Uni-Res vs. Multi-Res Performance (Scenario-1)

In contrast, the RMSE of the uni-res BPF steadily worsened as the particle count was reduced. The particle efficiency ratio,  $R_{PE}$ , was determined by extending a horizontal constant RMSE line (shown as a dotted green line in ) from the knee of the curve of the SD-MRES-PF RMSE plot to the point at which this line intersected the uni-res BPF RMSE curve. The knee of

the curve was chosen because it was at this point that SD-MRES-PF RMSE began to rapidly

degrade. In Scenario-1, this resulted in 
$$R_{PE} = \frac{N_{P\_uni}}{N_{P\_mres}} = \frac{1000}{56} = 17.9$$
.

The results for Scenario-2 are summarized in Table 7-10 and Figure 7.37. Although the SD-MRES-PF still outperformed the uni-res BPF, the improvement was much less noticeable

than in Scenario-1. In Scenario-2, this resulted in  $R_{PE} = \frac{N_{P\_uni}}{N_{P\_mres}} = \frac{700}{171} = 4.09$ .

S	Scenario-2: Uni-Res vs. Multi-Res RMSE						
Multi-Res Threshold	Particle Count	Uni-Res RMSE (For Same # Particles)	Multi-Res RMSE	Delta: Multi- Res vs. Uni- Res	% RMSE Delta		
1.0E-03	29	0.754	0.734	-0.021	-2.82%		
1.0E-04	68	0.695	0.682	-0.013	-1.87%		
1.0E-05	171	0.672	0.652	-0.020	-3.11%		
1.0E-06	475	0.656	0.650	-0.006	-0.94%		
0.0E+00	1008	0.647	0.649	0.002	0.29%		

## Table 7-10 Uni-Res vs. Multi-Res RMSE (Scenario-2)

The results for Scenario-3 are summarized in Table 7-11 and Figure 7.38. For this case, the SD-MRES-PF offered no improvement over the uni-res BPF (i.e.  $R_{PE} = 1$ ).

The particle efficiency ratios for all of the scenarios are summarized in Figure 7.39.



Figure 7.37 Uni-Res vs. Multi-Res Performance (Scenario-2)

Scenario-3: Uni-Res vs. Multi-Res RMSE					
Multi-Res Threshold	Particle Count	Uni-Res RMSE (For Same # Particles)	Multi-Res RMSE	Delta: Multi- Res vs. Uni- Res	% RMSE Delta
1.0E-03	29	0.768	0.729	-0.039	-5.34%
1.0E-04	67	0.689	0.675	-0.014	-2.02%
1.0E-05	169	0.656	0.662	0.007	0.99%
1.0E-06	467	0.641	0.648	0.007	1.07%
0.0E+00	1008	0.660	0.644	-0.016	-2.54%

Table 7-11 Uni-Res vs. Multi-Res RMSE (Scenario-3)



Figure 7.38 Uni-Res vs. Multi-Res Performance (Scenario-3)



Figure 7.39 Particle Efficiency Ratio Summary

#### 7.4 GMMPF AND K-GMMPF

The goal of the finite mixture modeling approach was to reduce particle counts while maintaining multimodality. Consequently, the GMMPF and K-GMMPF algorithms were both tested against a conventional bootstrap MMPF in a single target scenario in order to compare the particle counts vs. RMSE performance.

# 7.4.1 GMMPF AND K-GMMPF Scenario

The single target scenario used the following parameters:

- 3-dimensional state vector:  $[x, v_x, a_x]'$ , (clustering only performed on position-velocity dimensions);
- Single target with no false alarms  $(P_{FA} = 0)$ ;
- Probability of detection and gating is unity  $(P_D = 1, P_G = 1)$ ;
- Target acceleration:  $a = \pm 40$ ;
- Scan Period: T=8 seconds;
- Number of time steps per run: 60;
- Measurement error = 100m;
- Number of Runs = 50.

The performance metrics were position RMSE, velocity RMSE, and particle count. The "baseline" case was a 1,000-particle MMPF. Results for a 50 run simulation are listed in below in Table 7-12.

## 7.4.2 GMMPF and K-GMMPF Results

It is evident that both the GMMPF and K-GMMPF provided comparable RMSE performance to the MMPF but with a smaller particle count. It is interesting to note that the K-GMMPF with 200 particles performed nearly as well as the MMPF with 1000 particles but at 20% of the particle cost (Note: at 200 particles, the MMPF completely lost track of the target).

The GMMPF and K-GMMPF were also tested in a two-target scenario in which the targets nearly merged then separated. In this scenario, the filter diverged when K-means was used but maintained track when EM was used. This was the only scenario in which EM proved superior to K-means.

Tracking Scenario: Max Accel = 40					
	Num	nber of Run	s = 50		
Algorithm Number of Particles Particles Per Model RMS-X Error Error Base				Particle Cost (Relative to Baseline)	
MMPF (Baseline)	1000	N/A	69.5	17.9	1.00
GMMPF	600	3	68.9	16.5	0.60
K-GMMPF	600	3	69.6	15.5	0.60
K-GMMPF	200	3	70.7	17.8	0.20

# Table 7-12 GMMPF/K-GMMPF Results

### 7.5 Summary and Discussion of Results

Chapter 7 presented results from the simulations of the following algorithms:

- MRMMPF vs. MMPF, IMPDA, and IMMPDAF;
- MRMMPF-TBD vs. MMPF-TBD;
- Extended Spatial Domain Spatial-Domain Multi-Resolution Particle Filtering (E-SD-MRES-PF);
- Gaussian Finite Mixture Model Particle Filter (GMMPF) and Kalman GMMPF.

#### 7.5.1 MMPF and MRMMPF Summary

It is evident that the particle filter-based MMPF and MRMMPF algorithms significantly outperformed the Kalman-based algorithms. The MRMMPF outperformed all of the other algorithms in terms of both RMS position and velocity error. The key drawback of the particle filter algorithms, however, was that run times were between 2-3 orders of magnitude greater than that of Kalman-based algorithms. Consequently, next the step was to examine methods that could reduce particle filter computational costs while maintaining performance.

#### 7.5.2 MMPF-TBD and MRMMPF-TBD Summary

The MRMMPF-TBD had comparable error performance (or better performance for the low SNR cases) to the MMPF-TBD with approximately 40% to 60% of the particle cost of the latter (depending on the target maneuver scenario). This performance could probably be improved further by developing a better importance density (i.e. one that incorporates the current measurement).

#### 7.5.3 E-SD-MRES-PF Summary

The E-SD-MRES-PF provided large particle savings when the initial PDF was a complex Gaussian sum with widely dispersed modes. This is evident from the particle efficiency ratio R<sub>PE</sub>. As the number of Gaussians and the complexity of the initial PDF increased, the R<sub>PE</sub> of the E-SD-MRES-PF also increased. Conversely, for simple PDF scenarios, the E-SD-MRES-PF does not provide any R<sub>PE</sub> improvement. A plausible explanation for this trend is that complex PDFs with widely spaced modes have large areas where the PDF changes relatively slowly interspersed with a few areas in which the value of the PDF changes more rapidly. The areas that change slowly can be adequately modeled as a low-pass process and hence require relatively few samples while the few areas that change rapidly also require high-pass and high-high-pass components. The data compression ability of E-SD-MRES-PF allows it to concentrate particles on

those areas that are rapidly changing. In contrast, the regions that are changing slowly can be adequately described with fewer samples.

# 7.5.4 GMMPF AND K-GMMPF Summary

The GMMPF and K-GMMF, using K-means clustering, provided comparable single-target performance to the conventional MMPF but at substantially lower particle counts. In addition, the K-GMMPF was able to maintain track at 200 particles while the MMPF lost track at 200 particles. In a limited 2-target scenario, the K-means-based GMMPF and K-GMMPF diverged while their EM-based counterparts maintained track.

# 8 CONCLUSIONS

The dissertation explored the impact of non-Gaussian and multi-modal PDFs on target tracking. It first presented the MRMMPF tracking algorithm and determined that the MRMMPF produces a smaller RMSE than the IMMPDAF, IMPDA and full-rate MMPF (for the same particle count). During the course of this research, it became apparent that particle filter tracking algorithms were computationally costly and resulted in large runtimes. Consequently, the remainder to dissertation focused mainly on developing particle filter-based tracking algorithms that provide good performance at a reduced particle count.

The MRMMPF concept was then extended to include tracking of low SNR targets, resulting in the MRMMPF-TBD. The MRMMPF-TBD had comparable error performance to the full-rate MMPF-TBD with approximately 40% to 60% of the particle cost of the latter (depending on the target maneuver scenario). Additionally, when the MRMMPF-TBD was applied to very low-SNR, non-maneuvering targets, it provided both particle savings and much better RMSE performance than the MMPF-TBD.

The next topic examined was mutli-resolutional particle filtering. This dissertation developed an E-SD-MRES-PF that extended the basic multiresolutional PF and provided comparable RMSE performance and much lower particle costs. The E-SD-MRES-PF provided the greatest particle savings for complex, multi-modal PDFs with widely spaced modes.

The last area that the dissertation examined was particle filter applications of finite mixture models (FMM). The first two FMM-based algorithm developed were the single-target GMMPF and the K-GMMPF. Both of these algorithms provided comparable RMSE performance to the standard MMPF but at substantially lower particle cost.

# 9 FUTURE WORK

#### 9.1 MMPF-TBD and MRMMPF-TBD

This performance of MMPF-TBD and MRMMPF-TBD could probably be improved further by developing a better importance density (i.e. one that incorporates the current measurement). Both of these algorithms use the prior for the importance density. This arrangement proved adequate for the full-rate MMPF-TBD, which uses a 2-dimensional (2-D) position likelihood function. In contrast, the MRMMPF-TBD uses a 4-D position likelihood function which is quite narrow relative to the prior. It was necessary to artificially increase the process noise in order to ensure that the prior PDF provided support for the likelihood function. A better importance density would likely result in reduced RMSE.

Another potential MMPF-TBD and MRMMPF-TBD improvement would to develop an adaptive amplitude gate. At higher SNR values, the amplitude likelihood function differences between target generated 1/3-R measurements and noise-generated measurements are large. Thus, a smaller gate size should be sufficient. Conversely, at lower SNR values a larger amplitude gate is required to capture the true target measurement. An additional improvement could potentially be obtained by applying approximation methods for determining the PDF of sums of Rayleigh RVs such as are described in [68] and [69] or by computing the 1/3-R PDF numerically via a FFT-based discrete convolution of the 1-R PDFs.

Another area worth exploring would be to incorporate target feature information into the MRMMPF-TBD algorithm for the purpose of joint tracking and identification (JTID). In realistic military tracking scenarios one must consider target ID in order to avoid fratricide. JTID might also assist the data association process by helping to further separate targets from clutter.

#### 9.2 E-SD-MRES-PF

The E-SD-MRES-PF only operates on single dimension. Many particle filtering applications (especially for target tracking) require a multidimensional state vector. Thus, for an E-SD-MRES-PF to be practical for these applications, it must also be multidimensional. One possible way to accomplish this might be via a 2-D discrete wavelet transform that acted on the X-Y dimensions but did not process the velocity and acceleration components of each particle. Using the explicit method, redundant particles (based on X and Y component thresholding) would be removed. The velocity/acceleration components would then be taken "as is" from the particles that survived.

If the multi-dimensional E-SD-MRES-PF proved practical, it should then be possible to combine both multirate and multiresolutional processing in a single tracking filter, resulting in even greater particle savings.

#### 9.3 GMMPF and K-GMMPF

The GMMPF and K-GMMPF are three-dimensional tracking filters. It would be useful to extend the GMMPF and K-GMMPF to at least six dimensions so that they could process entire target state vectors. It would also be useful to provide them with the ability to track multiple targets. The main roadblock to this appears to be the EM algorithm. At dimensions greater than two, the EM algorithm often became unstable and generated poorly conditioned or even singular covariance matrices. In addition, EM is much slower than K-means. Thus, research that focused on developing a more stable and faster variant of the EM algorithm would be a useful endeavor in future GMMPF development.

# **10 REFERENCES**

- Shanmugan, S., Breipohl, A., Random Signals Detection, Estimation and Data Analysis, John Wiley and Sons, New York, NY, 1988.
- Fortman, T., and Bar Shalom, Y., "Multi-target Tracking Using Joint Probabilistic Data Association", *Proceedings of the 1980 IEEE Conference on Decision and Control*, Albuquerque, NM, December 1980, pp. 807-812.
- Bar Shalom, Y., Fortman, T., *Tracking and Data Association*, Academic Press, San Diego, Ca, 1988.
- 4. Bar-Shalom, Y., Editor, *Multitarget-Multisensor Tracking: Advanced Applications,* Artech House, Norwood, MA, 1990.
- 5. Bar-Shalom, Y., and Blair, W., Editors, *Multitarget-Multisensor Tracking: Applications and Advances Volume III,* Artech House, Norwood, MA, 2000.
- 6. Bar-Shalom, Y., and Li, X.R., *Estimation and Tracking: Principles, Techniques, and Software,* Artech House, Norwood, MA, 1993.
- 7. Bar-Shalom, Y., and Li, X.R., *Multitarget-Multisensor Tracking: Principles and Techniques,* Artech House, Norwood, MA, 1993.
- Blackman, S., *Multiple-Target Tracking with Radar Application*, Artech House, Norwood, MA, 1986.
- Blackman, S., Design and Analysis of Modern Tracking Systems, Artech House, Norwood, MA, 1999.

- Reid, D.B., "An Algorithm for Tracking Multiple Targets", *IEEE Transactions on Automatic Control*, Vol. 24, No. 6, pp. 843-854, December 1979.
- Fitzgerald, R, "Track Biases and Coalescence with Joint Probabilistic Data Association", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-21, November 1985, pp. 822-825.
- H. Gauvrit, J.P. Le Cadre, and J. Jauffret, "A Formulation of Multitarget Tracking as an Incomplete Data Problem", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, No. 4, pp. 1242-1257, October 1997
- Alspach, D.L., and Sorenson, H., "Non-Linear Bayesian Estimation Using Gaussian Sum Approximations", *IEEE Transactions on Automatic Control*, Vol. AC-17, No. 4, pp. 439-448, August 1972.
- 14. Alspach, D.L., and Sorenson, H., "Recursive Bayesian Estimation Using Gaussian Sums", *Automatica*, Vol. 7, pp. 465-479, August 1972.
- Caputi, M., "Non-Gaussian Estimation Using a Modified Gaussian Sum Adaptive Filter", Ph.D. Dissertation, Bradley Dept. of Electrical Engineering, Virginia Polytechnic Institute and State University, Blackburg, 1991.
- Caputi, M., and Moose, M., "A Modified Gaussian Sum Approach to Estimation of Non-Gaussian Signals", IEEE Transactions on Aerospace and Electronic Systems, Vol. 29, No. 2, pp. 446-451, April 1993.
- Tam, W., and Plataniotis, K.N., D. Hatzinakos, "An Adaptive Gaussian Sum Algorithm for Radar Tracking", Elsevier Signal Processing, Vol. 77, pp. 85-104, 1999.
- Kotecha, J.H. and Djuric, P.M., "Gaussian Sum Particle Filtering", Submitted to: *Transactions on Signal Processing*, Feb 2001.

- 19. Ito, K. and Xiong, K., "Gaussian Filters for Nonlinear Filtering Problems", *IEEE Transactions on Automatic Controls*, Vol. 45, No. 5, pp. 910-927, May 2000.
- Tam, W., and Hatzinakos, D., "An Efficient Radar Tracking Algorithm Using Multidimensional Gauss-Hermite Quadratures", 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), Vol. 5, Apr 1997 pp. 3777-3780, April 1997.
- Challa, S., Bar Shalom, Y., and Krishnamurthy, V., "Non-Linear Filtering using Gauss-Hermite Quadrature and Generalized Edgeworth Series, *Proceedings of the 1999 American Control Conference*, pp. 3397-3401, June 1999.
- 22. Reece, S., "Nonlinear Kalman Filtering with Semi-Parametric Biscay Distributions", *IEEE Transactions on Signal Processing*, Vol. 49, No. 11, pp. 2245-2253, November 2001.
- Julier, S.J., Uhlmann, J. K, and Durrant-Whyte, H.F., "A New Approach for Filtering Nonlinear Systems", *Proceedings of the American Control Conference*, Seattle, Washington, pp. 1628-1632, June 1995.
- Julier, S.J.,. Uhlmann, J. K, and Durrant-Whyte, H.F., "A New Approach for Non-Linear Transformation of Means and Covariances in Filters and Estimators, *IEEE Transactions on Automatic Control - Technical Notes and Correspondence*, Vol. 45, No. 3, pp. 477-482, March 2000.
- Julier, S.J.,. and Uhlmann, J. K, "Reduced Sigma Point Filters for the Propagation of Means and Covariances Through Non-Linear Transformations", *Proceedings of the American Control Conference*, Anchorage, Alaska, pp. 887-892, May 2002.
- 26. Gordon, N., Salmond, D., and Smith, A., "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation", IEE Proceedings-F140(2), pp. 107-113, 1993.

- Carpenter, J., Clifford, P., and Fearnhead, P., "Improved particle Filter for Non-Linear Problems, *IEE Proceedings on Radar, Sonar and Navigation*, Vol 146 Issue 1, pp. 2-7, Feb 1999.
- McGinnity S., and Irwin, P., "Multiple Model Bootstrap Filter for Maneuvering Target Tracking", IEEE Transactions on Aerospace and Electronic Systems, Vol 36, pp. 1006-1012, 2000.
- 29. McGinnity S., and Irwin, P., "Maneuvering Target Tracking Using a Multiple Model Bootstrap Filter", *Sequential Monte Carlo Methods in Practice,* Chapter 23, Springer, New York, NY, 2001.
- Djuric, P., and Kotcha, J., "Gaussian Sum Particle Filtering for Dynamic State Space Models", Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 6, pp. 3465-3468, 2001.
- Everett, N., Gordon, N., and Salmond, D., "Target Tracking and Guidance Using Particles", *Proceedings of the 2001 American Control Conference*, Vol 6, pp. 4387-4392, 2001.
- Musick, S., Greenewald, J., Kastella, K., and Kreucher, C., "Comparing Numerical Methods for Dim Target Tracking (A Challenge problem)", *Presentation given at: Fusion* 2001, 10 August 2001.
- 33. Hue, C., Le Cadre, J., and Perez, P., "A Particle Filter to Track Multiple Objects", *Proceedings of the 2001 IEEE Workshop on Multi-Object Tracking*, pp. 61-68, 2001.
- Higuchi, T., Ichimura, N., Ikoma, N., and Maeda, H., "Maneuvering Target Tracking by Using particle Filter", *Joint 9<sup>th</sup> IFSA World Congress and 20<sup>th</sup> NAFIPS International Conference*, Volume 4, pp. 2223-2228, 2001.

- 35. Karlsson, R., and Gustafsson, F., "Monte-Carlo Data Association for Multiple Target Tracking", *IEE Target Tracking: Algorithms and Applications*, Netherlands, Oct. 2001.
- Arulampalam, M.S., Clapp, T., Gordon, N., and Maskell, S, "A Tutorial on Particle Filters for Online Non-linear/Non-Gaussian Bayesian Tracking, *IEEE Transactions on Signal Processing*, Vol. 50, Issue 2, pp. 174-188, February 2002.
- 37. Doucet, A., De Freitas, N., and Gordon, N., (editors), Sequential Monte Carlo Methods in *Practice*, Springer, New York, NY, 2001.
- Bergman, N., Forssell, U., Gunnarsson, F., Gustafsson, F., Jansson, J., Karlsson, R., and Nordlund, P., "Particles for Positioning, Navigation, and Tracking", *IEEE Transactions on Signal Processing*, Vol. 50, Issue 2, pp. 174-188, February 2002.
- Gordon, N., Kirubarajan, T., Mallack, T., and Maskell, S, . "Littoral Tracking Using Particle Filter", *Proceedings of the 5<sup>th</sup> International Conference on Information Fusion*, Vol 2, pp. 935-942, 2002.
- 40. Farina, A., Ristic, B., and Benvenuti, D., "Tracking a Ballistic target: Comparison of Several Nonlinear Filters", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 3, pp. 854-867, July 2002.
- Blom, H., and Bloem, E., "Joint IMMPDA Particle Filter", *Proceedings of the 6<sup>th</sup> International Conference on Information Fusion*, Cairns, Queensland, Australia, July 8-11, 2003.
- 42. Maskell, S., Rollason, M., Gordon, N. and Salmond, D., "Efficient Particle Filtering for Multiple Target Tracking with Application to Tracking in Structured Images", *SPIE Signal and Data Processing of Small Targets,* Vol. 4728, Orlando, FI, pp. 1-12, April 1-5, 2002.
- 43. Frank, O., Nieto, J., Guivant, J., and Scheding, S., "Multiple Target Tracking using Sequential Monte Carlo Methods and Statistical Data Association", Proceedings of the

2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, NV, pp. 2718-2723, October 2003.

- 44. Vermaak, J., Doucet, A., and Perez, P., "Maintaining Multi-Modality through Mixture Tracking" (pre-publication white paper), 2003.
- 45. Hong, L., Cui, N., Cong, S., and Wicker, D., "An Interacting Multipattern Data Association (IMPDA) Tracking Algorithm", *Elsevier Signal Processing 71*, pp. 55-77, 1998.
- 46. Cui, N., Hong, L., and Layne, J., "A Comparison of Non-Linear Filtering Approaches with Application to Ground Target Tracking", (pre-publication white paper), 2003.
- Mars, A., Maskell, S., and Bar-Shalom, Y., "Expected Likelihood for Tracking in Clutter with Particle Filters", SPIE Signal and Data Processing of Small Targets 2002, Vol 4728, Orlando, April 1-5, pp. 1-10, 2002
- 48. Mallick, M., and Kirubarajan, T., "Multi-Sensor Single Target Bearing Only Tracking in Clutter", Defense Technical Information Service, October 2001.
- Verbeek, J., Vlassis, N., and Krose, B., "Efficient greedy learning of Gaussian Mixture Models", Neural Computation, Vol 15(2), pp. 469-485, February 2003.
- Schultz, D., Burgard, W., Fox, D., and Cremers, A.B., "Tracking Multiple Moving Targets with a Mobile Robot Using Particles and Statistical Data Association", IEEEE Intl. Conf. Robotics and Automation, 2001.
- 51. Hong, L., "Multirate Interacting Multiple Model Filtering for Target Tracking Using Multirate Models", *IEEE Transactions Autom. Control*, Vol. 44, No. 7, pp. 1326-1340, 1999.
- Hong, L., Cui, N., Bakich, M., and Layne, J.R., "Multirate Interacting Multiple Mode particle Filter for Terrain-Based Ground Target Tracking", *IEE Proc.-Control Theory Appl.* Vol. 153, No. 6, pp. 721-731, 2006.

- 53. Hong, L. and Wicker, D., "A Spatial-Domain Multiresolutional Particle Filter with Thresholded Wavelets", Signal Processing, Volume 87, Issue 6.
- 54. Salmond, D., Birch, H., "A Particle Filter for Track before Detect", *Proceedings of the American Control Conference*, Arlington, pp. 3755-3760, VA, June 2001.
- 55. M Rollason, D Salmond, "A Particle For Track-Before-Detect Of A Target With Unknown Amplitude", Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE Workshop Target Tracking: Algorithms and Applications, 16 October 2001, Enschede, Netherlands.
- 56. Ristic, B., "Detection and Tracking of Stealthy targets", in *Beyond the Kalman Filter: Particles for Tracking Applications.* Artech House, Ch. 11, 2004.
- 57. Oii, Y., Doucet, B., Vo, No., Ristic, B., "Particle Filter for Tracking Linear Gaussian Target with Non-Linear Observations", *Proc. SPIE (I. Kadar, ed),* Vol. 5096, 2003.
- Rutten, M., Gorden, N., and Maskell, S., "Efficient Particle-Based Track-Before-Detect in Rayleigh Noise", *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, June 28 to July 1, 2004.
- Vermaak, J., Godsill, S., and Perez, P., "Monte Carlo Filtering for Multi-Target Tracking and Data Association", IEEE Transactions on Aerospace and Electronic Systems, Vol. 41, No. 1, Jan 2005, pp. 309-332.
- Zhan Y., Tieng T. and Chin C., "PDF of the Sum of Rician Signals and Gaussian Noise", Vehicular Technology Conference, 2003, pp. 21 – 25, Vol. 1, 6-9 Oct. 2003.
- Hu, J., and Beaulieu, N., "Accurate Simple Closed-Form Approximations to Rayleigh Sum Distributions and Densities", IEEE Communications Letters, Vol. 9, No. 2, pp. 109-111, February 2005.
- Bar-Shalom,Y., Li., X.R., Multitarget-Multisensor Tracking: Principles and Techniques, Artech House, Norwood, MA, 1993.

- Boers Y. and Driessen H, "Particle Filter Based Track Before Detect Algorithms", Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets, Vol. 5204, San Diego, CA, Aug. 2003, pp 20-30.
- Doucet, A., De Freitas, N., Gordon, N., "An Introduction to Sequential Monte Carlo Methods", Sequential Monte Carlo Methods in Practice, Ch. 1, Springer, New York, NY, 2001.
- 65. Ristic, B., "A Tutorial on Particle Filtering", in Beyond the Kalman Filter: Particles for Tracking Applications. Artech House, Ch. 3, Norwood, MA, 2004.
- N. Gordon, N., Salmond, D., Smith, A., "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation", IEE Proceedings-F140(2), pp. 107-113, 1993.
- 67. S. Leonov, and A. Leonov, Handbook of Computer Simulation in Radio Engineering, Communications, and Radar, Artech House, pp. 430, Norwood, MA, 2001.
- Zhan Y., Tieng T. and Chin C., "PDF of the Sum of Rician Signals and Gaussian Noise", Vehicular Technology Conference, 2003, pp. 21 – 25, Vol. 1, 6-9 Oct. 2003.
- Hu, J., and Beaulieu, N., "Accurate Simple Closed-Form Approximations to Rayleigh Sum Distributions and Densities", *IEEE Communications Letters*, Vol. 9, No. 2, pp. 109-111, February 2005.
- Boers Y. and Driessen H, "Particle Filter Based Track Before Detect Algorithms", Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets, Vol. 5204, San Diego, CA, Aug. 2003, pp 20-30.
- 71. Hong, L., and Wicker, D., "A Spatial-Domain Multiresolutional Particle Filter with Thresholded Wavelets", *Signal Processing*, Vol. 87, pp. 1384-1401, 2007.

# **APPENDIX-A: LIST OF ABBREVIATIONS**

1-R	Full-rate
1/3-R	Third Rate
BPF	Bootstrap Particle Filter
BDF	Distribution filter
CA	Constant Acceleration
СН	Highpass
CH <sup>2</sup>	Constant High-Highpass
CV	Constant Velocity
DWT	Discrete Wavelet Transform
EKF	Kalman Filter
E-SD-MRES-PF	Extended Spatial Domain Multi-Resolution Particle Filter
FMM	Finite Mixture Model
GMMPF	Gaussian Mixture Model Particle Filter
GHF	Gauss-Hermite Filter
GSF	Gaussian Sum Filter
IDWT	Inverse Discrete Wavelet Transform
IMM	Interacting Multiple Model
IMMPDAF	Interacting Multiple Model Probabilistic Data Association Filter
IMPDA	Interacting Multipattern Data Association
JPDA	Joint Probabilistic Data Association
K-GMMPF	Kalman Gaussian Mixture Model Particle Filter

KF	Kalman Filter
MMPF	Multiple Model Particle Filter
MMPF-TBD	Multiple Model Particle Filter Track Before Detect
MRMMPF	Multirate Multiple Model Particle Filter
MRMMPF-TBD	Multirate Multiple Model Particle Filter Track Before Detect
PDA	Probabilistic Data Association
PDF	Probability density function
PF	Particle Filter
SD-MRES-PF	Spatial Domain Multi-Resolution Particle Filter
SIR	Sample Importance Resample
TBD	Track Before Detect
UKF	Unscented Kalman Filter