

MULTITARGET-MULTISENSOR TRACKING: PRINCIPLES AND TECHNIQUES

1995

Yaakov Bar-Shalom
Dept. of Electrical and Systems Engineering
University of Connecticut
Storrs, Connecticut

Xiao-Rong Li
Dept. of Electrical Engineering
University of New Orleans
New Orleans, Louisiana



**MULTITARGET-MULTISENSOR TRACKING:
PRINCIPLES AND TECHNIQUES**

1995

(3rd printing)

©1995, ISBN 0-9648312-0-1

Yaakov Bar-Shalom

Box U-157, Storrs, CT 06269-2157

Phone: 860-486-4823

Fax: 860-486-5585

E-mail: ybs@ee.uconn.edu

OUTLINE

These lecture notes in *textgraph*TM format — a completely self-contained text suitable for use as viewgraphs as well — cover the topic of the estimation of the states of targets in **surveillance systems** operating in a multitarget-multisensor environment. This problem is characterized by measurement origin uncertainty — typical for **low observables** — in addition to the usual noises in the state equations. Such a situation occurs in an environment where there is clutter or the false alarm rate is high or due to the presence of several targets in the same neighborhood.

The tools for *evaluation* and *design* of algorithms for **measurement association** and **tracking** are presented. Explicit consideration is given for measurements obtained from different sensors. These techniques form the basis for the design of automated decision systems in a multitarget-multisensor situation. The modeling accounts for target maneuvers, detection probability, false alarms, interference from other targets and the finite resolution capability of sensors. The problems of track formation, track maintenance and track-to-track association and fusion in a multisensor situation are considered. The optimization of certain signal processing parameters based on tracking performance is also discussed. Many of these techniques have applications to state estimation when using multiple sensors in control systems, e.g., autonomous vehicles and robotics.

Lemma.

We have to learn to live with uncertainty.

Theorem.

We can model uncertainty.

Corollary.

We can make a living out of uncertainty.

Contents

PREFACE	xv
FOREWORD	xv
1 INTRODUCTION	1
1.1 BACKGROUND	1
1.1.1 Estimation, Decision and Tracking	1
1.1.2 Surveillance Systems	3
1.2 TRACKING AND DATA ASSOCIATION	5
1.2.1 Tracking	5
1.2.2 Data Association	6
1.2.3 Some Terminology	7
1.3 SCOPE OF THE TEXT	8
1.3.1 Objectives	8
1.3.2 Overview and Prerequisites	9
1.4 OVERVIEW OF TRACKING PROBLEMS AND ALGORITHMS — A GUIDED TOUR	11
1.4.1 The Types of Problems	11
1.4.2 Track Formation with a Single Sensor	13
1.4.3 Track Maintenance with a Single Sensor without Measurement Origin Uncertainty	15
1.4.4 Track Maintenance with a Single Sensor with Measurement Origin Uncertainty	17
1.4.5 Track Formation with Multiple Sensors	20
1.4.6 Track Maintenance with Multiple Sensors	21
1.5 OVERVIEW OF THE BASIC TRACKING ALGORITHMS	23
1.5.1 Kalman Filter Overview	23
1.5.2 Alpha-Beta-Gamma Filter Overview	26
1.5.3 Extended Kalman Filter Overview	28
1.5.4 Multiple Model Approach Overview	32
1.6 CONSISTENT COORDINATE CONVERSION IN TRACKING	36
1.6.1 The Coordinate Problem	36
1.6.2 The Errors in the Converted Measurements	38
1.6.3 Consistency Evaluation of the Debiased and Standard Conversions	42
1.6.4 Tracking with Converted Measurements vs. EKF	46
1.6.5 Coordinate Conversion — Summary	55
1.7 EXAMPLE OF DESIGN OF AN IMM ESTIMATOR	56
1.7.1 A Generic ATC Tracking Problem	56
1.7.2 Mode Selection for the IMM Design	57
1.7.3 Design Parameter Selection — Filter Tuning	58
1.7.4 Simulation Results and Discussion	59

CONTENTS

1.7.5	IMM Estimator Design — Summary	66
1.8	ADAPTIVE BEAM POINTING CONTROL OF A PHASED ARRAY RADAR WITH AN IMM ESTIMATOR	67
1.8.1	The Radar Beam Pointing Problem	67
1.8.2	The Approach to Beam Pointing Control	70
1.8.3	IMM Estimator Design for Variable Sampling Interval	71
1.8.4	The Adaptive Sampling Policy	73
1.8.5	The Algorithm for Beam Pointing Control	74
1.8.6	Simulation Results	76
1.8.7	Radar Beam Control — Summary	83
1.9	BIBLIOGRAPHICAL NOTES	84
2	MULTIPLE SENSORS, CLUTTER AND MULTIPLE TARGETS	85
2.1	INTRODUCTION	85
2.1.1	Outline	85
2.1.2	Summary of Objectives	86
2.2	TRACKING WITH MULTIPLE SENSORS	87
2.2.1	The Multisensor Problem	87
2.2.2	Sequential Updating	88
2.2.3	Parallel Updating	89
2.2.4	State Update with Out of Sequence Measurements	91
2.2.5	Multisensor Tracking — Summary	93
2.3	VALIDATION OF THE MEASUREMENTS	94
2.3.1	The Association Problem	94
2.3.2	The Validation Region	95
2.3.3	Measurement Validation — Summary	97
2.4	TARGETS AND CLUTTER	98
2.4.1	A Single Target in Clutter	98
2.4.2	Multiple Targets in Clutter	100
2.5	A CLUTTER MODEL	102
2.5.1	The Model for False Detections	102
2.5.2	The Number and Location of False Alarms	103
2.6	TRACK FORMATION IN CLUTTER	104
2.6.1	A Logic-Based Track Formation Procedure	104
2.6.2	The Target Model	106
2.6.3	Markov Chain Model of the Track Formation Process	107
2.6.4	Numerical Examples	111
2.6.5	A Design Example	115
2.6.6	Cascaded Logic Track Formation Evaluation — Summary	116
2.6.7	Automatic Synthesis of the Markov Chain for Track Formation Evaluation	117
2.7	NOTES AND PROBLEMS	118
2.7.1	Bibliographical Notes	118
2.7.2	Problems	118
3	ALGORITHMS FOR TRACKING A SINGLE TARGET IN CLUTTER	119
3.1	INTRODUCTION	119
3.1.1	Outline	119
3.1.2	Summary of Objectives	121
3.1.3	The Model for Tracking a Single Target in Clutter	122
3.2	USE OF THE STANDARD FILTER	123
3.3	THE TRACK SPLIT FILTER — A MARGINAL LIKELIHOOD FUNCTION APPROACH	124

CONTENTS

3.3.1	The Track Splitting Approach	124
3.3.2	The Likelihood Function of a Track	125
3.3.3	The Track Splitting Approach — Summary	128
3.4	A SUBOPTIMAL BAYESIAN ALGORITHM: THE PROBABILISTIC DATA ASSOCIATION FILTER	129
3.4.1	Introduction to Probabilistic Data Association	129
3.4.2	The State Estimation	131
3.4.3	The Probabilistic Data Association	134
3.4.4	Summary of the Algorithm	137
3.4.5	A Nonlinear Multisensor Tracking in Clutter Example	138
3.4.6	Consistency and Robustness Evaluation of a Parametric and Nonparametric PDAF	142
3.4.7	Augmented Probabilistic Data Association with Feature Measurements	151
3.4.8	The PDAF — Summary	153
3.4.9	The Covariance Update in the PDAF	155
3.4.10	Probability Evaluations for the PDA	157
3.4.11	Probabilistic Data Association Filter Overview	158
3.5	THE OPTIMAL BAYESIAN APPROACH	162
3.5.1	The Optimal Approach	162
3.5.2	The Estimation	163
3.5.3	Evaluation of the Probabilities of Histories	165
3.5.4	Suboptimal Versions	166
3.5.5	Optimal Bayesian Approach — Summary	167
3.6	TRACKING OF A TARGET IN THE PRESENCE OF WAKE	168
3.6.1	The Wake Problem	168
3.6.2	The Estimation Algorithm	169
3.6.3	Tracking in the Presence of a Wake — Summary	170
3.7	TRACK FORMATION WITH BEARING AND FREQUENCY MEASUREMENTS IN THE PRESENCE OF FALSE DETECTIONS	171
3.7.1	The Problem	171
3.7.2	Notations	172
3.7.3	Statistical Assumptions	173
3.7.4	The Likelihood Function with False Measurements	174
3.7.5	The CRLB in the Presence of False Measurements	176
3.7.6	Test for Track Acceptance	177
3.7.7	Numerical Results	179
3.7.8	Low SNR Track Formation with a Passive Sonar — Summary	183
3.8	NOTES AND PROBLEMS	184
3.8.1	Bibliographical Notes	184
3.8.2	Problems	184
4	MANEUVERING TARGET IN CLUTTER	187
4.1	INTRODUCTION	187
4.1.1	Outline	187
4.1.2	Summary of Objectives	189
4.2	USE OF THE PDAF WITH A FEATURE MEASUREMENT FOR MANEUVERING TARGET TRACKING	190
4.2.1	The Feature Measurement and the Augmented PDA	190
4.2.2	Examples	192
4.2.3	PDAF with Feature Measurements — Summary	198
4.3	THE MULTIPLE MODEL PDAF	199
4.3.1	Introduction	199

CONTENTS

4.3.2	The MMPDAF Algorithm	200
4.3.3	Application of the MMPDAF to Track a Real Target	202
4.3.4	The MMPDAF — Summary	208
4.4	MULTIPLE MODEL TRACK FORMATION AND MAINTENANCE	209
4.4.1	Track Formation, Maintenance and Termination with the IMMPDAF	209
4.4.2	Automatic Track Formation with the IMMPDAF	212
4.4.3	Simulation Results	214
4.4.4	A Real-Data Electro-Optical Sensor Problem	217
4.4.5	Track Formation with Multiple Models — Summary	225
4.4.6	Performance Evaluation of the Automatic Track Formation Algorithm	226
4.5	MULTISENSOR TRACKING OF A MANEUVERING TARGET IN CLUTTER	227
4.5.1	Introduction	227
4.5.2	The Target and Sensor Models	228
4.5.3	The Multisensor PDAF	230
4.5.4	Integration of the IMM and the Multisensor PDAF	232
4.5.5	An Example with Sea Skimmers	236
4.5.6	The IMM/MSPDAF — Summary	246
4.6	INTERACTING MULTIPLE MODEL TRACKING WITH TARGET AMPLITUDE FEATURE	247
4.6.1	Introduction	247
4.6.2	Use of Amplitude Information in the PDAF	249
4.6.3	The IMMPDAFAI	251
4.6.4	Target Motion Models	252
4.6.5	The Mixing with Models of Different Dimension	256
4.6.6	The Simulations	257
4.6.7	Results for Track Formation	258
4.6.8	Results for Track Maintenance	267
4.6.9	The IMMPDAFAI — Summary	276
4.7	NOTES AND PROBLEMS	277
4.7.1	Bibliographical Notes	277
4.7.2	Problems	277
5	TRACKING PERFORMANCE PREDICTION AND DETECTION THRESHOLD OPTIMIZATION	279
5.1	INTRODUCTION	279
5.1.1	Outline	279
5.1.2	Summary of Objectives	281
5.2	THE COVARIANCE OF THE COMBINED INNOVATION IN THE PDAF	282
5.2.1	The Combined Innovation	282
5.2.2	The Decrease Factor	284
5.3	TRACKING ACCURACY IN CLUTTER	285
5.3.1	The Stochastic Riccati Equation	285
5.3.2	Approximation of the Stochastic Riccati Equation	286
5.3.3	Tracking Accuracy in Clutter — Summary	289
5.4	THE TRACKER OPERATING CHARACTERISTIC	290
5.4.1	Iteration of the Modified Riccati Equation	290
5.4.2	Tracker Operating Characteristic — Summary	293
5.5	A HYBRID ALGORITHM FOR PREDICTION OF TRACKING PERFORMANCE IN CLUTTER	295
5.5.1	The Hybrid Nature of the Performance Prediction Problem	295
5.5.2	The Hybrid Approach	296
5.5.3	Track Loss	299

CONTENTS

5.5.4	Example of Track Loss Prediction	300
5.5.5	Detection Threshold Optimization	302
5.5.6	Hybrid Performance Prediction — Summary	304
5.6	BIBLIOGRAPHICAL NOTES	305
6	MULTIPLE TARGETS IN CLUTTER: BAYESIAN APPROACHES	307
6.1	INTRODUCTION	307
6.1.1	Outline	307
6.1.2	Summary of Objectives	309
6.2	THE JOINT PROBABILISTIC DATA ASSOCIATION FILTER	310
6.2.1	Introduction	310
6.2.2	The Feasible Joint Events	312
6.2.3	Evaluation of the Joint Probabilities	314
6.2.4	The Parametric and Nonparametric JPDA	317
6.2.5	The State Estimation	319
6.2.6	Passive Sonar Tracking of Two Crossing Targets in Clutter	320
6.2.7	A Modification of the JPDAF: Coupled Filtering	328
6.2.8	The JPDAF — Summary	330
6.2.9	Joint Probabilistic Data Association Filter Overview	332
6.3	THE MULTIPLE HYPOTHESIS TRACKING	334
6.3.1	Introduction	334
6.3.2	The Hypothesis Generation Technique	335
6.3.3	Evaluation of the Hypothesis Probabilities	336
6.3.4	Clustering and Hypotheses Reduction Techniques	340
6.3.5	Inclusion of Occasional Measurements	341
6.3.6	Example of Track Initiation	342
6.3.7	Example of Crossing Tracks	344
6.3.8	Example of a High Target Density Environment	346
6.3.9	Correct Association Probability	347
6.3.10	Monte Carlo Runs	349
6.3.11	The MHT — Summary	350
6.3.12	Multiple Hypothesis Tracking Overview	352
6.4	MULTIPLE TARGET TRACKING WITH POSSIBLY UNRESOLVED MEASUREMENTS	355
6.4.1	Introduction	355
6.4.2	The Model for the Merged Measurements	356
6.4.3	The Data Association	358
6.4.4	The Association Probabilities	360
6.4.5	The State Estimation	361
6.4.6	Application to Crossing Targets	363
6.4.7	The JPDAM — Summary	366
6.4.8	The PDF of the Merged Measurements	367
6.4.9	Evaluation of the Association Probabilities	370
6.5	BIBLIOGRAPHICAL NOTES	372
7	MULTIPLE TARGETS IN CLUTTER: NON-BAYESIAN APPROACHES	373
7.1	INTRODUCTION	373
7.1.1	The Non-Bayesian Approach: an Assignment Problem	373
7.1.2	Outline	375
7.1.3	Summary of Objectives	376
7.2	A JOINT LIKELIHOOD FUNCTION METHOD FOR TRACK FORMATION VIA ASSIGNMENT	377

CONTENTS

7.2.1	The Model and Assumptions	377
7.2.2	Partitioning of the Measurements	378
7.2.3	The Joint Likelihood Function and the Likelihood Ratio	381
7.2.4	Extension to Incomplete Tracks	384
7.2.5	Track Formation for Ballistic Objects from a Single Passive Sensor	385
7.2.6	Track Formation with Joint Likelihood Function — Summary	387
7.3	A LOGIC-BASED MULTITARGET TRACK INITIATOR	388
7.3.1	Track Start-Up	388
7.3.2	Track Continuation	392
7.3.3	An Exoatmospheric Ballistic Missile Defense Application	398
7.3.4	A Logic Based Initiation — Summary	401
7.4	EVALUATION OF A COMPOSITE LOGIC FOR TRACK FORMATION: A CASE STUDY	403
7.4.1	Introduction	403
7.4.2	The Association Gates	405
7.4.3	Single Target in the Absence of False Alarms	407
7.4.4	Evaluation in a Realistic Environment	409
7.4.5	The Common Gate-History Algorithm	410
7.4.6	False Tracks in the Absence of Targets	416
7.4.7	Single Target in the Presence of False Alarms	418
7.4.8	Multiple Targets and False Alarms	422
7.4.9	Composite Logic Evaluation — Summary	426
7.5	BIBLIOGRAPHICAL NOTES	428
8	MULTISENSOR TRACKING AND DATA FUSION	429
8.1	INTRODUCTION	429
8.1.1	Outline	429
8.1.2	Summary of Objectives	431
8.2	MULTISENSOR TRACKING CONFIGURATIONS	432
8.2.1	Introduction	432
8.2.2	Single Sensor Tracking	433
8.2.3	Single Sensor Tracking Followed by Track Fusion	434
8.2.4	Static Intersensor Association Followed by Central Processing	436
8.2.5	Centralized Tracking	437
8.2.6	A Hierarchical Hybrid Configuration for Multisensor-Multisite Tracking	438
8.2.7	Multisensor Configurations — Summary	439
8.3	TESTING AND FUSION OF INDEPENDENT TRACKS	440
8.3.1	Introduction	440
8.3.2	The Test for Track Association	441
8.3.3	The Fusion of the Estimates	442
8.3.4	A Fusion Example	444
8.3.5	Fusion Under the Independence Assumption — Summary	446
8.4	TESTING AND FUSION OF DEPENDENT TRACKS	447
8.4.1	Introduction	447
8.4.2	The Cross-Covariance of the Estimation Errors	448
8.4.3	The Testing for Track Association	449
8.4.4	The Fusion of the Estimates	450
8.4.5	Evaluation of the Effect of the Cross-Covariance on the Intersensor Fusion	451
8.4.6	An Approximation of the Cross-Covariance	455
8.4.7	Fusion Accounting for Dependence — Summary	456
8.5	MULTISENSOR FUSION OF FEATURE ESTIMATES	457
8.5.1	The Single Sensor Feature Estimation	457

CONTENTS

8.5.2 Multisensor Feature Estimation	459
8.5.3 Feature Estimate Fusion — Summary	461
8.6 DISTRIBUTED APPROACHES TO MULTISENSOR TRACKING WITH OPTIMAL INFORMATION FUSION	462
8.6.1 Optimal Distributed Estimation with a Central Estimator	462
8.6.2 Distributed Estimation without a Central Estimator	466
8.6.3 Information Fusion — Summary	468
8.7 MULTISENSOR AIR TRAFFIC SURVEILLANCE — A CASE STUDY	469
8.7.1 Introduction	469
8.7.2 Description of the Multisensor Data	470
8.7.3 The Sensor Measurement Statistics	472
8.7.4 Design of the Tracking Filter	473
8.7.5 Data Association	479
8.7.6 Real Data Examples	482
8.7.7 Multisensor Air Traffic Surveillance — Discussion	499
8.8 A MULTISENSOR-MULTITARGET DATA ASSOCIATION ALGORITHM FOR HETEROGENEOUS SENSORS	500
8.8.1 Introduction	500
8.8.2 Problem Description	501
8.8.3 Problem Formulation	503
8.8.4 Solution Approach	504
8.8.5 The Relaxation Algorithm	511
8.8.6 Simulation Results	516
8.8.7 Association for Synchronized Sensors — Summary	521
8.9 MEASUREMENT ASSOCIATION FROM DIRECTION FINDERS	522
8.9.1 The S-D Assignment via Lagrangian Relaxation	522
8.9.2 Example of Localization of Emitters	523
8.9.3 Emitter Localization — Summary	525
8.10 NOTES AND PROBLEMS	526
8.10.1 Bibliographical Notes	526
8.10.2 Problems	526
9 TRACKING WITH IMAGING SENSORS	529
9.1 INTRODUCTION	529
9.1.1 Outline	529
9.1.2 Summary of Objectives	531
9.2 MEASUREMENT EXTRACTION FOR AN EXTENDED TARGET	532
9.2.1 Introduction	532
9.2.2 Modeling of the Image	533
9.2.3 Estimation of the Centroid	535
9.2.4 The Offset Measurement from Image Correlation	537
9.2.5 Application to a Gaussian Plume Target	539
9.2.6 Statistical Properties of Centroid Measurements — Summary	541
9.3 PRECISION TRACKING OF THE IMAGE CENTROID FOR A SINGLE TARGET	542
9.3.1 Introduction	542
9.3.2 Filter Model for White Measurement Noise	543
9.3.3 Filter Model for Autocorrelated Measurement Noise	544
9.3.4 Hot Spot Target Tracking Simulation Results	547
9.3.5 Image Centroid Tracking — Summary	549
9.4 PRECISION TRACKING OF THE IMAGE CENTROIDS FOR TWO TARGETS WITH OVERLAPPING IMAGES	550

CONTENTS

9.4.1	Introduction	550
9.4.2	The Overlapping Images	551
9.4.3	The State Estimation	555
9.4.4	Filter Model for White Measurement Noise	556
9.4.5	Filter Model for Autocorrelated Measurement Noise	559
9.4.6	Targets with Crossing Images — Simulation Results	562
9.4.7	Tracking of Targets with Overlapping Images — Summary	565
9.5	PRECISION TRACKING WITH SEGMENTATION FOR IMAGING SENSORS	566
9.5.1	Introduction	566
9.5.2	Centroid of a Random Cluster	568
9.5.3	Clustering and the Centroid Variance	571
9.5.4	The State Estimation Model	573
9.5.5	Precision Tracking of a Synthetic Image	574
9.5.6	Precision Tracking of a Real Target	578
9.5.7	Small Image Precision Tracking — Summary	582
9.6	NOTES AND PROBLEMS	583
9.6.1	Bibliographical Notes	583
9.6.2	Problems	583
	PROBLEM SOLUTIONS	585
	BIBLIOGRAPHY	593
	INDEX	607

PREFACE

FOREWORD

This text — set of lecture notes — presents material accumulated from a series of short courses held at UCLA Extension since 1985. The prerequisites are a solid knowledge of state estimation, which in turn relies on linear systems, probability theory, and some elements of statistics, summarized in the companion text [BL93].

The main goal of this text is to convey the knowledge necessary for the **design of state estimators that operate in a multitarget-multisensor environment**. This problem is characterized by measurement origin uncertainty in addition to the usual noises in the state and measurement equations. Such a situation occurs in an environment where there is clutter or the false alarm rate is high — typical for the case of **low observable targets** — or due to the presence of several targets in the same neighborhood.

The rationale of this text is the following: It is not rational to have sloppy post-processing information analysis (as is the case in many existing systems) in view of the vast amount of expenditure on sensors and signal processing.

The relevance of the techniques discussed extends beyond tracking to other areas dealing with robotics, computer vision for autonomous navigation and image feature extraction with application to medical diagnosis.

The emphasis is on mapping the physical quantities of the target(s) of interest and of the sensor(s) into the parameters of the mathematical model, namely, the statistical characterization of the uncertainties of this problem. The approach is a balanced combination of mathematics — linear systems, probability theory, statistics and state estimation — in order to understand how a state estimator/data associator should be designed, with the necessary tools from statistics in order to interpret the results. The use of statistical techniques has been somewhat neglected in the engineering literature pertaining to state estimation but it is necessary for the (nontrivial task of) interpretation of stochastic data and to answer the question whether a design can be accepted as “good.”

FOREWORD

The presentation of the material stresses the algorithms, their properties and the understanding of the assumptions behind them. Proofs are given to the extent they are relevant to understanding the results. The underlying philosophy behind the manner in which this text has been written is the KISS principle (Keep It Simple, Stupid!). The amount of mathematics used is as little as possible, but not less.

It is our hope that this text will contribute to the education of practitioners involved in the application of state estimation to real-world tracking problems.

We would like to thank the following colleagues and students for their friendship, inspiration and help with the material that makes up this book: W. D. Blair, H. A. P. Blom, L. Campo, K. C. Chang, E. Daeipour, S. Deb, O. E. Drummond, A. Houles, C. Jaufret, T. Kirubarajan, A. Kumar, D. Lerro, P. B. Luh, E. Oron, K. R. Pattipati, C. Rago, H. M. Sherkude and M. Yeddanapudi.

Also, thanks to the sponsors who have supported the research that lead to this book: J. G. Smith and R. Madan from ONR, N. Glassman from AFOSR, R. Baheti from NSF and L. Lome from BMDO.

A Note on the Printing Style

The format of this book — *textgraph*TM — attempts to accomplish two goals: to serve as a self-contained *concise text*, without excess verbosity, and at the same time to enable the lecturer to use the pages of this text directly as *viewgraphs for lectures*.

In order to make this textbook more reader-friendly (hopefully), all major concepts, terms and acronyms, when they are introduced or used in a particular application, are in *slanted boldface* and indexed. Also, all the index terms appear in the text as slanted boldfaced. The *acronyms* are also indexed.

Italics are used for emphasized ideas/properties.

The main equations of important algorithms are highlighted with boxes.

Chapter 1

INTRODUCTION

1.1 BACKGROUND

1.1.1 Estimation, Decision and Tracking

The process of inferring the value of a quantity of interest from indirect, inaccurate and uncertain observations is called **estimation**. More rigorously, estimation can be viewed as the *process of selection of a point from a continuous space* — the “best estimate.”

Decision can be viewed as the *selection of one out of a set of discrete alternatives* — the “best choice” from a discrete space. However, one can talk about estimation in a discrete-valued case with the possibility of not making a choice but obtaining some conditional probabilities of the various alternatives. This information can be used without making “hard decisions.”

Therefore, estimation and decision can be seen to be overlapping and techniques from both areas are used simultaneously in many practical problems.

Tracking is the estimation of the state of a moving object. This is done using one or more sensors at fixed locations or on moving platforms.

Tracking is wider in scope than estimation: not only does it use all the tools from estimation, but it also requires extensive use of statistical decision theory, especially when some of the practical problems, notably **data association** (“which is my measurement?”) are considered.

Filtering is the estimation of the (current) state of a dynamic system from noisy data — this amounts to “filtering out” the noise.

1.1.1 Estimation, Decision and Tracking

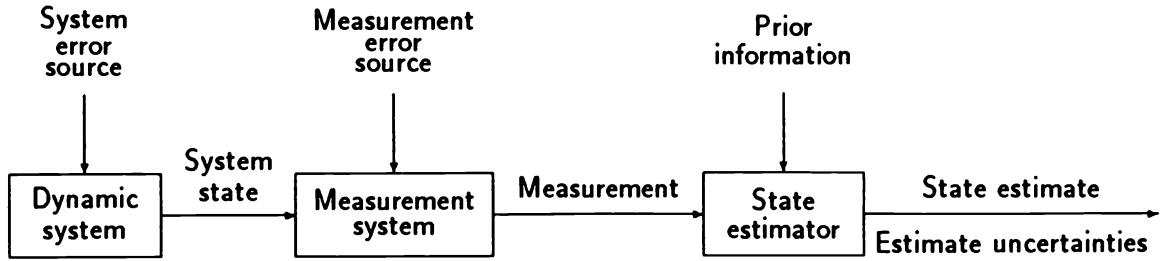


Figure 1.1.1-1: Mathematical view of state estimation.

The variable of interest in an estimation problem can be:

- a parameter — a time-invariant quantity (a scalar, a vector or a matrix)
- the state of a dynamic system (usually a vector).

Figure 1.1.1-1 presents a concise block diagram that illustrates state estimation. The first two blocks are “black boxes” — there is no access to variables inside them. The only variables available to the estimator are the **measurements**, which are affected by the error sources in the form of “noise.”

The estimator uses knowledge about

- the evolution of the variable (the system dynamics)
- the sensor (measurement system)
- the probabilistic characterization of the various random factors (uncertainties) and the prior information.

An **optimal estimator** is a computational algorithm that processes observations (measurements) to yield an estimate of a variable of interest, that minimizes a certain error criterion.

The advantage of an optimal estimator is that it makes best utilization of the data and the knowledge of the system and the disturbances. The disadvantages are that it is possibly sensitive to modeling errors and might be expensive.

1.1.2 Surveillance Systems

1.1.2 Surveillance Systems

The widespread use and increasing sophistication of **surveillance systems**, both military and civilian, has generated a great deal of interest in algorithms capable of tracking *large numbers of targets* using measurement data from *many and possibly diverse sensors*.

The high sensitivity of modern sensors and the need to work in low SNR environments can lead to huge data loads, and the presence of countermeasures can further increase the difficulty of tracking.

Furthermore, the tracking effort for n targets can be *substantially more costly than n times the effort for a single target*. This is due to the fact that establishing the correspondence between targets and observations is not a trivial matter — it can be a complex combinatorial problem.

In terms of Figure 1.1.1-1, the connection between the first two blocks is not perfectly known: there is no perfect certainty about the origin of the observation(s) obtained by the sensor. This is because one deals with **remote sensors** which sense energy emitted from or reflected by an object (or several objects) of interest, but there might be other spurious sources of energy as well.

The advances in hardware and algorithms have increased signal processing capabilities by several orders of magnitude in recent years. This has made the measurement data available for tracking even more numerous and complex, creating a demand for corresponding advances in **information processing** techniques to deal with them — such techniques are the subject of this text.

It was recognized as far back as 1964 [Sit64] that in tracking targets there can be an *additional uncertainty* associated with the measurements in addition to their *inaccuracy*, which is usually modeled by some additive noise. This additional uncertainty is related to the **origin of the measurements**: a measurement that is to be used in the tracking algorithm may not have originated from the target of interest.

This situation can occur in a surveillance system when a radar, sonar or optical sensor is operating in the presence of

- clutter
- countermeasures
- false alarms.

1.1.2 Surveillance Systems

Measurement origin uncertainty can also happen when several targets are in the same neighborhood and, even though one can resolve (separate) the observed detections, *one can not associate them with certainty* to the targets. A similar situation occurs in the track formation problem when there are several targets but their number is unknown and some of the measurements may be spurious.

The application of *standard estimation algorithms*, which would use the measurement nearest in some sense to the predicted measurement — “nearest neighbor approach” — can lead to very poor results in an environment where spurious measurements occur frequently. This is because such an approach does not account for the fact that the measurement used in the algorithm might have originated from a source different from the target of interest.

The Essence of Estimation

The ***raison d'être of estimation*** is the generation of information that has the following properties:

- quality (i.e., accuracy/reliability) higher than the raw measurements
- contains information not directly available in the measurements.

1.2 TRACKING AND DATA ASSOCIATION

1.2.1 Tracking

Tracking is the *processing of measurements* obtained from a target in order to maintain an *estimate of its current state*, which typically consist of:

- Kinematic components — position, velocity, acceleration, turn rate, etc.
- Feature components — radiated signal strength, spectral characteristics, radar cross-section, target classification, etc.
- Constant or slowly varying parameters — aerodynamic parameters, etc.

Measurements are noise-corrupted observations related to the state of a target, such as

- Direct estimate of position (usually range, azimuth and elevation)
- Range and azimuth (bearing) from the sensor
- Bearing only from the sensor
- Range rate
- Time difference of arrival (TDOA) of a signal between two sensors
- Frequency of narrow-band signal emitted by the target (Doppler shifted)
- Frequency difference of arrival (Doppler difference) at two sensors.

The measurements of interest are not raw data points but usually the outputs of complex signal processing and detection subsystems as shown in Figure 1.2.1-1.

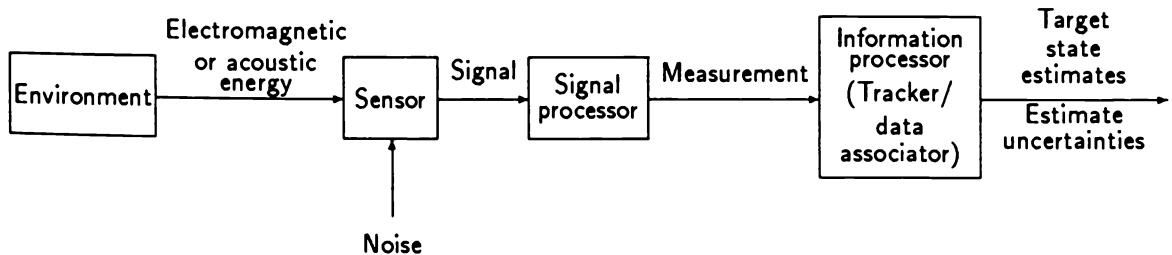


Figure 1.2.1-1: Outline of the components of a tracking system.

Active sensors emit energy into the environment and search for reflected energy, while **passive sensors** search for energy emitted from the object(s) of interest.

1.2.2 Data Association

A **track** is a state trajectory estimated from a set of measurements — the **data** — that have been *associated with the same target*.

The crux of the multitarget problem is to carry out this **association process**, also called by abuse of language “**data correlation process**” for measurements whose origin is uncertain due to

- random false alarms in the detection process
- clutter due to spurious reflectors or radiators near the target of interest
- interfering targets
- decoys and countermeasures.

Furthermore, the probability of obtaining a measurement from a target — the **target detection probability** — is usually less than unity.

Data association problems may be categorized according to what is associated with what:

- **measurement to measurement association** — track formation
- **measurement to track association** — track maintenance or updating
- **track to track association** — track fusion (for multisensor systems).

There are two fundamentally different approaches one can take in associating the data:

- **Non-Bayesian data association** — this approach carries out a decision procedure, using statistical tools (maximum likelihood or hypothesis testing) and, after arriving at an association decision, the fact that it is not necessarily correct is ignored in the sequel.
- **Probabilistic (Bayesian) data association** — this approach evaluates probabilities of associations and uses them throughout the estimation process.

Both of these approaches rely on certain models of the targets and the sensors from which the measurements are obtained.

Significant difficulties are encountered in the modeling of the behavior of the targets, which can maneuver, i.e., exhibit different **behavior modes**. This compounds the already complex problem of associating measurements under uncertainty.

1.2.3 Some Terminology

1.2.3 Some Terminology

The following terminology pertains to tracking and data association and is useful to define at this stage (for more details see [DF93])

Sensor: a device that observes the (remote) environment by reception of some signals (energy)

Frame or scan: “snapshot” of region of the environment obtained by the sensor at a point in time, called the sampling time (actually this is an integral over a short period of time)

Signal processing: processing of the sensor data to provide measurements (usually from a single sampling time)

Signal detection: thresholding of sensor data (usually from a single sampling time) for further processing

Measurement formation (extraction): the final stage of signal processing that generates a measurement

Measurement (observation, hit, return, report, threshold exceedance): the estimated parameter of a detected signal

Time stamp: the time to which a detection/measurement pertains

In a **scanning sensor** the time to which a detection pertains is somewhere within the scanning period

In a **staring sensor** the time for all the detections from a frame is the same

Registration: alignment of two or more sensors or alignment of moving sensor data from successive sampling times so that their data can be combined

Track formation (or track assembly, target acquisition, measurement to measurement association, scan to scan association): detection of a target (processing of measurements from a number of sampling times to determine the presence of a target) and initiation of its track (determination of the initial estimate of its state)

Tracking filter: state estimator of a target

Data association: process of establishing which measurement (or weighted combination of measurements) is to be used in a state estimator

Track continuation (maintenance or updating): association and incorporation of measurements from a sampling time into a tracking filter

Cluster tracking: tracking of a set of nearby targets as a group rather than individually.

1.3 SCOPE OF THE TEXT

1.3.1 Objectives

The objectives of this text are to present the tools for the *design* of state-of-the-art algorithms for *measurement association and target tracking*.

Various algorithms for *measurement association and target tracking* are presented that rely on

- techniques of system theory that deal with the state space representation of linear and nonlinear dynamic systems
- estimation techniques for random processes
- statistical decision theory (hypothesis testing)

and form the basis for the design of **automated decision systems for data association and tracking** (also called **correlator-trackers**) for realistic situations with multiple targets and multiple sensors.

The fundamentals of *state estimation theory* — the background material for this text can be found in the companion text [BL93].

The modeling will account for

- false alarms
- clutter
- target detection probability
- finite resolution of sensors
- target maneuvers
- interference from other targets.

The material to be discussed deals mainly with the information processing — last block of Figure 1.2.1-1 — which consists of

- data association and
- state estimation

using the output of the signal processor.

A *joint consideration* of the signal processor and the information processor for an overall system optimization is also presented.

Special attention is given to the *mathematical assumptions* underlying the various algorithms and their *applicability to practical situations*.

1.3.2 Overview and Prerequisites

The remaining part of this chapter presents

- A guided tour of the tracking problems and available algorithms
- Overview of the basic tracking algorithms (Kalman filter, alpha-beta-gamma filter, Extended Kalman filter, Multiple Model approach)
- Coordinate conversion
- An application to Air Traffic Control
- Adaptive beam pointing of a phased array radar.

The list below presents a brief description of the contents of each of the following chapters as well as their prerequisites, indicated in parentheses.

Multiple Sensors, Clutter and Multiple Targets — Chapter 2 (Prerequisite: [BL93] Chapter 5)

- Tracking with multiple sensors
- Measurement validation
- Clutter modeling
- Track initiation in clutter: performance evaluation and system design

Algorithms for Tracking a Single Target in Clutter — Chapter 3 (Prerequisite: Chapter 2 and [BL93] Chapter 5)

- The use of the standard filter
- Track splitting
- The Probabilistic Data Association Filter (PDAF)
- The optimal Bayesian approach
- Track formation with bearing and frequency measurements in the presence of false detections

Single Maneuvering Target in Clutter — Chapter 4 (Prerequisite: Chapter 3 and [BL93] Chapter 11)

- The Multiple Model PDAF
- Tracking with target features
- Multisensor tracking of a maneuvering target in clutter
- Track initiation and termination with multiple models
- IMMPDAF with target amplitude feature

1.3.2 Overview and Prerequisites

Tracking Performance Evaluation and Detection Optimization — Chapter 5 (Prerequisite: Chapter 4)

- Tracking accuracy in clutter
- Tracker Operating Characteristic

Multiple Targets in Clutter: Bayesian Approaches — Chapter 6 (Prerequisite: Chapter 3)

- The Joint Probabilistic Data Association Filter (JPDAF)
- The Multiple Hypothesis Tracker (MHT)
- Multiple target tracking with possibly unresolved measurements

Multiple Targets in Clutter: Non-Bayesian Approaches — Chapter 7 (Prerequisite: Chapter 3)

- Joint likelihood function method for track formation
- Logic-based multitarget track initiation
- Evaluation of a composite logic for track initiation: a case study

Multisensor Tracking and Data Fusion — Chapter 8 (Prerequisite: Chapter 3)

- Multisensor tracking architectures
- Testing for common origin and fusion of tracks
- Distributed and hierarchical multisite tracking
- A Multisensor-Multitarget data association algorithm for heterogeneous sensors

Tracking with Imaging Sensors — Chapter 9 (Prerequisite: Chapter 6)

- Measurement extraction for an extended target and tracking of the image centroid
- Precision tracking of the image centroids for two targets with overlapping images
- Precision tracking with segmentation for imaging sensors

1.4 OVERVIEW OF TRACKING PROBLEMS AND ALGORITHMS — A GUIDED TOUR ¹

The following three stages can be distinguished in target tracking:

- ***initial detection***
- ***target acquisition — track formation or initiation***
- ***precision tracking — track maintenance or continuation.***

1.4.1 The Types of Problems

Classification criteria

- track stage
 - track formation (initiation)
 - track maintenance (continuation)
- number of sensors
 - single sensor
 - multiple sensors
- sensor detection characteristics
 - target detection probability (P_D)
 - false alarm probability (P_{FA})
- target behavior
 - stationary (nonmaneuvering)
 - nonstationary (maneuvering)
- number of targets
 - single target
 - multiple targets
- target size
 - sensor resolution cell
 - extended target
 - image.

In the sequel the various problems are briefly specified and references are given.

¹Guide to the Perplexed.

Shortened Notations

The following shortened notations for references appearing often are used:

([BF88]:X.Y) denotes Section X.Y of Bar-Shalom and Fortmann, *Tracking and Data Association*, Academic Press, 1988

([Bar90]:X.Y) denotes Section X.Y of Bar-Shalom (ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, 1990

([Bar92]:X.Y) denotes Section X.Y of Bar-Shalom (ed.), *Multitarget-Multisensor Tracking: Applications and Advances*, Artech House, 1992

([BL93]:X.Y) denotes Section X.Y of Bar-Shalom and Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, 1993

(X.Y) denotes Section X.Y of the present text

[Mdat] denotes the software MULTIDATTM

[Pdat] denotes the software PASSDATTM

[Bdat] denotes the software BEARDATTM

[Idat] denotes the software IMDATTM

[Fdat] denotes the software FUSEDATTM

[Dest] denotes DynaEstTM, the companion software of the text [BL93].

The *DAT software packages listed above are available from the first author of this text.

1.4.2 Track Formation with a Single Sensor

1.4.2 Track Formation with a Single Sensor

Assuming **single frame signal processing** that yields detections and associated measurements, one has the following classes of problems:

Track formation with ideal sensor ($P_D = 1, P_{FA} = 0$):

- LS fit to motion model (usually a polynomial in time) ([BF88]:2.2;2.6) ([BL93]:3.5;5.5).

Track formation with realistic sensor ($P_D < 1, P_{FA} > 0$):

- Gating and data association ([Bla86]:4) ([BF88]:5.1) (8.3)
- Track life stages ([Bla86]:6)
- Cascaded logic ($2/2 \& M/N$): Evaluation for single target with white noise acceleration motion model of the true track acceptance probability with $P_D < 1$ and false track acceptance probability with $P_{FA} > 0$ (8.6) [BCS89] [Mdat]
- Incorporation of goodness of fit into logic [BL91]
- Composite logic ($M_2(M_1/N_1)/N_2$): Evaluation, for maximum maneuver capability, of the true track acceptance probability with $P_D < 1$ and $P_{FA} > 0$, false track acceptance probability and required target separation for non-interference (7.4) [BCL90]
- Application of logic to a large-scale problem ([BF88]:9.5) (7.3)
- Assignment with distance function criterion ([Bla86]:4.3;14.2)
- Assignment approach with maximum likelihood criterion ([BF88]:9.2) (7.2) [DPB92b, DPB92a] [PDBW92]
- IMMPDAF (Interacting Multiple Model Probabilistic Data Association Filter): use of two models — “no target” and “true target” — to obtain the “true target probability” of a tentative track ([Bar90]:2) (4.4) [Mdat]
- IMMPDAFAI — incorporation of amplitude information into the PDAF [LB91a] [LB93b] (4.6)
- MHT (Multiple Hypothesis Tracking) — evaluation of all the possible measurement sequences ([Bla86]:10.2) ([BF88]:9.4) ([Bar90]:3;8) (6.3)
- Group tracking for unresolved or marginally resolved targets ([Bla86]:11.2;11.3)
- Use of bearing and frequency measurements for target track detection and estimation [JB90] (3.7) [Bdat].

1.4.2 Track Formation with a Single Sensor

The above techniques all assume that **single frame (scan) thresholding** is used to obtain detections and the corresponding measurements at the output of the signal processor.

The following alternative approach, which retains single frame intensities in each resolution cell (pixel), can be taken:

- **Track before detect or Multiframe thresholding or Multiframe signal processing:** non-coherent energy integration is carried out along possible trajectories over several frames and an efficient search is carried out (via dynamic programming) to detect the target trajectory(ies) ([Bar90]:4).

1.4.3 Track Maintenance with a Single Sensor without Measurement Origin Uncertainty

1.4.3 Track Maintenance with a Single Sensor without Measurement Origin Uncertainty

The general goal of track maintenance is to *improve over the sensor accuracy* (“sophisticated software can save on sensor hardware!”) and obtain estimates of variables not measured by the sensor.

Nonmaneuvering Targets

- KF (Kalman filter²): linear system and linear measurements with the only uncertainty being white (Gaussian) noise with known moments ([BF88]:2.4) ([BL93]:5.2) [Dest]
- α - β - (γ) filter³: fixed gain (steady state) KF for polynomial motion models with assumed zero-mean acceleration (jerk) white noise ([BF88]:2.7) ([BL93]:6.5)
- EKF (Extended KF): system and/or measurements are nonlinear; requires linearization which can cause problems (1st order, 2nd order, iterated) ([BF88]:3.3-3.5) ([BL93]: 10.3–10.5).

Performance Prediction for Nonmaneuvering Targets

- Using the KF Riccati equation ([BF88]:2.4) ([BL93]:5.2).

Maneuvering Targets (variable behavior)

- α - β - (γ) filter with large process noise to “cover maneuvers” ([BF88]:2.7) ([BL93]:6.5)
- α - β - (γ) filter with heuristic (ad-hoc) gain switching ([BF88]:4.8) ([BL93]:11.2;11.5)
- KF/EKF with variable process noise ([BF88]:4.2) ([BL93]:11.2)
- VSD (Variable State Dimension) filter: switching from low order quiescent model to higher order maneuvering model ([BF88]:4.7;4.8) ([BL93]:11.4;11.5)
- IE (Input Estimation): the unknown input (maneuver) is estimated under the assumption that it is constant over a window and the state estimate is corrected with it ([BF88]:4.6;4.8) ([BL93]:11.3;11.5)

²The workhorse of tracking.

³The mule of tracking. It has been suggested at one time to outlaw this filter, which dates back to the Allies.

1.4.3 Track Maintenance with a Single Sensor without Measurement Origin Uncertainty

- MM (Multiple Model) approaches
 - static (that is, non-switching): the system obeys one of a finite number of models ([BF88]:4.3) ([BL93]:11.6)
 - optimal dynamic (has exponentially growing requirements) ([BL93]:11.6)
 - suboptimal dynamic (with fixed requirements; assumes Markov switching between the models): GPB (Generalized Pseudo-Bayesian), IMM (Interacting Multiple Mode ([BL93]:11.6) [Dest]; Application of IMM to Air Traffic Control Tracking (1.7).

Performance Prediction for Maneuvering Targets

- Monte Carlo simulations ([BF88]:4.8) ([BL93]:1.5.3;11.5)
- Non-simulation techniques: Hybrid technique for the prediction of the performance of an IMM estimator [LB93d].

Coordinate Selection

- Use of debiased coordinate conversion (1.6).

Sensor Scheduling

- Phased array radar scheduling for maneuvering targets (1.8).

Tracking in the presence of multipath propagation

- Use of the IMM to alleviate the multipath propagation effects in radar tracking of sea skimmers [BKBG94, BGBD94].

1.4.4 Track Maintenance with a Single Sensor with Measurement Origin Uncertainty

1.4.4 Track Maintenance with a Single Sensor with Measurement Origin Uncertainty

Single Nonmaneuvering Target in Clutter (Low Observable Target)

- Nearest (or strongest) Neighbor Standard Kalman filter: the measurement nearest to the predicted location or the strongest detection in the gate are used in a standard KF (as if it were the correct one) ([BF88]:6.2) (3.2) [LB94d] [Mdat]
- Track splitting and monitoring with likelihood function (goodness of fit) ([BF88]:6.3) (3.3)
- PDAF (Probabilistic Data Association filter): use of all the measurements in the current time validation region (gate) weighted by the computed association probability — **single frame association** ([BF88]:6.4) (3.4) [Mdat]
- Optimal Bayesian: track splitting with probabilistic evaluation of the measurement sequences — **multiple frame association** ([BF88]:6.5) (3.5).

Performance Prediction for a Single Target in Clutter

- association performance ([Bla86]:7.3) ([Bar92]:7)
- adaptive thresholding ([Bla86]:5.3) ([BF88]:8) [LB94b]
- track life prediction [LB91b].

Single Maneuvering Target in Clutter

- PDAF with target feature (4.4)
- MMPDAF: static Multiple Model PDAF with ad-hoc modifications for model switching ([BF88]:7.4) (4.3)
- IMMPDAF: IMM with Markov model switching combined with PDAF (4.6) [Mdat]
- IMMPDAFAI: IMMPDAF with amplitude information [LB93b]
- JumpDif: IMMPDAF with second order Markov jumps (switchings) between different models (with impulsive as well as diffusion type — continuous noise — driving functions) ([Bar92]:2).

1.4.4 Track Maintenance with a Single Sensor with Measurement Origin Uncertainty

Extended Target (*hot spot*, up to about 10 pixels)

- KF using centroid and offset (correlation of adjacent frames) measurements with statistics based on video noise (9.3) ([Bar92]:5).

Small Image (10–100 pixels)

- PDAF using centroid of segmented image (binarized by intensity bandpassing and clustered) (9.5) [Idat].

Navigation with Landmarks

- PDAF with landmark observations with identity (Bayesian or Dempster-Shafer model) ([Bar92]:9) [DB93].

Multiple Targets in Clutter

- Assignment approach with maximum likelihood criterion ([Bla86]:9;14) ([BF88]:9.2) (7.2)
- JPDAF (Joint PDAF): joint probabilistic association of the measurements to several targets in the same neighborhood followed by decoupled filtering ([BF88]:9.3) (6.2)
- JPDACF: same as above except the filtering is coupled (6.2); For two crossing targets: [Mdat]
- NNJPDAF (Nearest Neighbor JPDAF): only the most likely (nearest) measurement is used for updating each target state ([Bar90]:1)
- JPDAM (JPDAF with possibly Merged measurements): with separate model for the statistics of an unresolved measurement originating from two neighboring targets (6.4)
- JPDAMCF (JPDAM with Coupled filtering) ([Bar92]:5) (9.4)
- MHT (Multiple Hypothesis Tracking): track maintenance and initiation via a simultaneous consideration of all the hypotheses ([Bla86]:10.2) ([BF88]:9.4) ([Bar90]:3;8) (6.3)
- MHT with multiple models for maneuvers ([Bar90]:3).

Performance Evaluation for Multiple Targets in Clutter

- Fundamental limits in Multiple Target Tracking ([Bar92]:6).

1.4.4 Track Maintenance with a Single Sensor with Measurement Origin Uncertainty

Splitting Target in Clutter

- IMMJPDA (IMM combined with JPDAF for a Splitting target): “warm start” for a new target spawned from a platform ([Bar92]:4) [Mdat].

Crossing Extended Targets

- JPDAAMCF: as above for target centroids (which merge when the targets cross in the sensor field of view) and offsets (ambiguous in this situation) ([Bar92]:5) (9.3).

1.4.5 Track Formation with Multiple Sensors

1.4.5 Track Formation with Multiple Sensors

Alignment for Multiple Sensors

In order to effectively use ***multiple sensors***, they have to be aligned (“registered”):

- Multiradar registration: A least squares based off-line technique for estimating the alignment errors ([Bar90]:5).

Processing Configurations for Multiple Sensors

In the case of multiple sensors there are several possible configurations for data association and fusion (8.2) [DBP90]:

- Configuration I: single sensor situation (baseline)
- Configuration II: single sensor level track formation followed by track-to-track association and fusion ([BF88]:10) ([Bar90]:6)
- Configuration III: multisensor association of measurements from synchronized frames (full position estimates — “supermeasurements” — from sensor LOS (line of sight) data) followed by association across time (frames) of the supermeasurements
 - Combination of LOS data (triangulation) from 2 sensors ([Bar90]:6)
 - Association of LOS data across 3 sensors and “deghosting” via combinatorial optimization ([Bar90]:7) [Pdat] (8.7)
- Configuration IV: centralized (no synchronization among sensors required) — direct association of the sensor measurements across sensors and time ([Bar90]:6) [DPB92b] (8.7.7).

Target Track Detection Based on Time Difference of Arrival.

- For low SNR target in the near field of two sensors [SB90]

1.4.6 Track Maintenance with Multiple Sensors

1.4.6 Track Maintenance with Multiple Sensors

Processing Configurations for Multiple Sensor Data Fusion

As in track formation, the following configurations can be used:

- Configuration I: single sensor situation
- Configuration II: single sensor level track maintenance followed by track-to-track association and fusion ([Bla86]:13.3) ([BF88]:10) (8.2;8.3)
- Configuration III: multisensor association of measurements from synchronized frames to obtain full position estimates — “supermeasurements” — from sensor LOS data ([Bar90]:7) (8.8;8.9) followed by association across time (frames) of the supermeasurements
- Configuration IV: centralized (no sensor synchronization needed) — direct association of the sensor measurements across sensors and time followed by centralized filtering
 - Serial vs. parallel target state updating in the centralized filter with the multisensor data (2.2)
 - IMM/MSPDAF: Multisensor IMMPDAF (4.5)
 - A centralized/distributed architecture for multisensor track formation and maintenance (8.6) [Fdat]
 - Assignment algorithm for multisensor tracking in Air Traffic Control (8.7)
 - A multiassignment algorithm for “spawning” targets [Tsa91].

Alignment of Sensors in Real Time

The alignment of sensors can change in real time so it might be necessary to re-register them while tracking targets:

- Real-time estimation of misalignment in a multiradar system ([Bar92]:2).

Target Attribute Data Fusion Across Sensors

This can be done with classical probabilistic techniques or with “evidential reasoning”:

- With Bayesian approach (8.5)
- With Dempster-Shafer approach ([Bar90]:6) ([Bar92]:11).

1.4.6 Track Maintenance with Multiple Sensors

Resource Allocation in a Multisensor Environment

- Real-time sensor management for tracking and situation assessment ([Bar92]:10;11).

Predetection Fusion

- Fusion of the signals prior to thresholding from several sensors, followed by detection (SPIE Proc. #1954, #1955, April 1993, #2232, #2235, April 1994).

1.5 OVERVIEW OF THE BASIC TRACKING ALGORITHMS

This section presents brief descriptions of the main **tracking algorithms** discussed in detail in [BL93]: the Kalman filter, α - β filter, Extended Kalman filter and Multiple Model estimators.

1.5.1 Kalman Filter Overview

Modeling Assumptions

The fundamental assumptions of the **Kalman filter (KF)** are:

- Evolution of the system's state — according to a *known linear plant equation* (dynamics) driven by
 - A known input
 - An additive process noise — zero-mean white (uncorrelated) process with known covariance $Q(k)$
- Measurements — a *known linear* function of the state with
 - An additive measurement noise — zero-mean white with known covariance $R(k)$
- Initial state — unknown, assumed to be a random variable with a known mean (initial estimate) and covariance (the initial uncertainty)
- Initial error and noises — mutually uncorrelated.

Algorithm Flowchart

At every stage k the entire past is summarized by the state estimate $\hat{x}(k|k)$ and the associated covariance $P(k|k)$.

The left column of Figure 1.5.1-1 shows the true system's evolution from time k to time $k+1$ with the input $u(k)$ and the process noise $v(k)$; the measurement follows from the new state and the noise $w(k+1)$.

The state estimation cycle (middle column of the figure) consists of

- (i) state and measurement ***prediction*** (also called “time update”),
- (ii) state ***update*** (also called “measurement update”).

The known input (control or platform motion or sensor pointing) is used by the state estimator to obtain the state prediction for the next time. The state update requires the filter gain, obtained in the course of the covariance calculations in the right column.

1.5.1 Kalman Filter Overview

The Workhorse of Tracking — The Kalman Filter (KF)

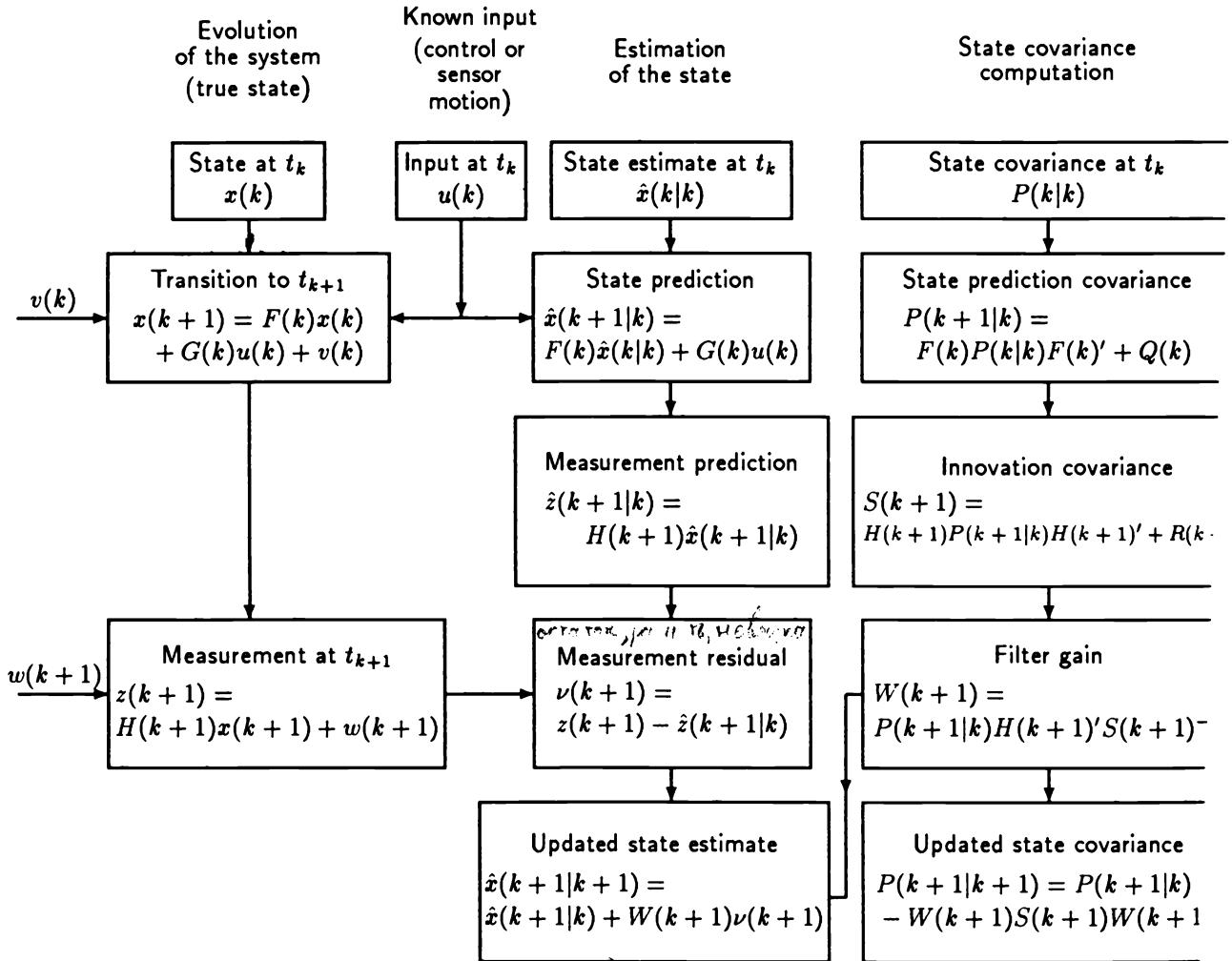


Figure 1.5.1-1: One cycle in the state estimation of a linear system.

1.5.1 Kalman Filter Overview

Remarks

Under the *Gaussian assumption* for the initial state error and all the noises entering into the system, the KF is the **MMSE state estimator**.

If the above random variables are *not Gaussian*, and one has only their first two moments, then, the Kalman Filter is the **best linear MMSE estimator** — best within the class of linear estimators.

The covariance calculations are *independent* of the state (and input — assumed to be known) and can, therefore, be performed *off-line*.

Practical issues in tracking not answered by the KF formulation

- nonlinear motion models (use linearization — extended Kalman filter (EKF))
- nonlinear measurement equations
 - use linearization — EKF
 - use **debiased consistent** transformation (polar to Cartesian) and then linear filter (see 1.6)
- unknown inputs into the dynamic equation and/or mode changes (different motion models — e.g., uniform motion vs. straight line acceleration vs. coordinated turn)
- auto- or cross-correlated noises
- uncertain origin measurements — data association uncertainty
- sensor resolution and multipath propagation
- how many targets are there?

1.5.2 Alpha-Beta-Gamma Filter Overview

1.5.2 Alpha-Beta-Gamma Filter Overview

Modeling Assumptions

For discrete-time linear kinematic (“polynomial”) models with

- zero-mean white process noise that models
 - acceleration (2nd order model)
 - acceleration increments (3rd order model — Wiener process acceleration)
- noisy position measurements

one has explicit expressions of the *steady state KF gain* and covariance.

These filters, called **alpha-beta** and **alpha-beta-gamma**, respectively, are the simplest possible: they use **precomputed fixed gains**.

Classes of models

- discretized continuous-time models based on *continuous-time zero-mean white noise*
- direct discrete-time models based on *piecewise constant zero-mean white noise*, i.e., a *zero-mean white sequence*.

These two classes have a different dependence on the sampling period of the effect of the noises on the motion.

None of these noise assumptions model exactly target maneuvers, which are neither zero-mean, nor white — they are not even random — but the state models require the specification of some randomness.

The filter gains

The gains of these filters depend *only* on the **target maneuvering index** — the ratio between the RMS values of the following two uncertainties:

- the position displacement over one sampling period due to the process noise, and
- the (position) measurement noise.

Advantages

- in implementations where the computational and memory requirements are a major consideration
- yield quick (but possibly dirty) evaluations of achievable tracking performance as measured by the steady-state error variances.

1.5.2 Alpha-Beta-Gamma Filter Overview

Disadvantages

- very limited ability in clutter
- when used independently for each coordinate, one can encounter instability under certain extreme circumstances due to the decoupling.

Selection of filter design parameters

- measurement noise standard deviation — the accuracy of the sensor
- process noise standard deviation:
 - for the white noise acceleration model (2nd order): 50%–100% of the maximum target acceleration
 - for the white noise acceleration increment model (3rd order): 50%–100% of the maximum target acceleration increment over the sampling interval.

Remark

These choices are conservative, i.e., when the target is in uniform motion, such a filter cannot take advantage of it and *will not yield enough noise reduction*.

Other uses of linear kinematic models

The linear kinematic models can be used for full-fledged Kalman filters (with real-time covariance calculations and time-varying gains) as **modules in Multiple Model tracking algorithms** as follows:

- 2nd order model
 - with low level process noise models **nearly constant velocity motion**
 - with high level process noise can model **not very well maneuvers**
- 3rd order model
 - with low level process noise models **nearly constant acceleration motion**
 - with high level process noise can model **acceleration onset/termination**.

The selection of process noise variances should be done according to the same rules as indicated above.

1.5.3 Extended Kalman Filter Overview

1.5.3 Extended Kalman Filter Overview

The **Extended Kalman Filter (EKF)** is a suboptimal⁴ state estimation algorithm for nonlinear systems.

Modeling Assumptions:

Same as for the KF except

- the dynamic and/or the measurement equation are nonlinear functions; the noises are still assumed to enter additively.

Algorithm versions

- Based on first order Taylor expansion of the nonlinear functions (in the prediction stage)
 - EKF
- Based on second order Taylor expansion of the nonlinear functions — 2nd order EKF.

The Algorithm

Figure 1.5.3-1 presents the flowchart of the EKF (first order version).

The main difference from the KF is the evaluation of the Jacobians of the state transition and the measurement equations. Due to this, the covariance computations are not decoupled anymore from the state estimate calculations and cannot be done, in general, off-line.

Linearization (evaluation of the Jacobians) options

- at the latest state estimate, or
- along a nominal trajectory — a deterministic precomputed trajectory based on a certain scenario — this allows off-line computation of the gain and covariance sequence.

⁴This is an elegant euphemism for “nonoptimal”.

1.5.3 Extended Kalman Filter (EKF) Overview

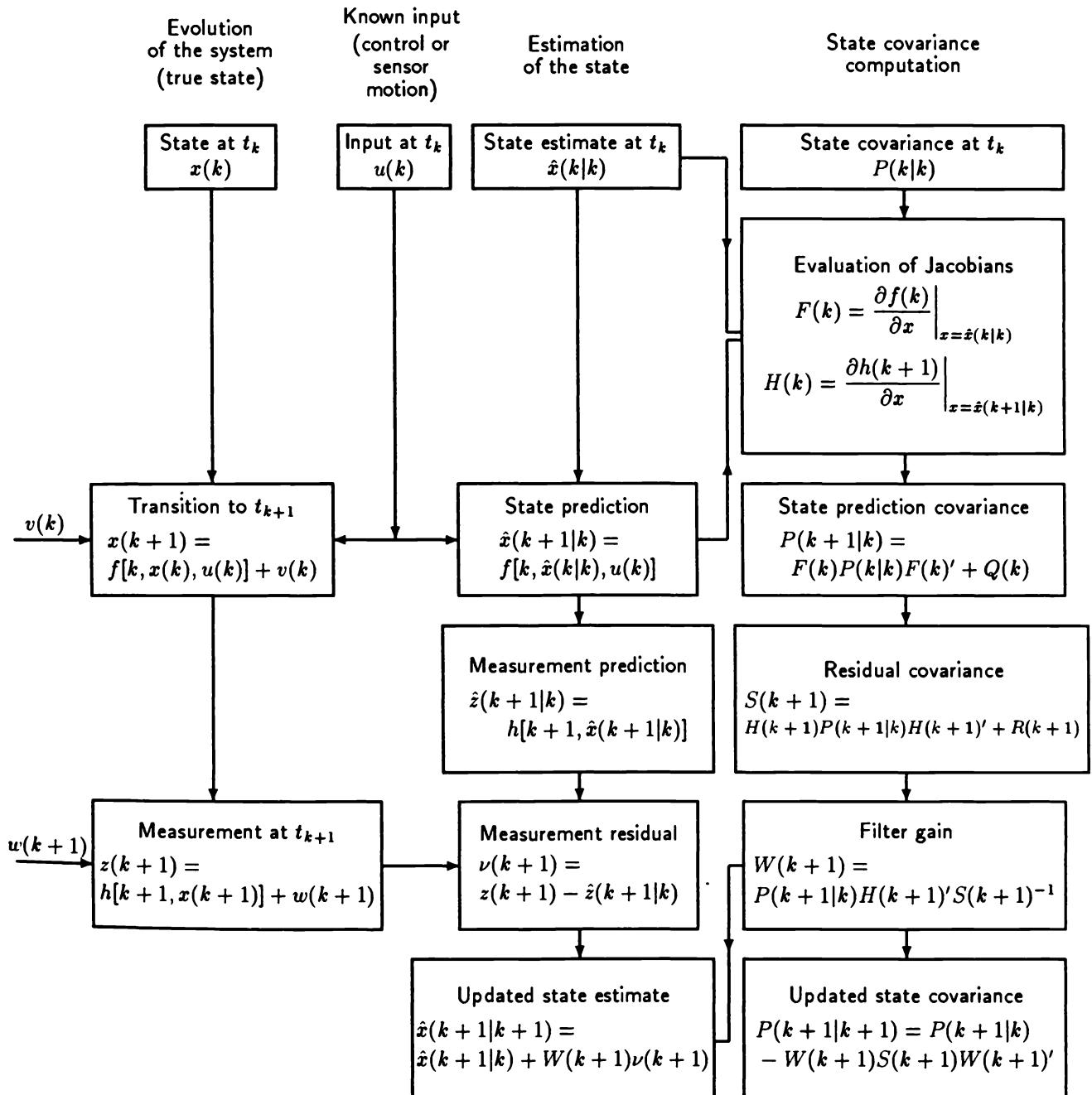


Figure 1.5.3-1: Flowchart of the EKF (one cycle, 1st order version).

1.5.3 Extended Kalman Filter (EKF) Overview

A Cautionary Note about the EKF

The use of the series expansion in the state prediction and/or in the measurement prediction has the potential of introducing unmodeled errors that violate some basic assumptions about the prediction errors:

- (a) they are zero-mean (unbiased), and
- (b) with covariances equal to the computed ones by the algorithm.

In general, a nonlinear transformation will

- introduce a *bias*, and
- the *covariance calculation based on a series expansion is not always accurate*

These expansions rely on Jacobians/Hessians evaluated at the estimated state (rather than the exact state) and neglect the higher order terms.

The EKF is very sensitive to the accuracy of the initial conditions.

Compensation techniques

- Use of second order terms (2nd order EKF) — however, there is no guarantee that even the second order terms can adequately compensate for such errors
- Use of “artificial noise” — increase Q and/or R , or directly increase the filter-calculated covariance, to compensate for unmodeled errors
- Use of numerical integration to predict the state instead of the series expansion
- Use of the “Iterated EKF” which performs a maximum likelihood estimation at the update stage (this amounts to a relinearization).

1.5.3 Extended Kalman Filter Overview

Remarks

Only if the filter's actual mean-square errors are **consistent** with the filter-calculated error variances, then

- the estimation can be considered practically optimal
- its results can be trusted.

In practice, if

- the initial errors and
- the noises

are not too large, then the EKF performs well.

The actual limits of successful use of the linearization techniques implicit in the EKF can be obtained only via extensive Monte Carlo simulations for consistency verification via statistical techniques.

An alternative to the EKF

If

- the motion equations are linear in Cartesian coordinates
- the measurements are in polar coordinates (range and azimuth)

then one can obtain a **polar to Cartesian conversion** that is

- **debiased**
- has **consistent covariance**.

This allows the use of the KF *without the linearization problems inherent in the EKF* (see Section 1.6). The validity of the debiasing is up to a bearing measurement standard deviation (RMS error) $\sigma_\theta = 10^\circ$ and it is beneficial down to $\sigma_\theta = 0.5^\circ$.

1.5.4 Multiple Model Approach Overview

1.5.4 Multiple Model Approach Overview

The Multiple Model approach provides an extremely versatile tool for adaptive state estimation in systems whose behavior pattern changes with time.

General Assumptions

- A **Multiple Model (MM) system** obeys one of a finite number r of models. Such systems are called **hybrid** — they have both *continuous* (noise) uncertainties as well as *discrete* (“model” or “mode”) uncertainties.
- A Bayesian framework is used: starting with prior probabilities of each mode being in effect (i.e., the corresponding model is correct), the posterior mode probabilities are obtained.
- Mode variation in time:
 - fixed (non-switching)
 - switching — the only realistic one.

Algorithms

- static MM algorithm (for non-switching modes): **complexity order** $O(r)$; heuristically adapted to switching situation (obsolete)
- dynamic MM algorithms (for modes switching according to a Markov chain)
 - **optimal MM estimator**: carries all the mode sequence hypotheses; complexity *increasing in time* $O(r^k)$ — *not practical*
 - **Generalized Pseudo-Bayesian estimator** of order n (**GPBn**): keeps hypotheses to depth n by merging at the end of each cycle; complexity of order $O(r^n)$
 - **Interacting Multiple Model estimator (IMM)**: mixes hypotheses with depth 1 at the start of each cycle; complexity $O(r)$ — as GPB1; its performance is almost the same as that of GPB2 — it is the *most cost-effective*.

1.5.4 Multiple Model Approach Overview

Modeling Assumptions

Base state model:

$$x(k) = F[M(k)]x(k-1) + v[k-1, M(k)] \quad (1.5.4-1)$$

$$z(k) = H[M(k)]x(k) + w[k, M(k)] \quad (1.5.4-2)$$

where $M(k)$ denotes the mode “at time k ” — in effect *during the sampling period ending at k* .

Mode (“modal state”) — among the possible r modes:

$$M(k) \in \{M_j\}_{j=1}^r \quad (1.5.4-3)$$

The structure of the system and/or the statistics of the noises can differ from mode to mode:

$$F[M_j] = F_j \quad (1.5.4-4)$$

$$v(k-1, M_j) \sim \mathcal{N}(u_j, Q_j) \quad (1.5.4-5)$$

Mode jump process: Markov chain with known transition probabilities

$$P\{M(k) = M_j | M(k-1) = M_i\} = p_{ij} \quad (1.5.4-6)$$

The IMM Estimation Algorithm (Figure 1.5.4-1)

- **Interaction:** Mixing of the previous cycle mode-conditioned state estimates and covariance, using the mixing probabilities, to initialize the current cycle of each mode-conditioned filter
- **Mode-conditioned filtering:** Calculation of the state estimates and covariances conditioned on a mode being in effect, as well as the mode likelihood function (r parallel filters)
- **Probability evaluation:** Computation of the mixing and the updated mode probabilities
- **Overall state estimate and covariance** (for output only): Combination of the latest mode-conditioned state estimates and covariances.

The IMM estimation algorithm has a *modular structure*.

1.5.4 Multiple Model Approach Overview

The steps of the IMM estimation algorithm

- Interaction ($j = 1, \dots, r$):

Initial estimate and covariance for filter j :

$$\hat{x}^{0j}(k-1|k-1) = \sum_{i=1}^r \hat{x}^i(k-1|k-1) \mu_{i|j}(k-1|k-1) \quad (1.5.4-7)$$

$$\begin{aligned} P^{0j}(k-1|k-1) &= \sum_{i=1}^r \mu_{i|j}(k-1|k-1) \{P^i(k-1|k-1) \\ &\quad + [\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)][\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)]'\} \end{aligned} \quad (1.5.4-8)$$

- Mode-conditioned filtering ($j = 1, \dots, r$):

The KF matched to $M_j(k)$ (filter j) uses $z(k)$ to yield $\hat{x}^j(k|k)$ and $P^j(k|k)$.

Likelihood function corresponding to filter j

$$\Lambda_j(k) = \mathcal{N}[z(k); \hat{z}^j(k|k-1), S_j(k)] \quad (1.5.4-9)$$

- Probability evaluations:

Mixing probabilities ($i, j = 1, \dots, r$):

$$\mu_{i|j}(k-1|k-1) = \frac{1}{\bar{c}_j} p_{ij} \mu_i(k-1) \quad (1.5.4-10)$$

$$\bar{c}_j \triangleq \sum_{i=1}^r p_{ij} \mu_i(k-1) \quad (1.5.4-11)$$

Update of the mode probabilities ($j = 1, \dots, r$):

$$\mu_j(k) = \frac{1}{c} \Lambda_j(k) \bar{c}_j \quad (1.5.4-12)$$

$$c \triangleq \sum_{j=1}^r \Lambda_j(k) \bar{c}_j \quad (1.5.4-13)$$

- Combination of the model-conditioned estimates and covariances (for output purpose):

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \mu_j(k) \quad (1.5.4-14)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \{P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)][\hat{x}^j(k|k) - \hat{x}(k|k)]'\} \quad (1.5.4-15)$$

1.5.4 Multiple Model Approach Overview

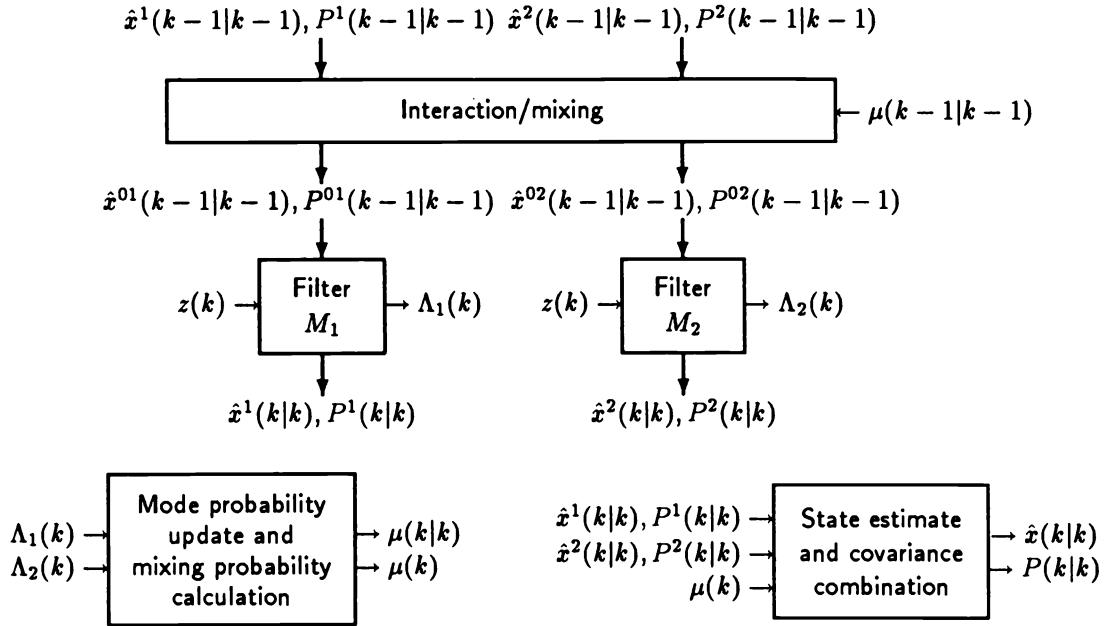


Figure 1.5.4-1: The IMM estimation algorithm: one cycle ($M_j(k) \triangleq \{M(k) = M_j\}$).

The Design Parameters for the IMM

- the set of modes:
 - the dynamic equations
 - * linear (2nd or 3rd order, etc.)
 - * nonlinear (**coordinated turn**, etc.) — the EKF is to be used instead of the KF
 - the process noise variances
- the mode transition probabilities.

Computational Requirements of the IMM

For 3 linear models the computational requirements are approximately 4 times higher than for the KF.

1.6 CONSISTENT COORDINATE CONVERSION IN TRACKING

1.6.1 The Coordinate Problem

In active sonar and radar systems the measurement of a target's position is reported in polar or spherical coordinates — its **range** and **azimuth** or **bearing**, as well as **elevation** angle in 3D radar, with respect to the sensor location. The measured range is actually the **slant range** if the target is at a different altitude than the sensor. Instead of elevation, altitude measurements are sometimes available in **Air Traffic Control (ATC)**.

In target tracking the *target motion can be best modeled in Cartesian coordinates*.

Tracking in Cartesian coordinates using such measurements can be done with:

- a linear Kalman Filter with the measurements converted to a Cartesian frame of reference — the **Converted Measurement KF (CMKF)**
- an Extended Kalman Filter (EKF), which incorporates the original measurements in a nonlinear fashion into the target state estimation — **mixed coordinate (linearized) filter**.

Using either technique the inaccuracy of the measurements when converted to a Cartesian frame of reference must be accounted for properly.

In the first approach the converted measurement covariance is recalculated at each sampling time.

In the second (EKF) approach the initial state covariance depends on the accuracy of the initial converted measurements and the gains depend on the accuracy of the subsequent linearizations — the overall performance depends critically on these accuracies.

The accuracy of the converted measurements depends on

- the target-sensor geometry, and
- the accuracy of the original measurements.

The mean and covariance of the converted measurements from **polar to Cartesian** are presented in Subsection 1.6.2. For certain levels of the measurement errors the mean of the errors after the conversion is significant — **debiasing compensation** is needed.

Both the debiasing terms as well as the covariance terms of the converted measurements require knowledge of the true location of the target. Since this is obviously not practical, the evaluation of these terms is usually done at the measured values, which introduces **secondary errors**. A procedure to account for these secondary errors is also given.

It is shown in Subsection 1.6.3 that, unlike the standard conversion (based on series expansion), this procedure guarantees **coordinate conversion consistency** over all practical

1.6.1 The Coordinate Problem

values of interest: the expressions for the first two moments of the converted measurements match their actual statistics.

Consequently, the CMKF is shown in Subsection 1.6.4 to be consistent — has estimation errors compatible with the calculated covariance. This is unlike the EKF which is consistent only for small errors.

Applicability of the CMKF

Since the CMKF has the correct covariance, it processes the measurements with a gain that is (nearly) optimal and yields smaller errors than the EKF, even in the case of moderately accurate sensors (.5° RMS azimuth error).

The EKF performs poorly at long range for an RMS azimuth error of 1.5° or more, while the CMKF is consistent even for a 10° RMS azimuth error.

1.6.2 The Errors in the Converted Measurements

1.6.2.2 The Errors in the Converted Measurements

The **measured range** r_m and **measured bearing** θ_m are defined w.r.t. the **true range** r and **true bearing** θ as

$$r_m = r + \tilde{r} \quad \theta_m = \theta + \tilde{\theta} \quad (1.6.2-1)$$

where the errors $\tilde{r}, \tilde{\theta}$, are assumed to be independent with zero mean and standard deviations σ_r and σ_θ , respectively.

The Standard Conversion

These polar measurements are transformed to Cartesian by the **standard coordinate conversion** as follows

$$\boxed{x_m = r_m \cos \theta_m \quad y_m = r_m \sin \theta_m} \quad (1.6.2-2)$$

The Error Statistics Obtained from Linearization

Denoting by (x, y) the true Cartesian position and taking the first order terms of the Taylor series expansion of (1.6.2-2) at (r_m, θ_m) , i.e., using *linearization*, yields the Cartesian coordinate errors

$$x_m - x \approx \tilde{r} \cos \theta_m - \tilde{\theta} r_m \sin \theta_m \triangleq \tilde{x}_L \quad (1.6.2-3)$$

$$y_m - y \approx \tilde{r} \sin \theta_m + \tilde{\theta} r_m \cos \theta_m \triangleq \tilde{y}_L \quad (1.6.2-4)$$

The mean of the errors as given by (1.6.2-3)–(1.6.2-4) is zero

$$\mu_L \triangleq \begin{bmatrix} E[\tilde{x}_L] \\ E[\tilde{y}_L] \end{bmatrix} = 0 \quad (1.6.2-5)$$

The elements of the corresponding covariance matrix R_L are

$$R_L^{11} \triangleq \text{var}(\tilde{x}_L) = r_m^2 \sigma_\theta^2 \sin^2 \theta_m + \sigma_r^2 \cos^2 \theta_m \quad (1.6.2-6)$$

$$R_L^{22} \triangleq \text{var}(\tilde{y}_L) = r_m^2 \sigma_\theta^2 \cos^2 \theta_m + \sigma_r^2 \sin^2 \theta_m \quad (1.6.2-7)$$

$$R_L^{12} \triangleq \text{cov}(\tilde{x}_L, \tilde{y}_L) = (\sigma_r^2 - r_m^2 \sigma_\theta^2) \sin \theta_m \cos \theta_m \quad (1.6.2-8)$$

1.6.2 The Errors in the Converted Measurements

The True Error Statistics

Rather than using linearization as in (1.6.2-3)–(1.6.2-4), the *exact errors* \tilde{x} and \tilde{y} in the Cartesian coordinates can be found by expanding

$$x_m \triangleq x + \tilde{x} = (r + \tilde{r}) \cos(\theta + \tilde{\theta}) \quad (1.6.2-9)$$

$$y_m \triangleq y + \tilde{y} = (r + \tilde{r}) \sin(\theta + \tilde{\theta}) \quad (1.6.2-10)$$

using trigonometric identities, to obtain

$$\tilde{x} = r \cos \theta (\cos \tilde{\theta} - 1) - \tilde{r} \sin \theta \sin \tilde{\theta} - r \sin \theta \sin \tilde{\theta} + \tilde{r} \cos \theta \cos \tilde{\theta} \quad (1.6.2-11)$$

$$\tilde{y} = r \sin \theta (\cos \tilde{\theta} - 1) + \tilde{r} \cos \theta \sin \tilde{\theta} + r \cos \theta \sin \tilde{\theta} + \tilde{r} \sin \theta \cos \tilde{\theta} \quad (1.6.2-12)$$

The mean and covariance of the errors (1.6.2-9)–(1.6.2-10) can be obtained explicitly and *exactly* assuming

- (i) zero-mean Gaussian errors in the polar measurements (1.6.2-1), and
- (ii) knowledge of the true location (r, θ) .

Assumption (i) yields

$$E[\cos \tilde{\theta}] = e^{-\sigma_\theta^2/2} \quad E[\sin \tilde{\theta}] = 0 \quad E[\sin \tilde{\theta} \cos \tilde{\theta}] = 0 \quad (1.6.2-13)$$

$$E[\cos^2 \tilde{\theta}] = \frac{1 + e^{-2\sigma_\theta^2}}{2} \quad E[\sin^2 \tilde{\theta}] = \frac{1 - e^{-2\sigma_\theta^2}}{2} \quad (1.6.2-14)$$

With the above, the *true mean* of the error in the converted measurements (1.6.2-2), *conditioned on the true location* according to Assumption (ii), is obtained as

$$\mu_t(r, \theta) \triangleq \begin{bmatrix} E[\tilde{x}|r, \theta] \\ E[\tilde{y}|r, \theta] \end{bmatrix} = \begin{bmatrix} r \cos \theta (e^{-\sigma_\theta^2/2} - 1) \\ r \sin \theta (e^{-\sigma_\theta^2/2} - 1) \end{bmatrix} \quad (1.6.2-15)$$

The elements of the *true covariance* R_t of the converted measurement are given by

$$\begin{aligned} R_t^{11} \triangleq \text{var}(\tilde{x}|r, \theta) &= r^2 e^{-\sigma_\theta^2} \{ \cos^2 \theta [\cosh(\sigma_\theta^2) - 1] + \sin^2 \theta \sinh(\sigma_\theta^2) \} \\ &\quad + \sigma_r^2 e^{-\sigma_\theta^2} [\cos^2 \theta \cosh(\sigma_\theta^2) + \sin^2 \theta \sinh(\sigma_\theta^2)] \end{aligned} \quad (1.6.2-16)$$

$$\begin{aligned} R_t^{22} \triangleq \text{var}(\tilde{y}|r, \theta) &= r^2 e^{-\sigma_\theta^2} \{ \sin^2 \theta [\cosh(\sigma_\theta^2) - 1] + \cos^2 \theta \sinh(\sigma_\theta^2) \} \\ &\quad + \sigma_r^2 e^{-\sigma_\theta^2} [\sin^2 \theta \cosh(\sigma_\theta^2) + \cos^2 \theta \sinh(\sigma_\theta^2)] \end{aligned} \quad (1.6.2-17)$$

$$R_t^{12} \triangleq \text{cov}(\tilde{x}, \tilde{y}|r, \theta) = \sin \theta \cos \theta e^{-2\sigma_\theta^2} [\sigma_r^2 + r^2 (1 - e^{\sigma_\theta^2})] \quad (1.6.2-18)$$

1.6.2 The Errors in the Converted Measurements

Remarks

Equations (1.6.2-15) and (1.6.2-16)–(1.6.2-18) are *exact explicit expressions* for the bias and covariance of the converted measurements.

The converted measurements have a significant bias for large cross-range errors (long ranges and large bearing errors).

Expressions (1.6.2-15) and (1.6.2-16)–(1.6.2-18) cannot be used due to Assumption (ii) — they are conditioned on the true values of range and bearing which are not available in practice.

To make these results useful, *the expected values of these true moments* have to be evaluated *conditioned on the measured position*.

Use of the True Error Statistics in Practice

The **average true converted measurement bias** and **average true converted measurement covariance** are

$$E[\mu_t(r, \theta)|r_m, \theta_m] \triangleq \mu_a \quad (1.6.2-19)$$

$$E[R_t(r, \theta)|r_m, \theta_m] \triangleq R_a \quad (1.6.2-20)$$

Using (1.6.2-13)–(1.6.2-15) and applying trigonometric identities the mean (1.6.2-19) becomes

$$\boxed{\mu_a \triangleq \begin{bmatrix} E[\tilde{x}|r_m, \theta_m] \\ E[\tilde{y}|r_m, \theta_m] \end{bmatrix} = \begin{bmatrix} r_m \cos \theta_m (e^{-\sigma_\theta^2} - e^{-\sigma_\theta^2/2}) \\ r_m \sin \theta_m (e^{-\sigma_\theta^2} - e^{-\sigma_\theta^2/2}) \end{bmatrix}} \quad (1.6.2-21)$$

Similarly, the covariance (1.6.2-20) has elements

$$\begin{aligned} R_a^{11} \triangleq \text{var}(\tilde{x}|r_m, \theta_m) &= r_m^2 e^{-2\sigma_\theta^2} [\cos^2 \theta_m (\cosh 2\sigma_\theta^2 - \cosh \sigma_\theta^2) + \sin^2 \theta_m (\sinh 2\sigma_\theta^2 - \sinh \sigma_\theta^2)] \\ &\quad + \sigma_r^2 e^{-2\sigma_\theta^2} [\cos^2 \theta_m (2 \cosh 2\sigma_\theta^2 - \cosh \sigma_\theta^2) + \sin^2 \theta_m (2 \sinh 2\sigma_\theta^2 - \sinh \sigma_\theta^2)] \end{aligned} \quad (1.6.2-22)$$

$$\begin{aligned} R_a^{22} \triangleq \text{var}(\tilde{y}|r_m, \theta_m) &= r_m^2 e^{-2\sigma_\theta^2} [\sin^2 \theta_m (\cosh 2\sigma_\theta^2 - \cosh \sigma_\theta^2) + \cos^2 \theta_m (\sinh 2\sigma_\theta^2 - \sinh \sigma_\theta^2)] \\ &\quad + \sigma_r^2 e^{-2\sigma_\theta^2} [\sin^2 \theta_m (2 \cosh 2\sigma_\theta^2 - \cosh \sigma_\theta^2) + \cos^2 \theta_m (2 \sinh 2\sigma_\theta^2 - \sinh \sigma_\theta^2)] \end{aligned} \quad (1.6.2-23)$$

$$R_a^{12} \triangleq \text{cov}(\tilde{x}, \tilde{y}|r_m, \theta_m) = \sin \theta_m \cos \theta_m e^{-4\sigma_\theta^2} [\sigma_r^2 + (r_m^2 + \sigma_r^2)(1 - e^{-\sigma_\theta^2})] \quad (1.6.2-24)$$

Thus, with μ_a given in (1.6.2-21), the **debiased conversion** is

$$\boxed{\begin{bmatrix} x^{dc} \\ y^{dc} \end{bmatrix} = \begin{bmatrix} r_m \cos \theta_m \\ r_m \sin \theta_m \end{bmatrix} - \mu_a} \quad (1.6.2-25)$$

1.6.2 The Errors in the Converted Measurements

The average covariance of the conversion is R_a with elements (1.6.2-22)–(1.6.2-24), which are *larger* than those in (1.6.2-16)–(1.6.2-18) — they account for the additional errors incurred by evaluating it at the measured position.

1.6.3 Consistency Evaluation of the Debiased and Standard Conversions

1.6.3.3 Consistency Evaluation of the Debiased and Standard Conversions

This section examines the *single scan* converted measurement errors — the static situation. The use of the converted measurements in tracking — the dynamic situation — is discussed in the next subsection.

The debiased conversion (1.6.2-25) with covariance R_a (1.6.2-22)–(1.6.2-24) is shown to be the *only* consistent conversion of the measurements when using measured values of range and bearing for all target locations and sensor accuracies.

The (ideal) true covariance R_t is always consistent provided the true position is used in its evaluation, but it becomes optimistic when it relies on the measured position.

The *limits of validity of the standard conversion* with the covariance R_L based on linearization are also discussed.

The analysis is done by performing the statistical consistency check (Section 5.4 of [BL93]) as follows.

For a zero-mean random vector \tilde{z} of dimension n_z and covariance P the expected value of the **Normalized Error Squared (NES)**

$$\psi = \tilde{z}' P^{-1} \tilde{z} \quad (1.6.3-1)$$

is equal to n_z .

The sample mean of (1.6.3-1) from n independent realizations

$$\bar{\psi} = \frac{1}{N} \sum_{i=1}^N \tilde{z}_i' P^{-1} \tilde{z}_i \quad (1.6.3-2)$$

with \tilde{z}_i the 2-dimensional vector of converted measurement errors in realization i , is used to perform the consistency test.

The mean value of the sample test statistic is 2 when there is no bias and the assumed covariance matches the actual errors.

If the errors are jointly Gaussian then the distribution of (1.6.3-2) multiplied by N is chi-square with $2N = 2000$ degrees of freedom and an acceptance region is used for the test.

1.6.3 Consistency Evaluation of the Debiased and Standard Conversions

Numerical Results for Consistency

The above test was performed using $N = 1000$ converted position measurement for each position and accuracy considered.

The average NES for the debiased conversion (1.6.2-25) with covariance R_a evaluated at the measured position, is plotted as a function of the bearing error in the top part of Figure 1.6.3-1. The true target position is at $10^5 m$ with azimuth 45° and the standard deviation of range error is $50m$. The plot also indicates the chi-square 0.99 probability bounds.

The same results for the standard conversion (1.6.2-2) and covariance R_L are shown in the bottom part of Figure 1.6.3-1.

The covariance R_a is consistent with the errors in the debiased conversion even though the variability in the plot marginally violates the chi-square bounds since the errors are not Gaussian.

The linear approximation for the standard conversion results in an *optimistic covariance* R_L , which becomes inconsistent when the bearing error exceeds 0.8° at this range.

Nearly identical results are obtained at different bearing angles.

1.6.3 Consistency Evaluation of the Debiased and Standard Conversions

Validity Limit of the Standard Conversion

The maximum bias magnitude is, from (1.6.2-21), using a series expansion,

$$\|\mu\|_{\max} = \frac{r\sigma_\theta^2}{2} \quad (1.6.3-3)$$

Note that σ_θ above is in *radians*.

The minimum standard deviation from the covariance matrix (1.6.2-22)–(1.6.2-24) is

$$\sqrt{\lambda_{\min}(R_a)} = \min(\sigma_r, r\sigma_\theta) \quad (1.6.3-4)$$

where $\lambda_{\min}(R_a)$ denotes the minimum eigenvalue.

Define the **bias significance** as the ratio of the above two quantities

$$\beta \triangleq \frac{r\sigma_\theta^2}{2 \min(\sigma_r, r\sigma_\theta)} \quad (1.6.3-5)$$

Based on the results of Figure 1.6.3-1 (bottom part), the maximum bias significance that can be tolerated corresponds to $\sigma_\theta = 0.8^\circ$, i.e.,

$$\beta_{\max} = \frac{10^5 (0.8\pi/180)^2}{2 \cdot 50} \approx 0.2 \quad (1.6.3-6)$$

Thus the **validity limit of the standard conversion** is

$$\frac{r\sigma_\theta^2}{\sigma_r} < 0.4 \quad (1.6.3-7)$$

$$\sigma_\theta < 0.4 \text{ rad} \approx 23^\circ \quad (1.6.3-8)$$

In practically all systems inequality (1.6.3-8) is satisfied. In most radar systems, inequality (1.6.3-7) is satisfied, but this is not always the case in sonar systems or in **over the horizon radar**.

While the 0.8° threshold in (1.6.3-6) is somewhat subjective — one could use a slightly higher threshold — this is a useful guideline for the limit of validity of the standard conversion.

Conclusion

The debiased conversion from polar to Cartesian measurements of Section 1.6.2 is

- unbiased
- has a correct covariance,

i.e., it is a *consistent conversion* for all practical parameters.

1.6.3 Consistency Evaluation of the Debiased and Standard Conversions

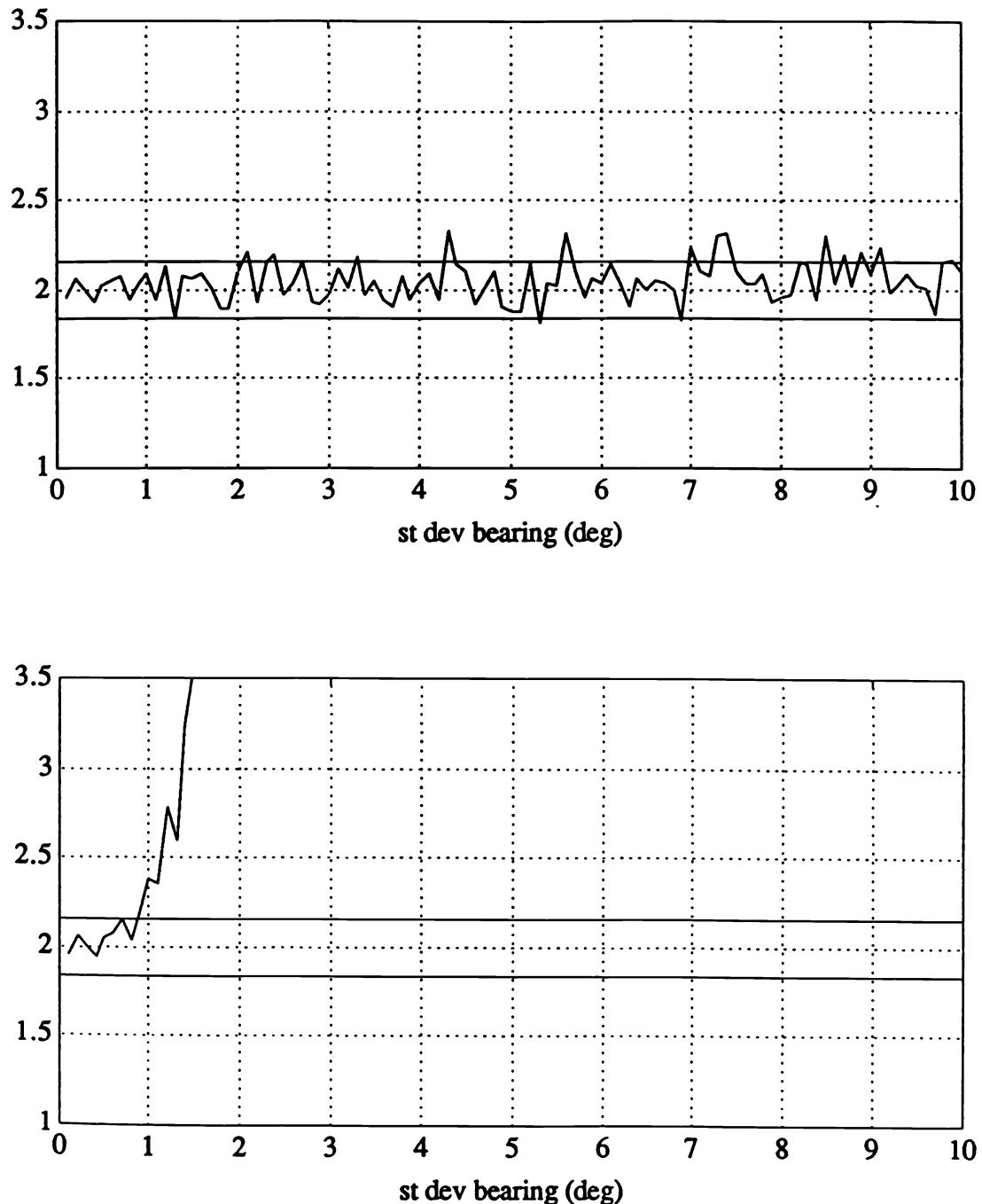


Figure 1.6.3-1: Average NES for the debiased conversion (top) and for the standard conversion (bottom) evaluated at the measured position. ($r = 10^5 m$, $\theta = 45^\circ$, $\sigma_r = 50 m$)

1.6.4 Tracking with Converted Measurements vs. EKF

1.6.4 Tracking with Converted Measurements vs. EKF

The motion of a target is best described in Cartesian coordinates. When tracking it with polar measurements, there are three options for the tracking filter:

- (i) Convert the measurements from polar to Cartesian and use a linear filter — the **Converted Measurement Kalman Filter (CMKF)**
 - (a) using the standard conversion (1.6.2-2) and the covariance matrix R_L , given in (1.6.2-6)–(1.6.2-8), based on linearization — this is the **Converted Measurement Kalman Filter with Linearization (CMKF-L)**
 - (b) using the debiased conversion (1.6.2-25) with the covariance matrix R_a , given in (1.6.2-22)–(1.6.2-24), which yields unbiased converted measurements — this is the **Converted Measurement Kalman Filter with Debiasing (CMKF-D)**
- (ii) Keep the measurements in their original (polar) form and use a **mixed coordinate Extended Kalman Filter (EKF)**.

These three design options are described next and their performance compared.

The Converted Measurement Filter

With the CMKF, the polar position measurement $z^p(k)$ at time k is converted to a Cartesian coordinate measurement $z^c(k)$ using the nonlinear transformation η as in (1.6.2-25)

$$z^c(k) = \begin{cases} \eta[z^p(k)] & \text{for CMKF-L} \\ \eta[z^p(k)] - \mu_a(k) & \text{for CMKF-D} \end{cases} \quad (1.6.4-1)$$

where $\mu_a(k)$ is the bias compensation (1.6.2-21) and

$$\eta[z^p(k)] \triangleq \begin{bmatrix} r_m(k) \cos \theta_m(k) \\ r_m(k) \sin \theta_m(k) \end{bmatrix} \quad (1.6.4-2)$$

The filter is entirely in Cartesian coordinates — it is a linear filter.

The predicted state with the state transition matrix F is

$$\hat{x}(k|k-1) = F\hat{x}(k-1|k-1) \quad (1.6.4-3)$$

Assuming a white noise acceleration and sampling period T , one has

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6.4-4)$$

1.6.4 Tracking with Converted Measurements vs. EKF

The predicted measurement is

$$\hat{z}^c(k|k-1) = H\hat{x}(k|k-1) \quad (1.6.4-5)$$

where the measurement matrix is

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1.6.4-6)$$

The updated state estimate is, with the Kalman gain $W(k)$,

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)[z^c(k) - \hat{z}^c(k|k-1)] \quad (1.6.4-7)$$

The predicted state covariance is

$$P(k|k-1) = FP(k-1|k-1)F' + Q \quad (1.6.4-8)$$

where $P(k-1|k-1)$ is the state error covariance at time $k-1$ and Q is the assumed process noise covariance.

The predicted measurement covariance is

$$S(k) = HP(k|k-1)H' + R^c(k) \quad (1.6.4-9)$$

where $R^c(k)$ is the **converted measurement covariance** which will be

- R_L , from linearization, for CMKF-L
- R_a , the average true value, for CMKF-D.

The Kalman gain is

$$W(k) = P(k|k-1)H'S(k)^{-1} \quad (1.6.4-10)$$

and the updated state covariance is

$$P(k|k) = P(k|k-1) - W(k)S(k)W(k)' \quad (1.6.4-11)$$

1.6.4 Tracking with Converted Measurements vs. EKF

Evaluation of the Converted Measurement Covariance

The converted measurement covariance is a function of the target location, which is imperfectly known; the debiased converted measurement covariance is also a function of the *uncertainty in the location at which it is evaluated*.

In the static (single measurement) case the only option for the evaluation of the converted measurement covariance is at the measurement. In the dynamic situation one can evaluate it at

- the measurement, or
- the predicted position — by applying the inverse mapping of η , defined in (1.6.4-2).

The converted measurement covariance should be evaluated using the *more accurate* of the above options.

In general, the predicted position is more accurate than the latest measurement. The only time this does not hold is at initialization, where the first few time steps may provide poor estimates of the target position relative to the measurement accuracy. In this situation, the measurements themselves should be chosen to evaluate the conversion covariance.

The following simple rule determines when to use the current measurements vs. the predicted state. The measurements are used if

$$|HP(k|k-1)H'| \geq |R^c(k)| \quad (1.6.4-12)$$

i.e., if the uncertainty volume of the predicted measurement exceeds the one of the measurement error then the “more accurate” measurements should be used to evaluate the converted measurement error covariance.

1.6.4 Tracking with Converted Measurements vs. EKF

The Mixed Coordinate Extended Kalman Filter

In this case there is a nonlinear measurement prediction equation and linearization of the measurement equation is required for the update.

The measurements at time k are

$$z^p(k) = \begin{bmatrix} r_m(k) \\ \theta_m(k) \end{bmatrix} \quad (1.6.4-13)$$

The state prediction proceeds as in (1.6.4-3) but the measurement prediction is the nonlinear mapping

$$\hat{z}^p(k|k-1) = h[\hat{x}(k|k-1)] \quad (1.6.4-14)$$

where $h \triangleq \eta^{-1}$ from (1.6.4-2) and the updated state is

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)[z^p(k) - \hat{z}^p(k|k-1)] \quad (1.6.4-15)$$

where the “mixed coordinate” gain $W(k)$ is different from the gain in (1.6.4-7) since it is obtained from a different covariance update and is applied to the polar measurements.

The predicted state covariance is the same as (1.6.4-8) and the predicted measurement covariance (the innovation covariance) is

$$S(k) = H_x(k)P(k|k-1)H_x(k)' + R^p \quad (1.6.4-16)$$

where $H_x(k)$ is the Jacobian of the transformation h evaluated at the predicted state estimate $\hat{x}(k|k-1)$ and R^p is the polar measurement covariance

$$R^p = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \quad (1.6.4-17)$$

The gain in the EKF is

$$W(k) = P(k|k-1)H_x(k)'S(k)^{-1} \quad (1.6.4-18)$$

and the updated state covariance is the same as (1.6.4-11) using (1.6.4-16) and (1.6.4-18).

The use of the Jacobian, a first order approximation, can cause inconsistency of the innovation covariance (1.6.4-16) when the linearization is poor, which results in an improper gain (1.6.4-18).

1.6.4 Tracking with Converted Measurements vs. EKF

Simulation Results

A long range target tracking application was simulated to compare the three filter designs.

The sensor is assumed fixed at the origin. The measurement standard deviations were $\sigma_r = 50m$ in range and $\sigma_\theta = 1.5^\circ$ and 2.5° in bearing.

The initial target location is at a range of 70km and azimuth 45° with initial velocity of $15m/s$ heading due North. The target trajectory includes process noise (white noise acceleration) with a standard deviation $0.01m/s^2$ in each coordinate.

Tracking is performed using 50 measurements obtained with sampling interval of $T = 60s$. The tracks were initiated with two point differencing.

Two converted measurement filters, the CMKF-D using the *debiased* conversion and the CMKF-L with the standard conversion with *linearized* measurement covariance are examined. These algorithms are compared with the mixed coordinate EKF.

All results presented are based on 1000 Monte Carlo runs.

The RMS position error and the RMS velocity error for the three filters are shown in Figures 1.6.4-1 and 1.6.4-2, respectively, for $\sigma_\theta = 1.5^\circ$. The CMKF-D provides the best accuracy. The other two filters are close in accuracy except for the EKF which is poor for the first 10 scans.

Figures 1.6.4-3 and 1.6.4-4 present the NEES for position and velocity with their associated 99% probability regions. The CMKF-D is consistent while the CMKF-L is inconsistent for the first 15 scans and the EKF is inconsistent over the first 40 scans.

Figures 1.6.4-5–1.6.4-8 present the same results with $\sigma_\theta = 2.5^\circ$. In this case the CMKF-D is clearly superior to both other filters.

1.6.4 Tracking with Converted Measurements vs. EKF

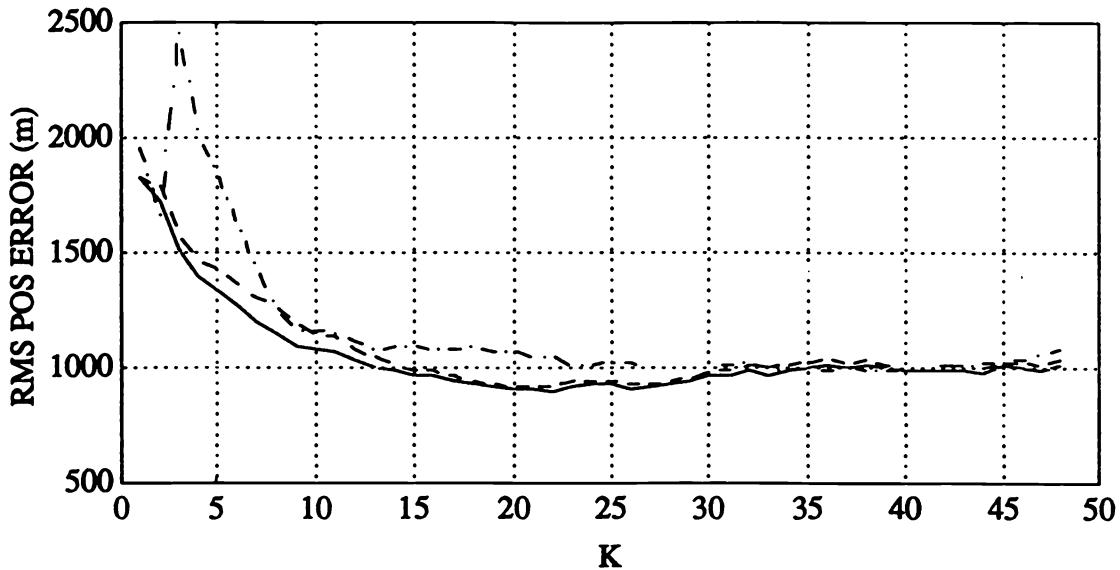


Figure 1.6.4-1: Comparison of position RMS errors for $\sigma_\theta = 1.5^\circ$: — CMKF-D; - - CMKF-L; - · - EKF.

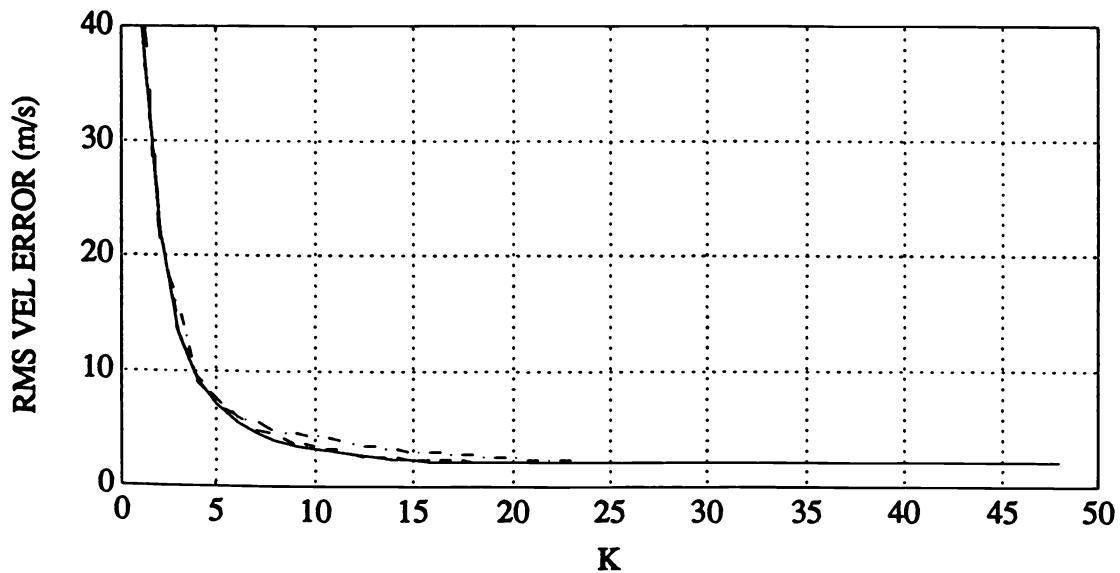


Figure 1.6.4-2: Comparison of velocity RMS errors for $\sigma_\theta = 1.5^\circ$: — CMKF-D; - - CMKF-L; - · - EKF.

1.6.4 Tracking with Converted Measurements vs. EKF

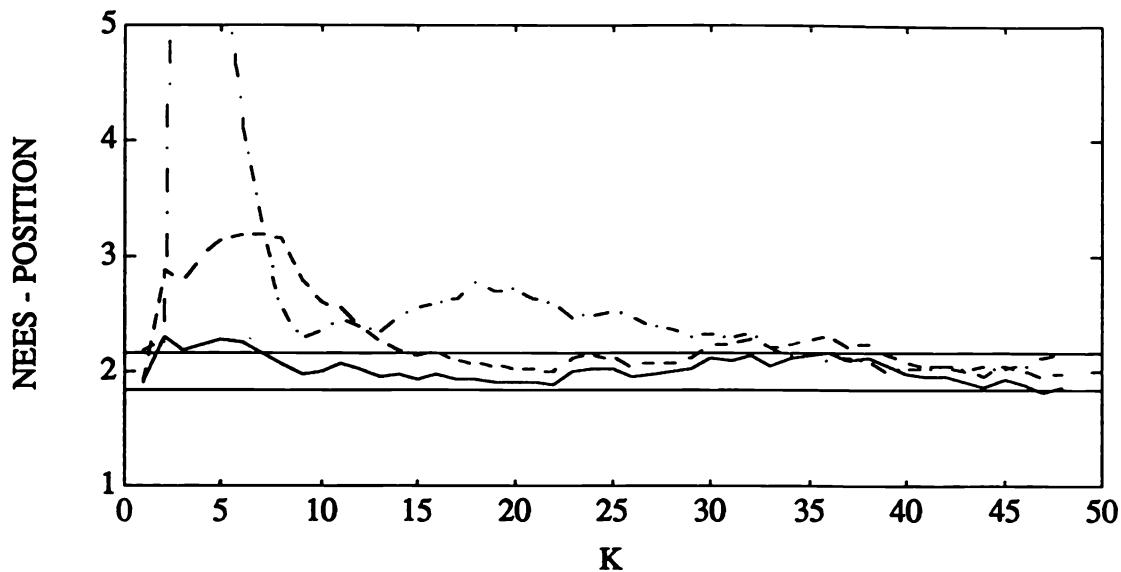


Figure 1.6.4-3: Comparison of position NEES errors for $\sigma_\theta = 1.5^\circ$: — CMKF-D; -- CMKF-L; - · - EKF.

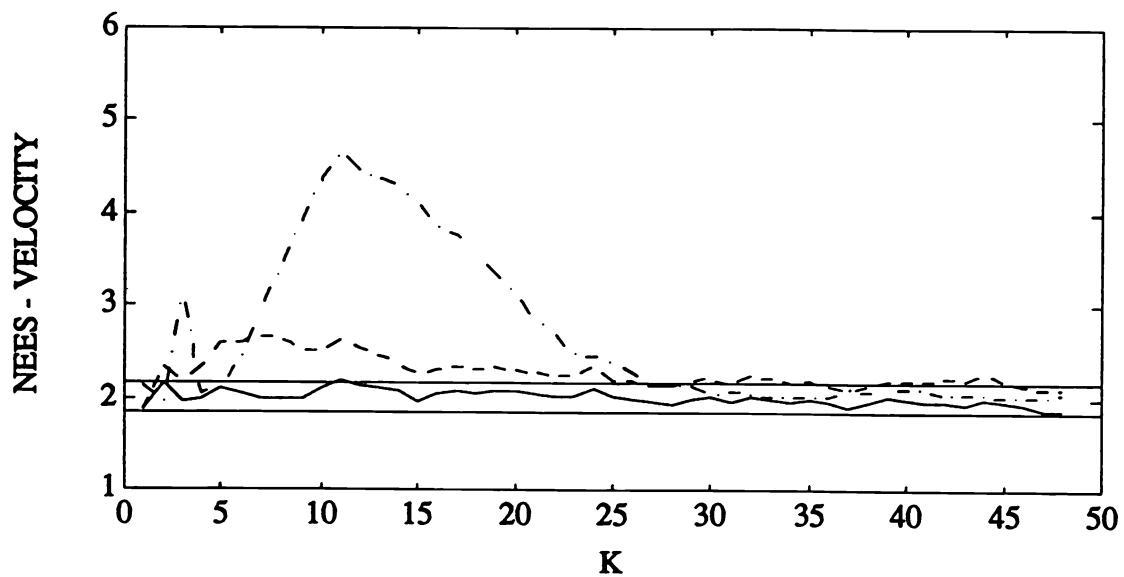


Figure 1.6.4-4: Comparison of velocity NEES errors for $\sigma_\theta = 1.5^\circ$: — CMKF-D; -- CMKF-L; - · - EKF.

1.6.4 Tracking with Converted Measurements vs. EKF

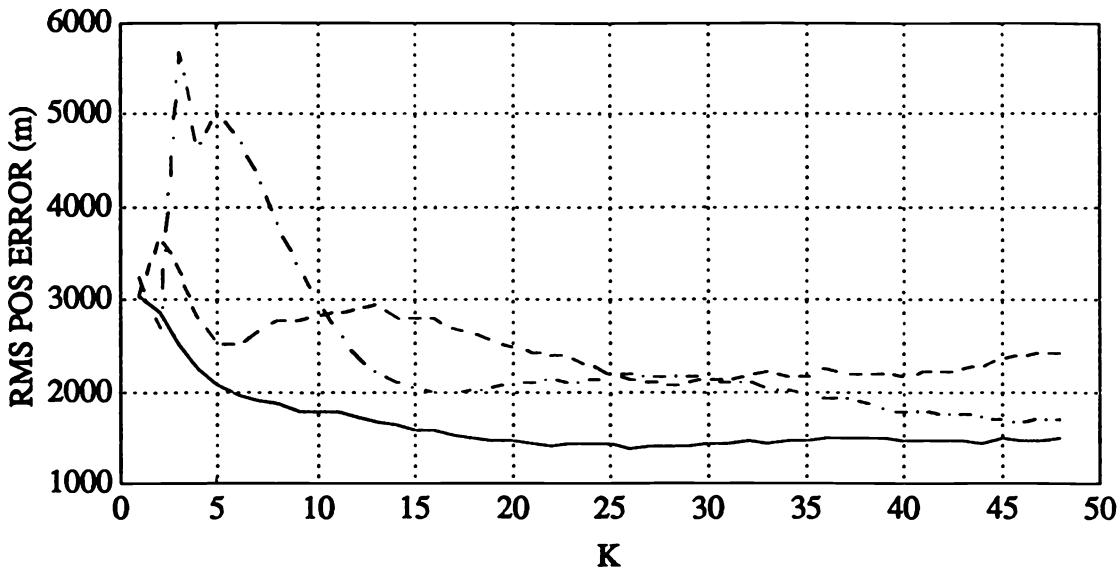


Figure 1.6.4-5: Comparison of position RMS errors for $\sigma_\theta = 2.5^\circ$: — CMKF-D; - - CMKF-L; - · - EKF.

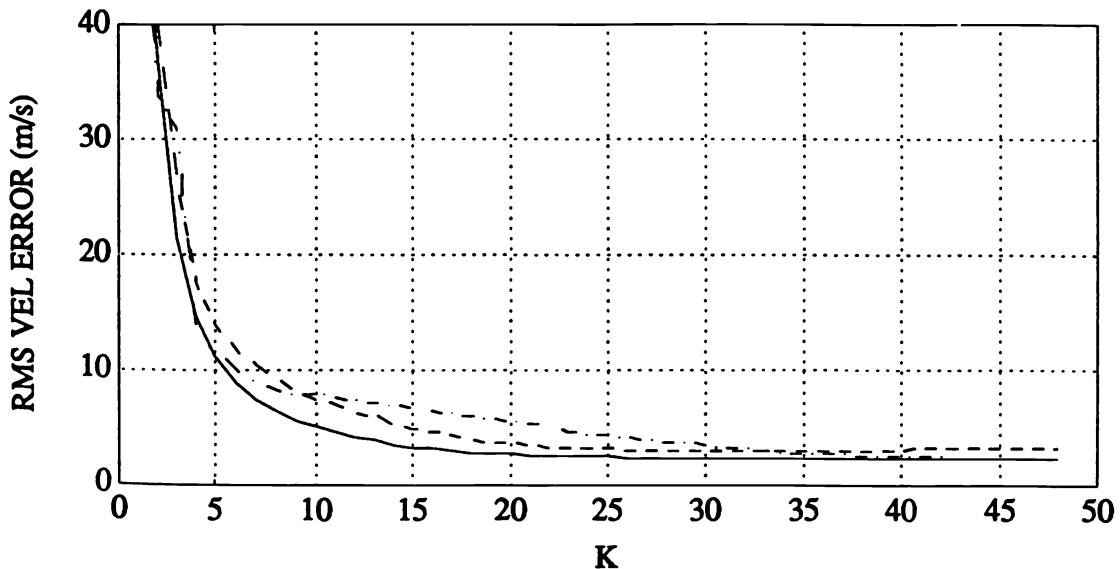


Figure 1.6.4-6: Comparison of velocity RMS errors for $\sigma_\theta = 2.5^\circ$: — CMKF-D; - - CMKF-L; - · - EKF.

1.6.4 Tracking with Converted Measurements vs. EKF

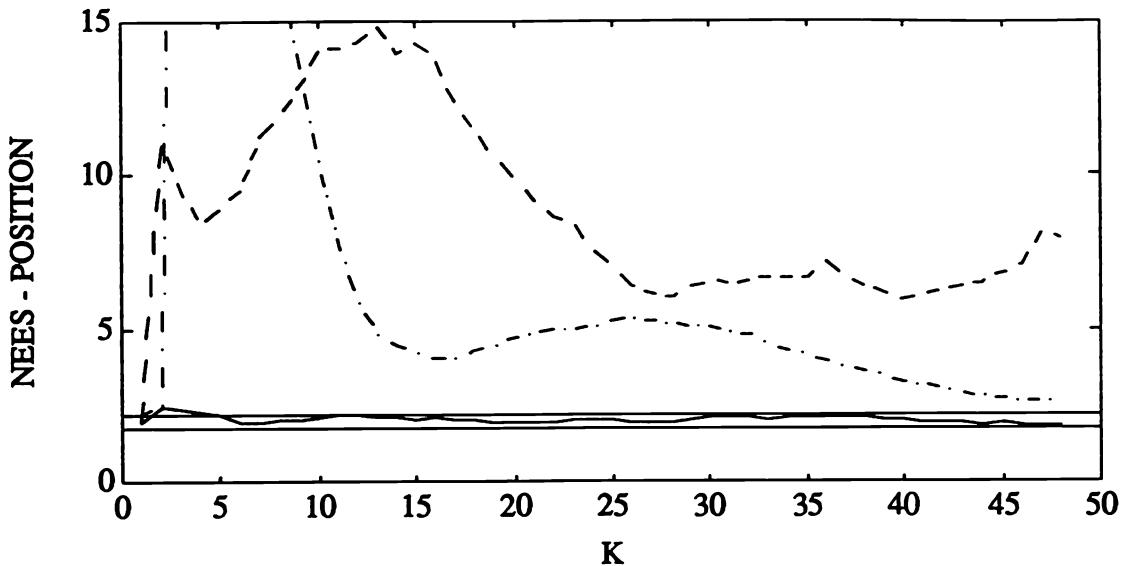


Figure 1.6.4-7: Comparison of position NEES errors for $\sigma_\theta = 2.5^\circ$: — CMKF-D; - - CMKF-L; - · - EKF.

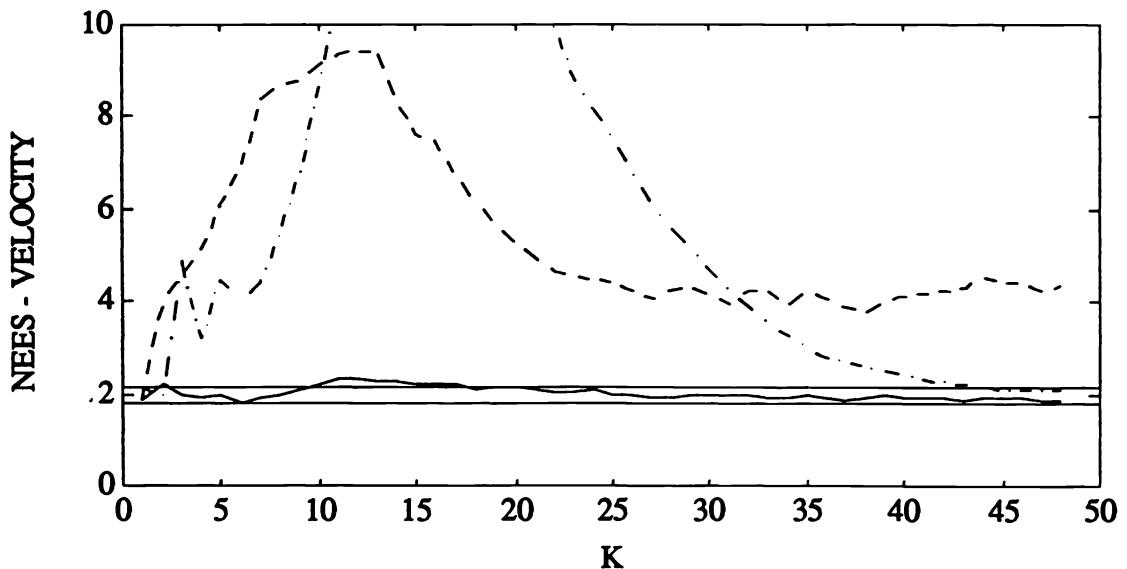


Figure 1.6.4-8: Comparison of velocity NEES errors for $\sigma_\theta = 2.5^\circ$: — CMKF-D; - - CMKF-L; - · - EKF.

1.6.5 Coordinate Conversion — Summary

1.6.5 Coordinate Conversion — Summary

The mean and covariance of the errors of Cartesian measurements which are obtained by converting polar measurements have been derived. The standard conversion is

- biased
- has optimistic covariance.

A procedure has been presented which guarantees consistency of the conversion for any target geometry or polar measurement accuracy using the measured target position.

This procedure, which carries out

- debiasing
- covariance calculation that accounts for the unavoidable errors in the position at which it is evaluated,

is only slightly more complex than the commonly used standard conversion.

When used in target tracking, this clearly provides better estimation accuracy than the filter using the standard conversion and the mixed coordinate Extended Kalman Filter.

Furthermore, this filter is the only filter which is consistent over all practical geometries and sensor accuracies.

Recommendation

The debiased conversion should be used whenever the limit of the validity of the standard conversion is exceeded.

1.7 EXAMPLE OF DESIGN OF AN IMM ESTIMATOR

1.7.1 A Generic ATC Tracking Problem

A generic **Air Traffic Control (ATC)** tracking problem is considered.

The **ground truth** is a target moving in a plane with a constant speed of $120m/s$ with initial state in Cartesian coordinates

$$\dot{x} = [\xi \ \dot{\xi} \ \eta \ \dot{\eta}]' = [0 \ 0 \ 0 \ 120]' \quad (1.7.1-1)$$

of dimension $n_x = 4$.

The sampling period is $T = 10s$. This low sampling rate is typical of the current long range radars for the “en route” traffic.

The following maneuver is considered: at $k = 7$ ($t = 70s$) the target starts a right turn with a rate of $3^\circ/s$ (which corresponds to an acceleration of $6m/s^2$) for $30s$, then continues straight until $k = 20$.

Measurements are assumed to be available on the Cartesian position (ξ, η) of this target with additive zero-mean white Gaussian noise with covariance matrix $R = \text{diag}(\sigma_{w_\xi}^2, \sigma_{w_\eta}^2)$, where $\sigma_{w_\xi} = \sigma_{w_\eta} = 100m$. While the radar measurements are the (slant) range and azimuth, with typical standard deviations of $100\text{--}300m$ in range and $4\text{--}5mrad$ in azimuth, the present example considers for simplicity direct measurements of the Cartesian position. The altitude is not considered here; see Section 8.7 for an example with real data and motion in 3 dimensions.

The state estimator is assumed to be initialized from measurements as above at $k = -1$ and $k = 0$ and starts running from $k = 1$. Each model-matched filter in the IMM is initialized in the same manner and their initial probabilities are taken as equal.

1.7.2 Mode Selection for the IMM Design

1.7.2 Mode Selection for the IMM Design

Kalman filters will be used as modules of the IMM estimation algorithm.

The type of model to be used is 2nd order kinematic (Subsection 6.3.2 of [BL93]) in Cartesian coordinates, with the process noise, which represents the unknown acceleration to the tracker, assumed zero-mean white Gaussian with different variances for the different modes.

For this type of models,

- the process noise and
- the Markov chain transition matrix between the modes

are the **IMM estimator design parameters**.

Other models, e.g.,

- constant speed circular motion (“coordinated turn” — a nonlinear model that requires as additional state the turn rate — see Subsection 11.6.8 of [BL93]), and
- acceleration that is exponentially autocorrelated (see Section 8.2 of [BL93]) or a Wiener process (see Subsection 11.6.7 of [BL93]),

could also be used.

The following are the modes (models) considered here:

- (i) Mode \mathcal{U} for the **uniform motion** (constant velocity, also called **straight and level**) portion of the trajectory, with a low level of process noise;
- (ii) Mode \mathcal{T} for the **turning** (maneuvering) portion, with process noise of a level compatible with known magnitude of the maneuver, uncorrelated between the two coordinates.

A 2-model IMM, designated as IMM2, using the model set $\{\mathcal{U}, \mathcal{T}\}$, will be designed and its performance illustrated.

Also a Kalman filter based on a (single) model of the same type (without adaptation) is examined as a baseline for comparison.

1.7.3 Design Parameter Selection — Filter Tuning

1.7.3 Design Parameter Selection — Filter Tuning

The process noise standard deviations (accelerations in m/s^2) in the ξ and η coordinates are shown in Table 1.7.3-1 for the two models described above and for the Kalman filter, denoted as KF.

Mode	\mathcal{U}	T	KF
$\sigma_{v_\xi} = \sigma_{v_\eta}$	0.3	6	5

Table 1.7.3-1: Process noise design parameters.

These modes have a **target maneuvering index** — ratio of the **RMS (root mean square)**⁵ motion uncertainty in position over one sampling period to the RMS measurement uncertainty)

$$\lambda \triangleq \sigma_v T^2 / \sigma_w \quad (1.7.3-1)$$

(see Section 6.5 of [BL93]) of 0.3 and 6, respectively.

The Markov chain transition matrix between the two modes of the IMM2 was chosen as

$$[p_{ij}] = \begin{bmatrix} 0.9 & 0.1 \\ 0.33 & 0.67 \end{bmatrix} \quad (1.7.3-2)$$

The rationale for the choice of $p_{12} = 0.1$ is that a maneuver is likely to start with a relatively low probability; however, if the assumed probability is too low, the filter will be slow in adapting to it.

The rationale for the choice of $p_{22} = 0.67$ is the following: **ICBS**⁶ that the *expected sojourn time* τ_i of a Markov chain in state i (mode i) is, in *sampling periods*, given by

$$E[\tau_i] = \frac{1}{1 - p_{ii}} \quad (1.7.3-3)$$

Thus, the above indicated value is selected with an anticipation of turns that last about 3 periods.

⁵The square root of the expected value of the square of the corresponding variable.

⁶The common acronym for It Can Be Shown, as opposed to **ICBES** (It Can Be Easily Shown).

1.7.4 Simulation Results and Discussion

The results presented next are all based on $N_r = 1000$ Monte Carlo runs. The *experimental* (i.e., Monte Carlo) **RMS (root mean square)** error in position from N_r runs is defined as

$$\sigma_{\text{pos}}(k) = \sqrt{\frac{1}{N_r} \sum_{i=1}^{N_r} \{[\xi^i(k) - \hat{\xi}^i(k|k)]^2 + [\eta^i(k) - \hat{\eta}^i(k|k)]^2\}} \quad (1.7.4-1)$$

where superscript i denotes the results from run i . A similar expression yields the velocity RMS error. Note that this is, in statistical terminology, a *sample error*, rather than the expected value.

Estimation Accuracy

Figure 1.7.4-1 presents the RMS position errors (ξ and η *combined*) for the IMM2 and the Kalman filter, implemented with the above parameters. Note that the IMM2 has the largest RMS position error between 140 and 150m vs. the single measurement position RMS error, which is $100\sqrt{2} \approx 140m$. However, when the target is not maneuvering, the RMS error is around 100m, i.e., a reduction of about 30% from the raw measurement RMS error. This corresponds to a 50% reduction in the MS error.

The KF has *all the time* an RMS position error of slightly under 140m — this can be predicted based on the fact that the maneuvering index used in the KF is $\lambda = 5$, which yields (according to the curves of the α - β filter, Subsection 6.5.3 of [BL93]) $\alpha = 0.945$, i.e., a minimal $(1 - \sqrt{\alpha} = 3\%)$ position RMS error improvement (**RMS noise reduction**) over the single measurement accuracy. During the maneuver, the error stays at about the same level.

Using a KF with a lower maneuvering index of $\lambda = 3$ (not shown) yields a peak error of 150m, and 130m when the target is not maneuvering, i.e., significantly inferior to the IMM. Thus, the higher value $\lambda = 5$ is preferable.

The reason for choosing a certain level of process noise for model \mathcal{U} is that, even in the absence of maneuvers, changes in the speed of the order of 3m/s over the 10s sampling period can occur due to winds, etc.

The velocity errors shown in Figure 1.7.4-2 are similar: the IMM2 exhibits a higher error during the maneuver (by about 25%), but during the straight portions of the trajectory the velocity error is *5 times smaller* than in the KF.

1.7.4 Simulation Results and Discussion

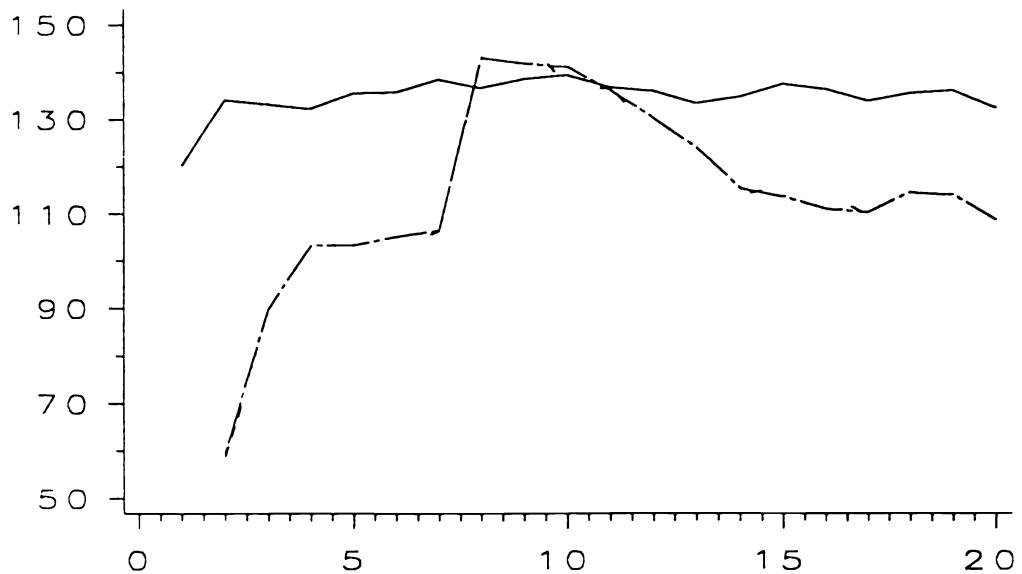


Figure 1.7.4-1: RMS errors in position (m): — KF; - - IMM2.

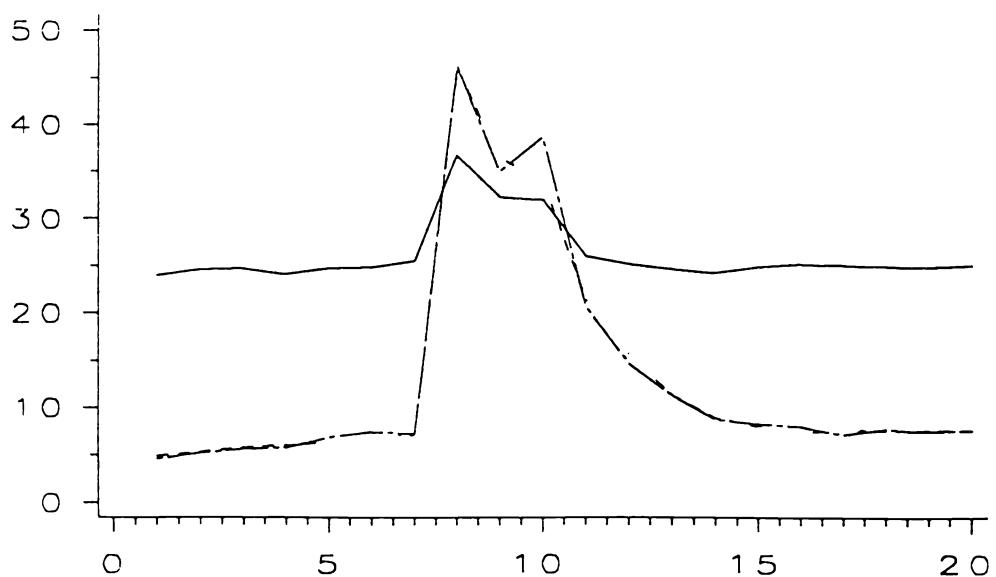


Figure 1.7.4-2: RMS errors in velocity (m/s): — KF; - - IMM2.

1.7.4 Simulation Results and Discussion

Estimator Consistency

Figures 1.7.4-3 and 1.7.4-4 present the comparison of the RMS errors and the corresponding estimator-calculated standard deviations (*SD*) for ξ and $\dot{\xi}$, respectively, in the KF and the IMM2. The KF, which is very conservative — uses permanently a large process noise variance to “cover” the maneuver — has a large SD and commensurate position errors; its velocity SD is “pessimistic,” except during the maneuver.

The IMM2, with overall lower errors, is somewhat “optimistic” during the maneuver (both in position and velocity) — its RMS errors exceed the SD calculated by the algorithm; on the other hand, during the straight and level portions of the trajectory, it is somewhat “pessimistic” — the RMS errors are below the estimator-calculated SD.

Thus the IMM is not entirely consistent (in the sense of having errors that are commensurate with the estimator-calculated SD; see [BL93] Section 5.4) — its errors are sometimes larger, sometimes smaller than the calculated SD. This is typical of adaptive algorithms and is the price one has to pay for *overall better performance*. This is discussed in more detail later.

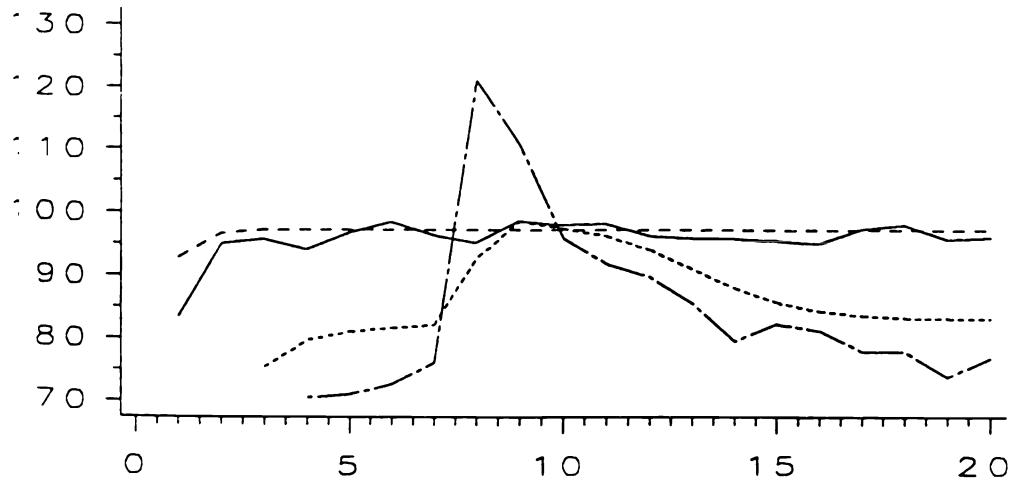


Figure 1.7.4-3: RMS errors in ξ vs. calculated SD (m): — KFRMS; - - KFSD; - - - IMM2RMS; - - - IMM2SD.

1.7.4 Simulation Results and Discussion

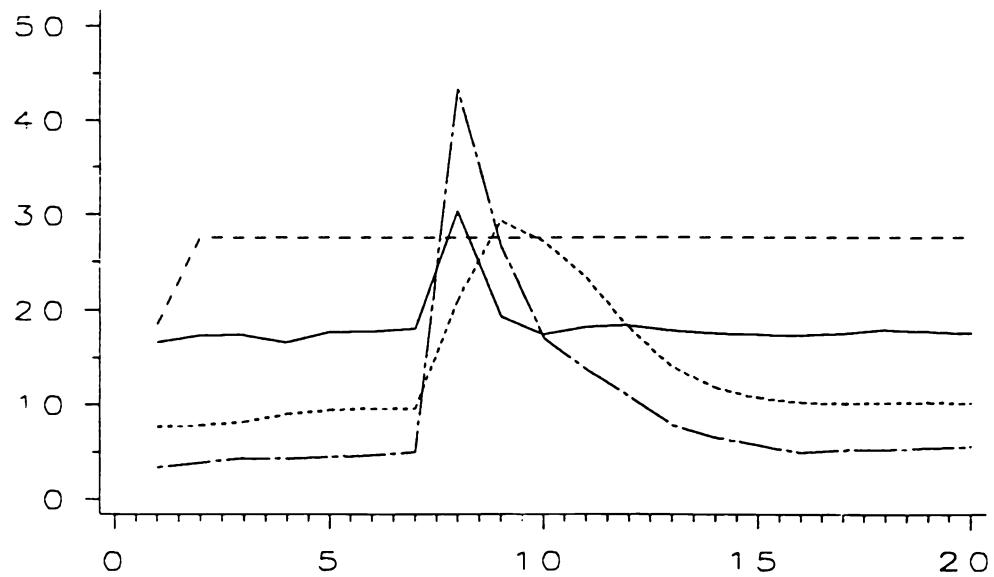


Figure 1.7.4-4: RMS errors in $\dot{\xi}$ vs. calculated SD (m/s): — KFRMS; -- KFSD; - - IMM2RMS; - - - IMM2SD.

1.7.4 Simulation Results and Discussion

The **Normalized Estimation Error Squared (NEES)** is defined as

$$\bar{\epsilon} = \frac{1}{N_r} \sum_{i=1}^{N_r} \{[x^i(k) - \hat{x}^i(k|k)]' P^i(k|k)^{-1} [x^i(k) - \hat{x}^i(k|k)]\} \quad (1.7.4-2)$$

where superscript i denotes the results from run i . Under ideal conditions, the NEES has an n_x (dimension of the state x) degrees of freedom chi-square distribution.

Figure 1.7.4-5 shows the average NEES from 1000 runs of the IMM2 for the state vector (of dimension 4). For a 1000-run average, the 95% probability region based on the $n_x = 4$ degrees of freedom chi-square distribution is the interval [3.83, 4.17]. Since the conditions are not ideal — the filter is facing either no process noise at all or a maneuver which is not a zero-mean Gaussian process noise — the NEES is smaller than expected before and after the maneuver and larger than expected at time $k = 8$, the first instant the maneuver is noticed. The large overshoot in the NEES is due to the error (bias) that develops with the onset of the maneuver, which is, however, corrected by $k = 9$.

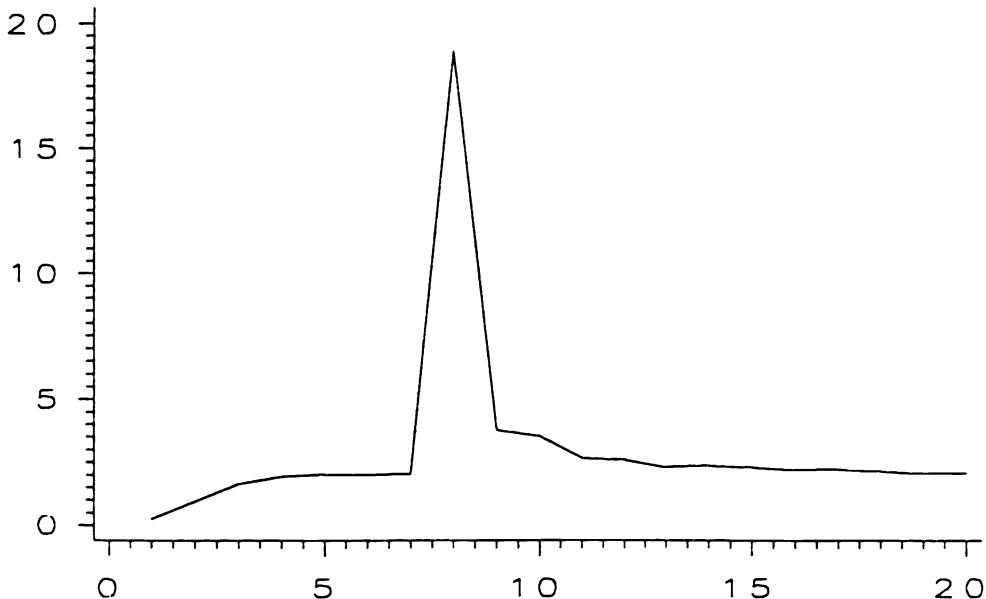


Figure 1.7.4-5: The average state NEES for the IMM2.

This parallels the earlier observations about Figures 1.7.4-3 and 1.7.4-4 that when the target is not maneuvering the algorithm is pessimistic, while at the onset of the maneuver it is optimistic until it “catches up.” No adaptive filter can be made entirely consistent because *it is the inconsistency that drives the timely adaptation.*

1.7.4 Simulation Results and Discussion

Maneuver Detection

Figure 1.7.4-6 shows the evolution of the average mode probabilities in the IMM2. Mode \mathcal{U} (#1) has about a 95% probability when the target is not maneuvering (both before as well as shortly thereafter). The maneuver starts at $k = 7$, i.e., it can be “noticed” at $k = 8$, at which time the probability of mode 2 becomes 25%; by $k = 9$ it is 90% and then it decays rapidly. This indicates rapid “detection” of the maneuver by the IMM algorithm.

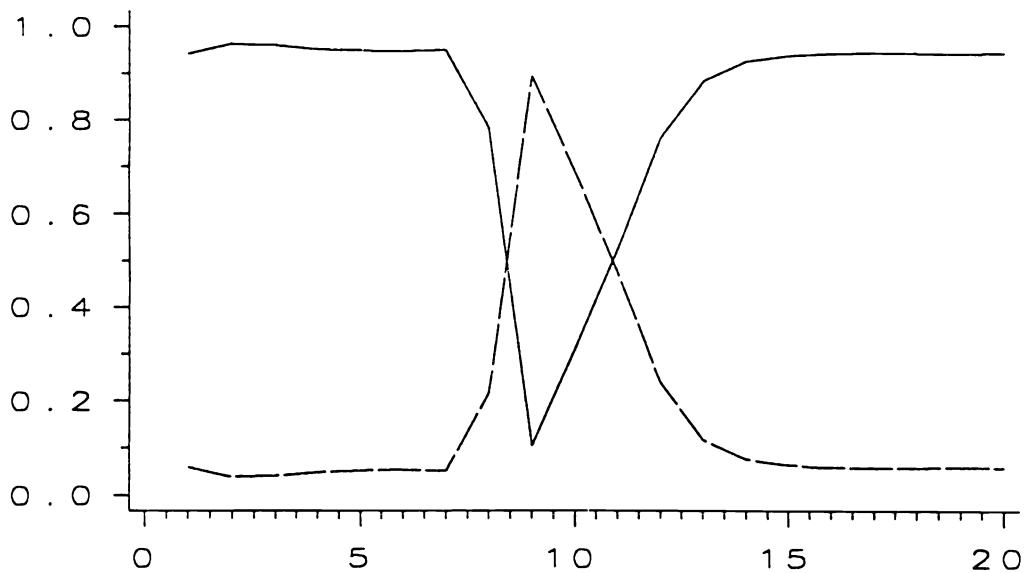


Figure 1.7.4-6: Mode probabilities in the IMM2: — Mode 1 (\mathcal{U}); - - Mode 2 (T).

Remarks

A **noise level switching** version of the KF (see [BL93] Section 11.2) with one model at a time:

- running a filter with a low variance process noise during the uniform motion portion of the trajectory and,
- following a maneuver detection based on a logic that uses the innovation magnitude⁷, switching to a filter with a high level process noise.

could be considered for this problem. While simpler, this is not practical due to the very short duration of the maneuver and the possibility of extraneous returns or noise that could trigger the logic.

Since the maneuver is not a *zero-mean white random sequence*, as the standard KF assumptions require, one cannot expect any filter to be perfectly consistent.

The **selection of the best design** is somewhat subjective, taking into account

- the performance (RMS errors) in both position and velocity: the trade-off between the *peak error* and the *average error* during the uniform (straight and level) motion,
- the timeliness of the maneuver detection,
- one's desire for the filter to be consistent,
- the complexity of the implementation.

Based on the above considerations, the IMM2 seems to be the best choice.

Sensitivity of the IMM to the Design Parameters

A large ratio of the process noise variances among the models is needed: otherwise the models cannot be distinguished (if both are the same, the IMM becomes a disguised KF). In the design illustrated above, this ratio was $6^2/0.3^2 = 400$. Experience indicates that the minimum value of this ratio is 10.

Values of the maneuver onset probability p_{12} between 0.05 and 0.15 yield similar results. Overall, the sensitivity of the performance to such variations in the transition probabilities is relatively small. A lower level process noise than the one chosen for mode \mathcal{U} would reduce further the errors when the target is not maneuvering, but it would degrade the performance around the maneuver time, especially at its end.

⁷Such a logic is sometimes called, in view of its reliability, “spaghetti logic”.

1.7.5 IMM Estimator Design — Summary

1.7.5 IMM Estimator Design — Summary

The design of an IMM estimator consists of the following:

- selection of the models for the various modes of behavior of the system
- selection of the Markov chain transition probabilities among the modes
- selection of the parameters of the various models — typically, the process noise levels.

A relatively small number of models can cover a wide variety of target behaviors.

The process noise levels are selected based on the expected disturbances and target maneuver magnitudes.

The target maneuvering index — ratio of RMS motion uncertainty to RMS observation uncertainty — is a key variable in this case.

The Markov chain transition probabilities are related to the expected sojourn time in the various modes.

The sensitivity of the performance of the IMM estimator w.r.t. the transition probabilities is relatively small.

1.8 ADAPTIVE BEAM POINTING CONTROL OF A PHASED ARRAY RADAR WITH AN IMM ESTIMATOR

1.8.1 The Radar Beam Pointing Problem

The benchmark problem for tracking of highly maneuvering targets proposed in [BWH94] is considered in this section. The problem involves beam pointing control of a phased array monopulse radar to track a variety of maneuvering targets that are moving in a 3-dimensional space and exhibit up to $7g$ of lateral acceleration and $2g$ of longitudinal acceleration as well as fluctuations in cross-section according to a Swerling 3 type model. The radar to target range is 20km to 150km.

The Radar

An amplitude-comparison monopulse radar with a phased array antenna is modeled (see [BWH94] for details of the radar signal processor model). The radar outputs are: the azimuth and elevation difference channel signals, the sum channel signal, the signal to noise ratio (SNR) at the sum channel and the target range. If the radar beam, whose (3dB) width is 1.7° at broadside, does not point exactly at the target, the SNR is lower and this has the following consequences:

- the target detection probability decreases
- the angle measurement errors increase

which can clearly lead to loss of track.

Pointing of the Radar Beam

The problem requires the design of a tracking algorithm to control the beam pointing of the radar, i.e., pointing the radar beam to the predicted position of the target at the next dwell (sampling) time. The next dwell time also has to be selected by the algorithm. Accurate pointing to the target is vital in a radar with a narrow beamwidth.

1.8.1 The Radar Beam Pointing Problem

Performance Criteria

The evaluation criteria for the design are:

- the average number of dwells per track while the percentage of the track loss is less than a certain value (4%);
- the computations required to implement the algorithm and
- the state estimate average error.

Scenarios for Evaluation

The six scenarios considered are

1. An incoming antiship missile with a constant speed of 300–335m/s which drops and performs $7g$ weaves
2. A commercial aircraft flying a racetrack pattern with a speed of 200kts (103m/s) and turning with $3^\circ/\text{s}$
3. A manned aircraft with longitudinal acceleration, a high pitch angle (for launching a weapon) and an evasive maneuver
4. A dogfight yo-yo maneuver and a $7g$ S-turn
5. Maneuvers including longitudinal acceleration, climbing and downward pitching with lateral turns
6. Similar to scenario 5, with different order and magnitudes of the maneuvers.

Only the first four scenarios were used for the design and all six for evaluation.

Figure 1.8.1-1 illustrates (partially) the scenarios considered, by showing the magnitudes of the accelerations vs. time. Most of the turns were coordinated turns, i.e., with constant acceleration magnitude; however, the components of the acceleration are changing sinusoidally in this case, i.e., they are far from being constant. The speed is constant in such a turn but varies when there is longitudinal acceleration. For more details, see [BWH94, Bla94].

1.8.1 The Radar Beam Pointing Problem

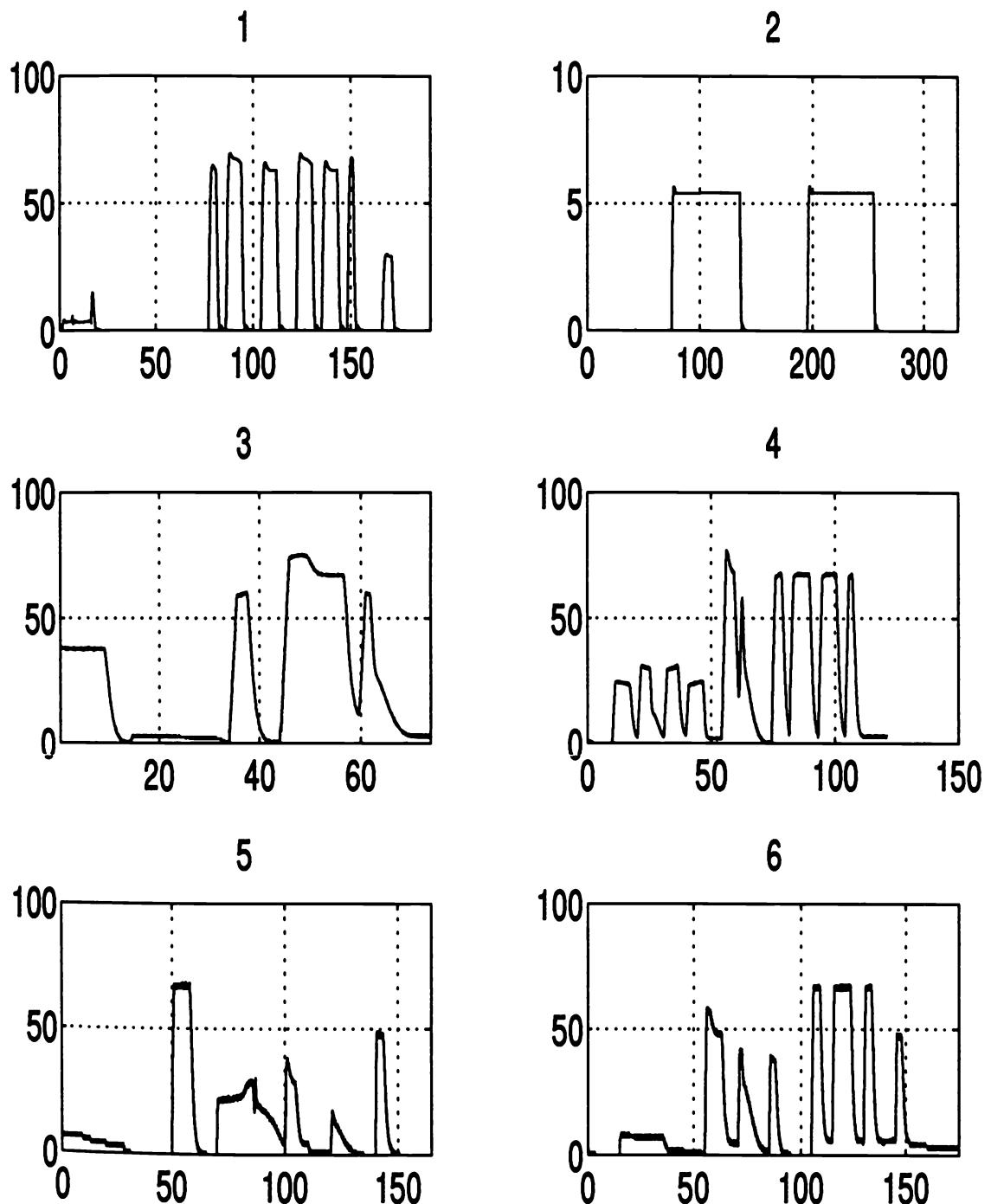


Figure 1.8.1-1: The acceleration magnitudes (m/s^2) for the scenarios considered.

1.8.2 The Approach to Beam Pointing Control

The evolution of the target is modeled in Cartesian coordinates and considered as a hybrid dynamic system with various modes of operation which differ in their structure/order as well as in the level of the process noise. Three models are considered:

1. A nearly constant velocity motion model — second order per coordinate with a low level white noise acceleration
2. Same model as above but with a high level of noise for an on-going maneuver
3. A Wiener process acceleration model, for the situations when the target starts or terminates a maneuver (large acceleration increments).

The tracking algorithm is an Interacting Multiple Model (IMM) estimator with three Kalman filters based on the above three models of the dynamic system.

At each time step t_k the measured values of elevation $\theta_e(k)$ and azimuth $\theta_a(k)$ are determined by a signal processor based on the measured elevation and azimuth difference channel signals $d_e(k)$, $d_a(k)$ and the sum channel signal $s(k)$ of the phased array radar. These values, together with the range $r(k)$, are the set of measurements for the tracking algorithm. The variances corresponding to these values are also calculated by knowing the value of the signal-to-noise ratio $\text{SNR}(k)$.

The set of measurements is converted from (the radar's) spherical coordinates to Cartesian coordinates consistent with the state. By this, the measurement equations become linear and the use of extended Kalman filters in the estimator is eliminated. For the trajectories considered and the radar measurement accuracies in this problem, the bias in the converted measurement is negligible.

The algorithm also selects the revisit (i.e., the next dwell) time for the target. The interval to the next dwell is selected from a predetermined set of values as the largest interval such that the predictions of the elevation and azimuth angle measurement errors are smaller than a certain selected threshold related to the beamwidth.

Beam pointing control is done based on the prediction of the azimuth and elevation angles of the target at the selected revisit time and pointing the beam to this direction. The range gate of the radar is centered at the predicted range of the target at the revisit time. The pointing accuracy (how much the target is off the beam center) is accounted for in the next measurement since it affects the SNR and the angle measurement accuracies depend on the SNR.

1.8.3 IMM Estimator Design for Variable Sampling Interval

1.8.3 IMM Estimator Design for Variable Sampling Interval

Following the selection of the motion models (e.g., white noise acceleration, Wiener process acceleration), the design of an IMM estimator requires selection of the parameters discussed below. The selection of these parameters should (and will) account for the **variable sampling interval**.

The Transition Probability Matrix

The evolution of the hybrid system among the modes is modeled as a first order Markov chain. The transition probability matrix of this chain is designed based on the expected sojourn time in each mode of the hybrid system, i.e.,

$$\tau_i = \frac{T}{1 - p_{ii}} \quad (1.8.3-1)$$

where τ_i is the expected sojourn time of mode i , p_{ii} is the probability of transition from mode i to itself and T is the sampling interval. Thus,

$$p_{ii} = 1 - \frac{T}{\tau_i} \quad (1.8.3-2)$$

The above equation, valid for small values of $\frac{T}{\tau_i}$, may lead to unacceptable values for p_{ii} as $\frac{T}{\tau_i}$ approaches to (or exceeds) 1. Therefore, a lower limit for p_{ii} has been used, i.e.,

$$p_{ii} = \max\{l_i, 1 - \frac{T}{\tau_i}\} \quad (1.8.3-3)$$

where l_i is the lower limit for mode i .

The expected sojourn times which are determined based on the available trajectories as well as the transition probability matrix are given in Subsection 1.8.6.

The transition probabilities p_{ij} for $i \neq j$ are selected using the identity

$$\sum_{j \neq i} p_{ij} = 1 - p_{ii} \quad (1.8.3-4)$$

and “apportioning” the probability mass $1 - p_{ii}$ to the various possible jumps according to the designer’s intuition.

1.8.3 IMM Estimator Design for Variable Sampling Interval

The Process Noise Variances

For each mode of operation, the process noise level is determined based on the maximum acceleration (or acceleration increment) corresponding to that mode.

For a second order dynamic model (white noise acceleration), the standard deviation of the process noise should be of the order of the maximum acceleration

$$\sigma_v = \alpha_2 a_{\max} \quad (1.8.3-5)$$

where σ_v is the process noise standard deviation, a_{\max} is the maximum acceleration and $0 < \alpha_2 \leq 1$.

For a third order dynamic model (Wiener process acceleration), the maximum acceleration increment during the sampling interval is the determining factor. Consequently,

$$\sigma_v = \min\{\alpha_3 \dot{a}_{\max} T, a_{\max}\} \quad (1.8.3-6)$$

where \dot{a}_{\max} is the maximum acceleration increment per unit time, T is the sampling interval and $0 < \alpha_3 \leq 1$. Obviously, the increment in the acceleration is always smaller than the maximum acceleration.

The process noise variances for all models are given in Subsection 1.8.6.

The above summarizes the **variable sampling interval estimator design procedure**.

1.8.4 The Adaptive Sampling Policy

1.8.4 The Adaptive Sampling Policy

Since the state prediction is less reliable during a maneuver than during a benign motion, the need for accurate measurements during maneuvers is more crucial than during periods of benign motion. Therefore, larger sampling intervals can be allowed during benign periods and shorter sampling intervals are needed during maneuvering periods. This requires an *adaptive policy to determine the sampling intervals based on the accuracy of the predictions.*

A predetermined set of sampling intervals is used, with the selection done based on the prediction of the radar angle innovation standard deviations compared to the radar beamwidth.

After each state update with a measurement, the following steps are carried out:

1. The estimator predicts the innovation covariance for the next measurement assuming the largest possible sampling interval.
2. This prediction covariance is transformed from Cartesian to spherical coordinates to obtain the azimuth and elevation angle innovation variances. The uncertainty caused by the fluctuations of the target cross-section is also considered in the calculations of the variances [She84].
3. The standard deviations of both angle innovations are compared with a certain fraction of the antenna beamwidth.
4. If any of the azimuth or elevation angle innovation standard deviations exceeds this threshold, the candidate sampling interval is rejected and the test is repeated for the next largest sampling interval.

1.8.5 The Algorithm for Beam Pointing Control

The major steps of the estimation algorithm are described next. The algorithm starts with a two point initialization of the state [BL93]. The initial estimate of the acceleration (needed for the third order model only) is zero. Sampling for initialization is done with the maximum allowable rate (10Hz). This rate lasts for a total of four steps to assure a sufficient decrease of the initial errors.

One cycle of the algorithm consists of the following:

1. The radar is pointed to the position predicted from the previous time and receives the returned signals.
2. If the target is detected, the range, azimuth and elevation angles and their variances are calculated as functions of the azimuth and elevation difference signals, the sum signal and the SNR. Assuming a **range resolution cell** of C_r , and a uniform distribution of the true target location in the cell that is “lit”, the variance of the range measurement is constant and equal to $(C_r)^2/12$. This cell was assumed to be $C_r = 50\text{m}$.
3. The expression (1.6.3-7) is calculated with the azimuth and elevation angle standard errors to assure that the coordinate transformation will not cause any bias.
4. The measurement and its covariance matrix are transformed from spherical to Cartesian coordinates.
5. This converted measurement is used in each mode-matched Kalman filter of the IMM estimator to update the state predictions from the previous time step and to calculate the likelihood functions of the modes. In the case that the measurement attempt fails, the previous state prediction of each mode is considered as the new estimate and the likelihood functions are all set to a constant.
6. The new probability of each mode is calculated and the updated estimates are combined to yield the final estimate. The covariances corresponding to each mode are also combined to yield the final state estimate covariance matrix.

1.8.5 The Algorithm for Beam Pointing Control

7. In order to select the next sampling interval, the predetermined possible values are examined in descending order:

- 1) If the last measurement has not yielded a detection, the minimum possible value is selected as the next sampling interval (0.1s).
- 2) Otherwise, the largest unexamined sampling interval is tried.
- 3) The probability transition matrix and the process noise variances (for all modes) are calculated for this sampling time.
- 4) The mixing of the estimates for all modes is done to calculate the (current cycle) initial values of the state and its covariance for each filter module of the IMM estimator.
- 5) In each mode-matched filter of the IMM, the state prediction, its covariance, the Kalman gain and the innovation covariance are calculated.
- 6) The mode probabilities at the next time step are predicted via the propagation of the current mode probabilities through the probability transition matrix.
- 7) The innovation covariances are combined via the predicted mode probabilities.
- 8) The final innovation covariance is transformed to spherical coordinates and the standard deviations of the azimuth and elevation angle innovations are compared with a threshold as described in Section 1.8.4.

If any of the angle standard deviations exceeds the selected threshold, the above procedure is repeated for the next (lower) sampling interval candidate.

8. After the sampling interval is determined, the position of the target is predicted and used to point the beam.

1.8.6 Simulation Results

1.8.6.1 Simulation Results

The IMM estimation algorithm described above has been run for the six trajectories and the results are shown in Table 1.8.6-1 and Figs. 1.8.6-1 through 1.8.6-3.

A (single-model based) Kalman filter — KF — with a second order model per coordinate was used as a baseline and the results are compared with the results of the IMM estimator.

Based on the first four trajectories, the maximum acceleration is $a_{\max} = 70 \text{m/s}^2$ and the maximum acceleration rate (jerk) as $\dot{a}_{\max} = 60 \text{m/s}^3$. Based on Eqs. (1.8.3-5) and (1.8.3-6) the standard deviation of the process noise for each model is selected as

$$\begin{aligned}\sigma_{v_1} &= 2 \text{m/s}^2 && \text{(for non-maneuvering intervals).} \\ \sigma_{v_2} &= 30 \text{m/s}^2 && \text{(for maneuvering intervals).} \\ \sigma_{v_3} &= \min\{30T, 70\} && \text{(for maneuver start/termination).}\end{aligned}$$

For the KF, a 50 m/s^2 process noise standard deviation is used.

The average sojourn times for the three modes are selected as

$$\tau_1 = 20 \text{s}, \tau_2 = 4 \text{s}, \tau_3 = 1 \text{s}$$

and the probability transition matrix of the Markov chain as

$$\begin{aligned}p_{11} &= 1 - 0.05T & p_{12} &= 0.05(1 - p_{11}) & p_{13} &= 0.95(1 - p_{11}) \\ p_{22} &= 1 - 0.25T & p_{21} &= 0.05(1 - p_{22}) & p_{23} &= 0.95(1 - p_{22}) \\ p_{33} &= \max\{1 - T, 0.1\} & p_{31} &= 0.33(1 - p_{33}) & p_{32} &= 0.67(1 - p_{33})\end{aligned}$$

For the adaptive sampling interval, a set of predetermined sampling intervals is considered and for each sampling interval, the azimuth and the elevation angle innovations are compared with the antenna beamwidth, which is 30 mrad (at the broadside direction). Empirically, 3.3 mrad was found as a good threshold for the angle innovation standard deviations. If any of the azimuth or elevation angle innovation standard deviations exceeded this threshold, the candidate sampling interval was rejected and the test repeated for the next largest sampling time.

The following set of sampling intervals was used: $\{0.1, 1, 2, 2.5, 3\}$. The 0.1s sampling interval is used for the first 4 steps at the beginning of the trajectory and for **revisit**⁸ at the times that the target is not detected, i.e., the last measurement attempt has failed.

⁸Practically all existing systems lack this revisit feature.

1.8.6 Simulation Results

Track Loss

Table 1.8.6-1 shows the results of 200 Monte Carlo simulations for the IMM estimator with the above sampling set. Table 1.8.6-2 corresponds to the (single-model) KF with the same adaptive sampling set.

It is seen that the track loss percentage for the KF is completely unsatisfactory (4% is the maximum allowed). As a comparison, Table 1.8.6-3 shows the results of a KF with constant sampling interval of 1.5s with revisit (if the target is not detected) after 0.1s. For sampling interval of 1s without revisit, the KF has 1–8% loss rate for these scenarios, i.e., *worse* than the KF with period 1.5s and revisit (which averages to 1.4s).

The results of these tables reveal that the IMM estimator in conjunction with the adaptive sampling policy yields satisfactory results in terms of track loss percentage and average dwells per run.

A (single-model) KF with the same adaptive sampling policy does not yield satisfactory results. Even in comparison with the KF with 1s constant sampling, the IMM with the average sampling interval of about 2s has less track loss.

Computational Requirements

The average number of **floating point operations (Flops)** per dwell (i.e., per sampling period) and the Flops per second are also shown in the tables. While these numbers are not very accurate (because the algorithms were not coded optimally), they show that the IMM operations per second (with adaptive sampling) increase by about a factor of 3–9 compared to the 1.5s rate KF (with revisit).

For the highly maneuvering trajectories the adaptive sampling part of the IMM was about 50–60% of the total.

1.8.6 Simulation Results

Estimation Errors

Another advantage of the IMM estimator is the noise reduction in the state estimate, especially in the velocity components of the state. The Monte Carlo based **RMS errors (RMSE)** of the position and the velocity components for all trajectories are shown in Figs. 1.8.6-1 and 1.8.6-2, respectively. The errors for both the IMM estimator (with adaptive sampling) and the Kalman filter (with 1.5s sampling and revisit) are plotted in the same frame. The RMSE is calculated based on the following equation⁹

$$\epsilon(t_k) = \sqrt{\frac{1}{N_k} \sum_{i=1}^{N_k} \sum_j [x_j^i(t_k) - \hat{x}_j^i(t_k)]^2} \quad (1.8.6-1)$$

where x is the true value of the state, \hat{x} is the estimated value, the superscript i stands for the Monte Carlo run index, subscript j corresponds to the state components (3 position or 3 velocity), t_k is the k -th sampling time and N_k is the total number of times that the state was estimated at time t_k during the Monte Carlo runs. Since in the IMM, the dwell times are changing at each run, sometimes N_k is not large enough. Therefore, the squared errors were also averaged over time intervals of 1s.

It can be seen from these figures that during the non-maneuvering intervals, the IMM has a significant error reduction compared with the Kalman filter, especially in the velocity component. During the maneuvering intervals the performance of the IMM is almost the same as Kalman filter's (which is running at a higher rate); however, sometimes the errors are larger, because the IMM suffers from larger measurement errors due to longer sampling intervals. This is the price that has to be paid to achieve longer sampling intervals.

The sampling intervals (averaged over 200 Monte Carlo runs as well as 1s intervals) versus the trajectory times are shown in Fig. 1.8.6-3 for all six trajectories (with the IMM estimator). The adaptability of the estimator is clearly seen from these plots. During the non-maneuvering periods the estimator selects the larger intervals while during a maneuver, shorter intervals are preferred.

⁹Note that, according to the RMS definition, the errors have to be first squared, then averaged, and then the square root is taken.

1.8.6 Simulation Results

Trajectory	1	2	3	4	5	6
Total Time	191	330	74	121	165	175
Dwells	96.2	117.9	39.8	50.3	81.8	87.7
Ave. Samp. Int.	1.99	2.80	1.85	2.40	2.02	1.99
Track Loss%	3	0	1	1.5	0.5	1.5
Flops/Dwell (k)	30	11.5	29.7	20.2	30.6	30.5
Flops/s (k)	15.7	4.2	17.6	9.1	15.7	16
Detection%	93.4	94.4	93.3	93.7	93.9	93.7
Position RMSE (m)	56.3	117.5	80.6	166.3	53.6	59.2
Velocity RMSE (m/s)	43.5	30.3	62.7	81.9	35.4	43.1

Table 1.8.6-1: Performance of the IMM estimator with sampling set {0.1,1,2,2.5,3}.

Trajectory	1	2	3	4	5	6
Total Time	191	330	74	121	165	175
Dwells	90.5	118.0	33.4	44.9	74.4	75.1
Ave. Samp. Int.	2.11	2.79	2.21	2.69	2.22	2.33
Track Loss%	54	17	21	24	35	51
Flops/Dwell (k)	3.4	3.5	3.2	3.3	3.3	3.4
Flops/s (k)	1.7	1.3	1.6	1.3	1.6	1.5
Detection%	87	92.7	87.5	90.0	88.6	86.5
Position RMSE (m)	89.8	189.6	125.8	233	83.9	94.5
Velocity RMSE (m/s)	91.5	108.3	106.8	132.2	89.6	96.7

Table 1.8.6-2: Performance of the Kalman filter with sampling set {0.1,1,2,2.5,3}.

Trajectory	1	2	3	4	5	6
Total Time	191	330	74	121	165	175
Dwells	135.4	232.5	52.5	84.8	117.1	123.7
Samp. Int.	1.41	1.42	1.41	1.43	1.41	1.41
Track Loss%	3	1	2	3	2	1
Flops/Dwell (k)	3.5	3.5	3.3	3.5	3.5	3.5
Flops/s (k)	2.5	2.5	2.5	2.6	2.6	2.5
Detection	93.7	94.3	93.6	93.8	93.8	93.9
Position RMSE (m)	62.0	139.5	85.2	149.5	60.0	61.0
Velocity RMSE (m/s)	62.0	68.8	73.9	86.3	59.3	60.1

Table 1.8.6-3: Performance of the Kalman filter with 1.5s period and revisit.

1.8.6 Simulation Results

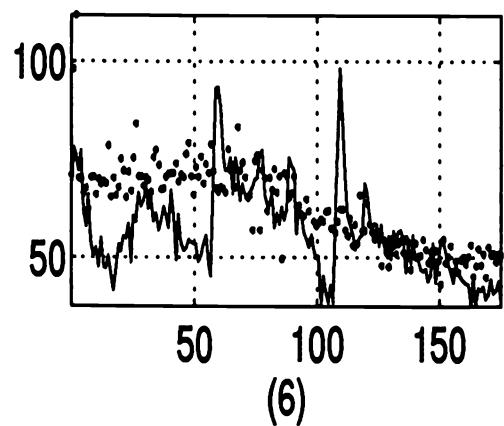
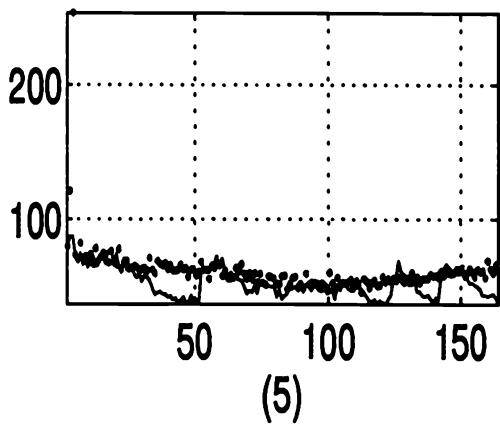
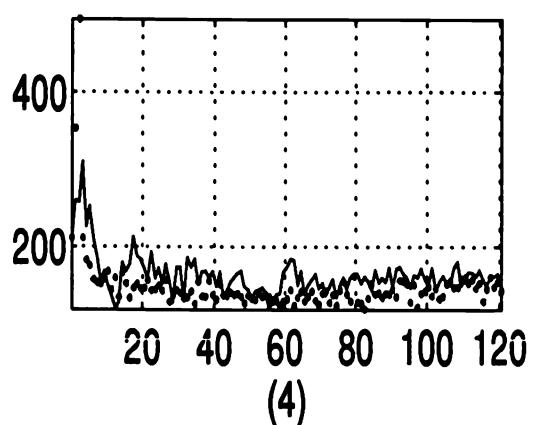
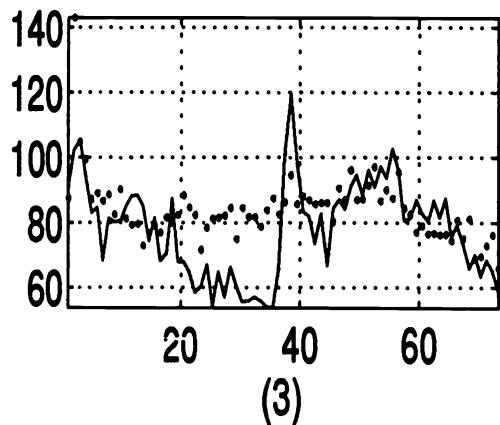
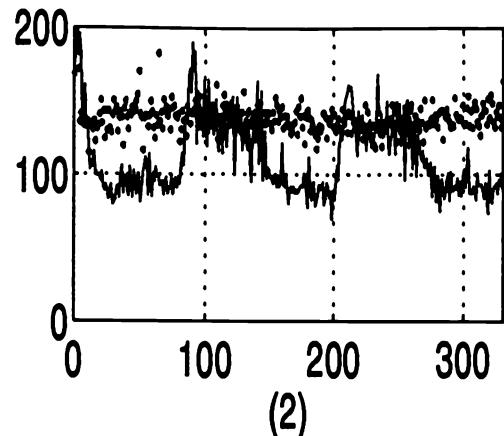
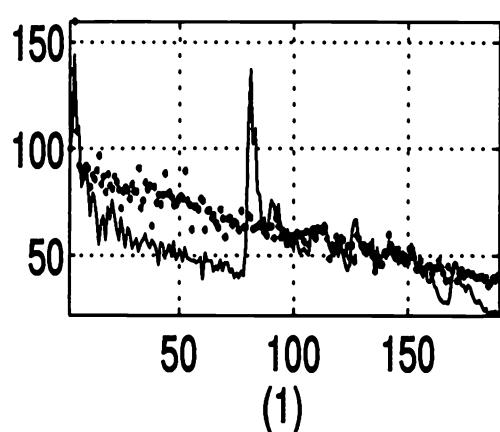


Figure 1.8.6-1: The position RMSE (three coordinates combined) of the six trajectories. Averaged over 1s intervals and over 200 Monte Carlo runs. Solid: IMM with adaptive sampling policy. Dashed: Kalman filter with 1.5s sampling with revisit.

1.8.6 Simulation Results

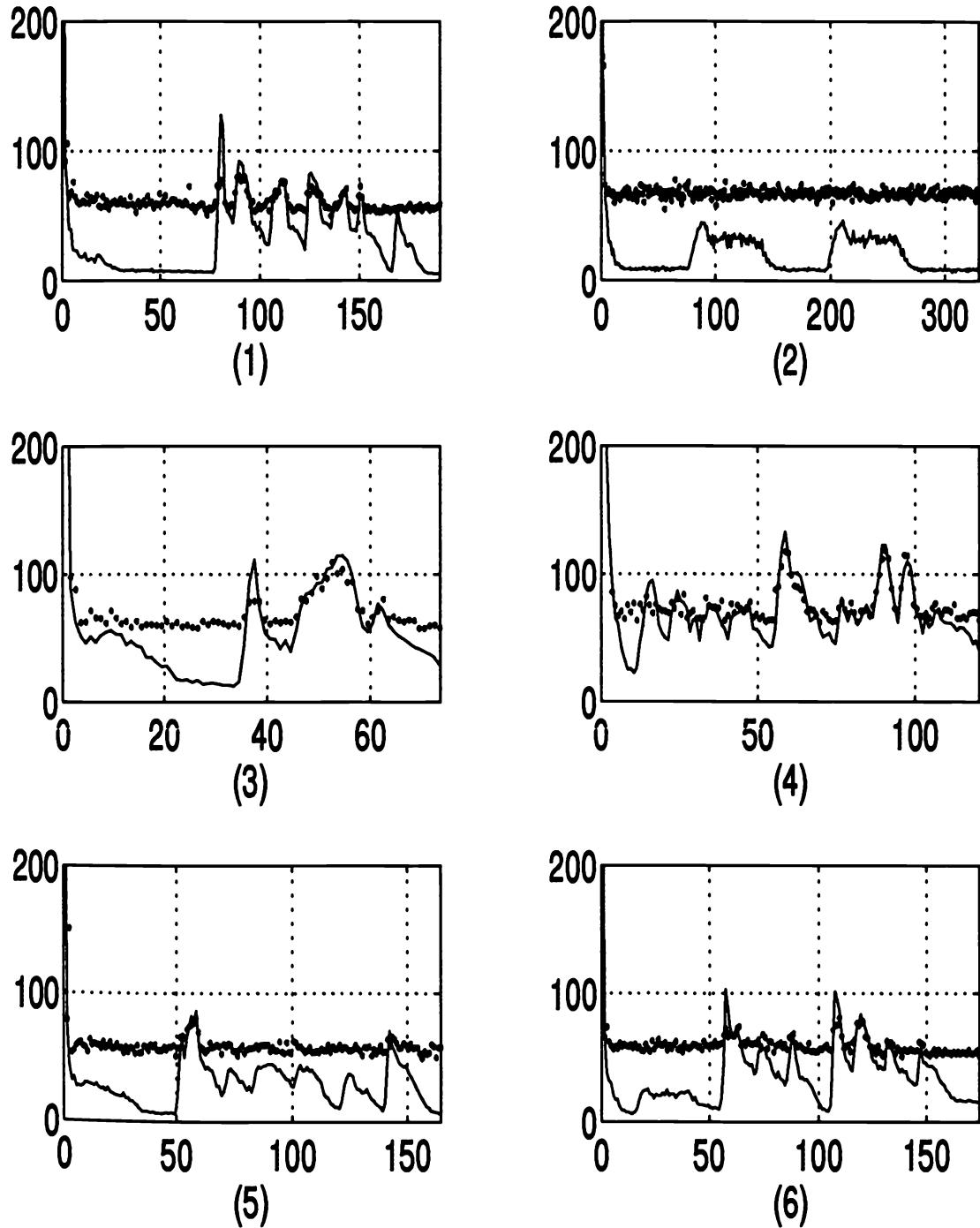


Figure 1.8.6-2: The velocity RMSE (three coordinates combined) of the six trajectories. Averaged over 1s intervals and over 200 Monte Carlo runs. Solid: IMM with adaptive sampling policy. Dashed: Kalman filter with 1.5s sampling with revisit.

1.8.6 Simulation Results

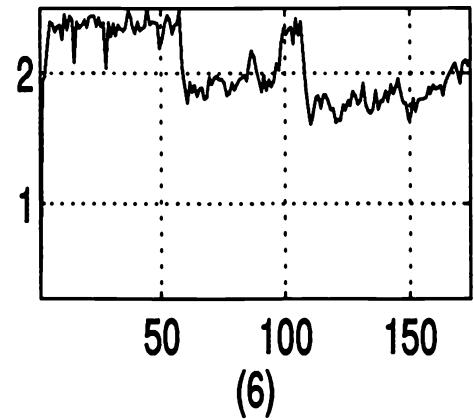
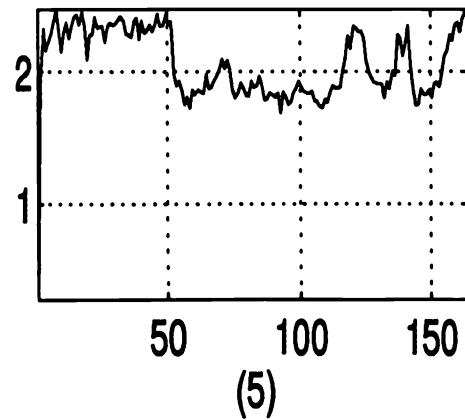
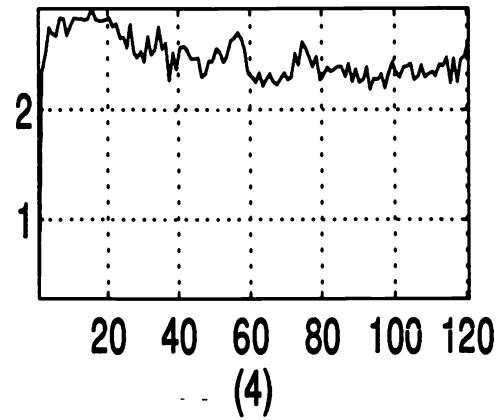
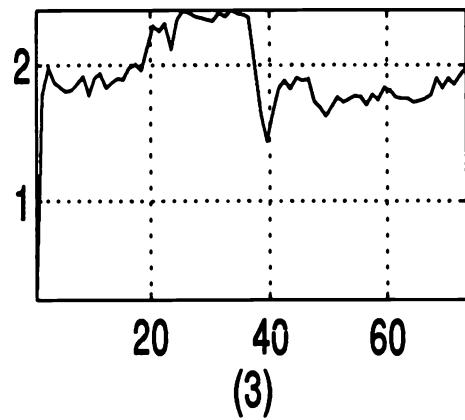
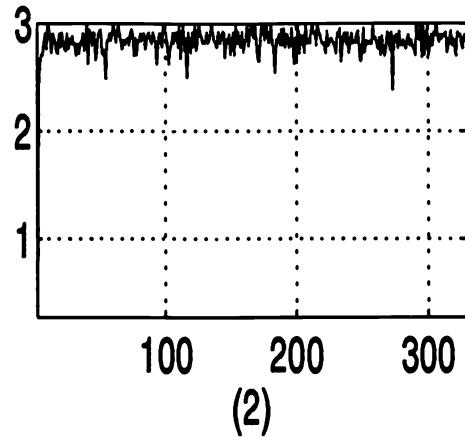
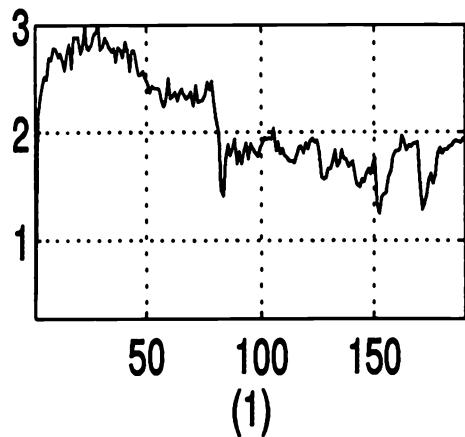


Figure 1.8.6-3: The IMM sampling interval average vs. time for the six trajectories. Averaged over 1s intervals and 200 Monte Carlo runs.

1.8.7 Radar Beam Control — Summary

1.8.7 Radar Beam Control — Summary

An IMM estimator with an adaptive policy for real-time selection of the sampling interval for a phased array radar has been designed and illustrated.

This algorithm allows the use of larger revisit intervals while keeping the track loss percentage below a specified level.

It also improves the tracking performance in terms of the state estimation errors compared with the Kalman filter, while running at a *lower rate*, i.e., saving

- radar energy
- signal processing computations.

The computational cost of the IMM vs. the KF is, per unit time, about 3-7 times higher.

The performance of the IMM estimator can be further improved by fine-tuning of the design parameters.

The design procedure for an IMM estimator with *variable sampling interval*, needed in the adaptive sampling situation, has also been presented in detail and illustrated.

1.9 BIBLIOGRAPHICAL NOTES

The estimation theory background for this text can be found in the companion text [BL93] or, in an abbreviated manner, in the first part of [BF88]. The monograph [BF88] also covers about half the material of the present text.

The pioneering work on multitarget tracking was done by Sittler [Sit64]. This work was motivated by the need to find a reasonable way to incorporate measurements of uncertain origin into existing tracks. Earlier, Wax [Wax55] carried out an analysis of tracks as birth-death processes generated by some fixed probabilistic models but did not consider association ambiguities. These works preceded the introduction of the recursive state estimation algorithm by Kalman and Bucy [Kal60, KB61]. The tracking algorithm from [Sit64] was still of the type used prior to the development of what became known as “Kalman filtering.”

It was around 1970 that the systematic treatment of tracking targets in the presence of false alarms using Kalman filtering techniques has started with the work of [SS71, Sea71]. Following this, a large number of papers and reports have been published. An extensive and probably exhaustive list of references on this topic up to about 1979 can be found in the Naval Research Laboratory Handbook [WWGK79, GWW80].

The monograph [Bla86] covers a significant amount of material on tracking of multiple targets and discusses radar applications. Tracking and radar signal processing are discussed in [FS85]. Additional relevant material can be found in [DB88], [DBP90].

Several conferences (IEEE Conf. on Decision and Control, American Control Conference) have been organizing special sessions dedicated to recent results in multitarget tracking since 1980. The SDIO Panel on Tracking (within IDA, Alexandria, VA) published a number of Proceedings volumes in the late 80’s and early 90’s. The SPIE Conferences on Data Processing for Small Targets (since 1989) have been the largest meetings with papers on tracking. Regularly held short courses at UCLA Extension (since 1985), organized by the first author of this text, have been a forum for the presentation of practical advances in this field.

Section 1.6 is based on [LB93b].

A similar approach to the one in Section 1.7 to the ATC tracking problem, but with coordinated turn models, is discussed in [LB93c].

An approach to the radar beam pointing with different motion models than those of Section 1.8 is presented in [Bla94]. The exponentially autocorrelated acceleration model (with increasing acceleration) is discussed in [BWR91]. The modeling of constant speed turns via a kinematic constraint is discussed in [BWA91].

Chapter 2

MULTIPLE SENSORS, CLUTTER AND MULTIPLE TARGETS

2.1 INTRODUCTION

2.1.1 Outline

This chapter discusses the situations where a tracking filter receives measurements from **multiple sensors** and possibly more than one measurement from a sensor. In the latter case one can face an **origin uncertainty**: from which target, if any, a particular measurement originated.

Section 2.2 presents the sequential and parallel procedures to incorporate into a tracking filter the measurements from various sensors if there is no origin ambiguity. These procedures assume that a **central filter** processes all the measurements. Decentralized configurations are discussed in Chapter 8.

The general **association procedure** (or **gating**) of measurements is discussed in Section 2.3.

The problem of extraneous measurements due to random phenomena (false alarms or clutter) as well as neighboring targets is presented in Section 2.4.

A clutter model that yields the pmf of the number of “clutter points” and the pdf of the location of the corresponding measurements is presented in Section 2.4. Track initiation in clutter and a logic-based algorithm for track formation are discussed in Section 2.5 together with the performance evaluation of this algorithm.

2.1.2 Summary of Objectives

2.1.2 Summary of Objectives

Present procedures to incorporate measurements from several sensors into tracks using a centralized configuration — a central information processor receives all the measurements for filtering

Describe a procedure to handle out-of-sequence measurements, a common situation in multisensor tracking

Discuss the association of measurements to tracks via validation regions

Indicate the issues posed by false measurements in

- tracking a single target
- tracking several targets

Discuss track initiation (track formation) in clutter

Present an algorithm for track formation in clutter and its performance evaluation.

2.2 TRACKING WITH MULTIPLE SENSORS

2.2.1 The Multisensor Problem

The target of interest is described by the standard linear dynamic equation

$$x(k+1) = F(k)x(k) + v(k) \quad (2.2.1-1)$$

taken, for simplicity, without the control input.

Synchronous Case

It is assumed that there are N_S **synchronized sensors**. At a given sampling time there is one measurement from each sensor, with no origin uncertainty. The measurement from sensor j at time k is

$$z(k, j) = H(k, j)x(k) + w(k, j) \quad j = 1, \dots, N_S \quad (2.2.1-2)$$

The measurement noise sequences are zero-mean, white, independent of the process noise in (2.2.1-1) and *independent from sensor to sensor*, with covariances

$$E[w(k, j)w(l, i)'] = R(k, j)\delta_{ji}\delta_{kl} \quad (2.2.1-3)$$

Two techniques are described in the sequel for this situation:

- Sequential updating: the updating is carried out with the measurement of one sensor at a time — this is possible in view of (2.2.1-3).
- Parallel updating: the measurements from each sensor at the same time are stacked and a simultaneous update is carried out.

These two techniques, which are equivalent, are discussed in the sequel.

Asynchronous Case

If the measurements are obtained **asynchronously**¹, then the prediction and update are carried out according to the order in which the measurements were obtained. In this case the dynamic equation will have in general unequal sampling intervals and has to be set up accordingly (see Subsection 8.7.4). The case where the measurements arrive at the central processor **out of sequence** is discussed in Subsection 2.2.4.

¹Only in academic environments are multiple sensors synchronized. Also, limit theorems for the number of sensors going to infinity have only pure academic relevance.

2.2.2 Sequential Updating

2.2.2.2 Sequential Updating

The **sequential updating** across sensors for *synchronized measurements* is as follows.

Denote the predicted state at time k and its covariance as

$$\hat{x}(k|k, 0) \triangleq \hat{x}(k|k - 1) \quad (2.2.2-1)$$

$$P(k|k, 0) \triangleq P(k|k - 1) \quad (2.2.2-2)$$

The updates with the measurements (2.2.1-2) at time k are

$$\boxed{\hat{x}(k|k, j) = \hat{x}(k|k, j - 1) + W(k, j)\nu(k, j)} \quad j = 1, \dots, N_S \quad (2.2.2-3)$$

where

$$\nu(k, j) \triangleq z(k, j) - H(k, j)\hat{x}(k|k, j - 1) \quad (2.2.2-4)$$

$$S(k, j) = H(k, j)P(k|k, j - 1)H(k, j)' + R(k, j) \quad (2.2.2-5)$$

$$W(k, j) = P(k|k, j - 1)H(k, j)'S(k, j)^{-1} \quad (2.2.2-6)$$

$$\boxed{P(k|k, j) = P(k|k, j - 1) - W(k, j)S(k, j)W(k, j)'} \quad j = 1, \dots, N_S \quad (2.2.2-7)$$

This is possible in view of (2.2.1-3), which states that the measurements across *time and sensors* are a white sequence.

The final updated estimate and covariance at time k are

$$\hat{x}(k|k) = \hat{x}(k|k, N_S) \quad (2.2.2-8)$$

$$P(k|k) = P(k|k, N_S) \quad (2.2.2-9)$$

Order of Update

For linear measurements, the order of updating in the sequential procedure makes no difference. For nonlinear measurements, however, one should update first with the measurement from the most accurate sensor to reduce subsequent linearization errors. If the order of the update matters much, then the filter is probably marginal.

2.2.3 Parallel Updating

2.2.3.1 Parallel Updating

In the **parallel updating** procedure (for synchronized measurements) the state is updated simultaneously with the *stacked vector* of all the measurements

$$z(k) = \begin{bmatrix} z(k, 1) \\ \vdots \\ z(k, N_S) \end{bmatrix} = H(k)x(k) + w(k) \quad (2.2.3-1)$$

where

$$H(k) = \begin{bmatrix} H(k, 1) \\ \vdots \\ H(k, N_S) \end{bmatrix} \quad (2.2.3-2)$$

$$w(k) = \begin{bmatrix} w(k, 1) \\ \vdots \\ w(k, N_S) \end{bmatrix} \quad (2.2.3-3)$$

and

$$E[w(k)w(k)'] = R(k) = \text{diag}[R(k, j)] \quad (2.2.3-4)$$

is the *block-diagonal* covariance matrix of the stacked noise vector.

Using the filter gain expression [BL93] in terms of the updated covariance, the state update equation is

$$\hat{x}(k|k) = \hat{x}(k|k-1) + P(k|k)H(k)'R(k)^{-1}\nu(k) \quad (2.2.3-5)$$

where the stacked innovation is

$$\nu(k) = \begin{bmatrix} z(k, 1) - \hat{z}(k, 1|k-1) \\ \vdots \\ z(k, N_S) - \hat{z}(k, N_S|k-1) \end{bmatrix} = \begin{bmatrix} z(k, 1) - H(k, 1)\hat{x}(k|k-1) \\ \vdots \\ z(k, N_S) - H(k, N_S)\hat{x}(k|k-1) \end{bmatrix} \quad (2.2.3-6)$$

and $\hat{z}(k, j|k-1)$ denotes the predicted measurement for sensor j .

The updated state can be written, in view of the block-diagonal form of the matrix $R(k)$ given in (2.2.3-4), as

$$\hat{x}(k|k) = \hat{x}(k|k-1) + P(k|k) \sum_{j=1}^{N_S} H(k, j)'R(k, j)^{-1}\nu(k, j) \quad (2.2.3-7)$$

where

$$\nu(k, j) \triangleq z(k, j) - H(k, j)\hat{x}(k|k-1) \quad (2.2.3-8)$$

2.2.3 Parallel Updating

Similarly, the recursion for the inverse covariance has, from Subsection 5.2.3 of [BL93], the following simple form:

$$P(k|k)^{-1} = P(k|k-1)^{-1} + \sum_{j=1}^{N_S} H(k,j)'R(k,j)^{-1}H(k,j) \quad (2.2.3-9)$$

Remarks

This procedure is algebraically equivalent (for linear systems) to the sequential one, but computationally more expensive [WCD76].

While the notations of the innovations in (2.2.2-4) and (2.2.3-8) are the same, their definitions are different.

2.2.4 State Update with Out of Sequence Measurements

2.2.4 State Update with Out of Sequence Measurements

When the measurements from sensors arrive to the central processor in batches (e.g., corresponding to scans or frames — a common situation) it is quite likely to have an earlier measurement from a target arrive *after* a later one. This is the ***out of sequence measurements*** situation.

For example, assume that scan j of sensor 1 covers the time interval $[T_{1,j-1}, T_{1,j}]$ and scan l of sensor 2 corresponds to the interval $[T_{2,l-1}, T_{2,l}]$, where

$$T_{1,j-1} < T_{2,l-1} < T_{1,j} < T_{2,l} \quad (2.2.4-1)$$

i.e., they have a certain overlap.

Then, if the data from each sensor are sent to the central processor at the *end of each scan*, the measurements from scan l of sensor 2 will arrive after those of scan j of sensor 1. Now, if sensor 1 obtains a measurement from a target at t_1 near the end of its scan j sensor 2 obtains a measurement from the same target at t_2 near the beginning of its scan l , i.e.,

$$T_{1,j-1} < T_{2,l-1} < t_2 < t_1 < T_{1,j} < T_{2,l} \quad (2.2.4-2)$$

then the (earlier) measurement at t_2 will arrive to the information processor *after* the one at t_1 , i.e., *out of sequence*.

Thus one has to carry out what can be called a ***negative-time measurement update*** because one has a negative sampling interval

$$\delta_k = t_{k+1} - t_k < 0 \quad (2.2.4-3)$$

In the standard case, when $\delta_k > 0$, the sufficient statistic for prediction of the measurement is the latest state estimate $\hat{x}(k|k)$. In the present case, the (backward) “prediction” to t_{k+1} would require smoothing or “retrodiction” [BL93], which in turn, requires the availability of past measurements. In practice this is quite complicated since it also involves arbitrary times and accounting for the process noise in the state equation.

2.2.4 State Update with Out of Sequence Measurements

The Algorithm

A simple technique that can be used is the following. The state is “predicted” (actually, retrodicted) from t_k to t_{k+1} ignoring the process noise. For a linear system with transition matrix $F(t_{k+1}, t_k)$, with current state estimate $\hat{x}(k|k)$ and covariance $P(k|k)$, the (suboptimal) prediction is

$$\hat{x}(k+1|k) = F(t_{k+1}, t_k)\hat{x}(k|k) \quad (2.2.4-4)$$

$$P(k+1|k) = F(t_{k+1}, t_k)P(k|k)F(t_{k+1}, t_k)' \quad (2.2.4-5)$$

The goal is to update the state at time t_k using the measurement $z(k+1)$ at (the earlier) time t_{k+1} . This is done using the above “predicted” state as follows.

The innovation at t_{k+1} and its covariance are, with the standard notations,

$$\nu(k+1) = z(k+1) - H(k+1)\hat{x}(k+1|k) \quad (2.2.4-6)$$

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1) \quad (2.2.4-7)$$

The filter gain will be

$$W(k+1) = P(k+1|k)H(k+1)'S(k+1)^{-1} \quad (2.2.4-8)$$

it seems it's a typing error

and the new updated state at t_k (which now incorporates the measurement at t_{k+1}) and its covariance are

$$\hat{x}(k|k+1) = \hat{x}(k|k) + W(k+1)\nu(k+1) \quad (2.2.4-9)$$

$$P(k|k+1) = P(k|k) - W(k+1)S(k+1)W(k+1)' \quad (2.2.4-10)$$

A technique that relies on the (suboptimal) measurement prediction (2.2.4-5) but accounts for the process noise can be found in [HMB93].

2.2.5 Multisensor Tracking — Summary

2.2.5 Multisensor Tracking — Summary

When measurements are obtained from multiple sensors, there are several cases to be distinguished:

- synchronized sensors — the sampling times of all the sensors are the same
- asynchronous sensors — the sampling times are arbitrary (the realistic case).

If the sensors are synchronized and the measurement noises across the sensors are uncorrelated (a realistic assumption), then the two update schemes

- sequential — with the data from one sensor at a time
- parallel — simultaneously with a stacked measurement

are equivalent *for linear systems*, with the first being less expensive.

If the sensors are not synchronized, then one will (most certainly) encounter in a centralized filter the situation of *out of sequence measurements*. This requires special consideration in carrying out the update.

A suboptimal scheme for updating the state with an “old” measurement has been described.

In all these techniques it is assumed that there is no origin uncertainty for the measurements.

2.3 VALIDATION OF THE MEASUREMENTS

2.3.1 The Association Problem

In tracking applications, following the signal detection process that yields measurements, there is a procedure that *selects* the measurement(s) to be incorporated into the state estimator.

In a radar the “return” from the target of interest is sought within a time interval determined by the anticipated range of the target when it reflects the energy transmitted by the radar: a “range gate” is set up and the detection(s) within this gate can be *associated* with the target of interest.

In general the measurements have a higher dimension:

- range, azimuth (bearing), elevation or direction cosines for radar, possibly also range rate,
- bearing and frequency (when the signal is narrow-band) or time difference of arrival and frequency difference in passive sonar,
- two line of sight angles or direction cosines for optical sensors.

Then a *multidimensional gate* is set up for detecting the signal from the target. This is done to avoid searching for the signal from the target of interest in the entire measurement space.

A measurement in the gate, while not guaranteed to have originated from the target the gate pertains to, is a *valid association candidate* — thus the name **validation region** or **association region**.

If there is more than one detection (measurement) in the gate, this leads to an **association uncertainty**.

It will be assumed that one has **point measurements**, rather than distributed over several **resolution cells** of the sensor as in the case of an **extended target** or a **small image**, to be discussed in Chapter 9.

2.3.2 The Validation Region

In view of the variety of variables that can be measured, a generic gating (or validation or association) procedure for continuous-valued measurements is discussed.

Consider a target that is in track, i.e., its filter has been initialized. Then, according to Subsection 5.2.3 of [BL93] one has the predicted value (mean) of the measurement $\hat{z}(k+1|k)$ and the associated covariance $S(k+1)$.

ASSUMPTION: The true measurement conditioned on the past is **normally (Gaussian) distributed**² with its **probability density function (pdf)** given by

$$p[z(k+1)|Z^k] = \mathcal{N}[z(k+1); \hat{z}(k+1|k), S(k+1)] \quad (2.3.2-1)$$

Then the true measurement will be in the following region

$$\boxed{\mathcal{V}(k+1, \gamma) = \{z : [z - \hat{z}(k+1|k)]' S(k+1)^{-1} [z - \hat{z}(k+1|k)] \leq \gamma\}} \quad (2.3.2-2)$$

with probability determined by the **gate threshold** γ .

The region defined by (2.3.2-2) is called **gate** or **validation region** (hence the symbol \mathcal{V}) or **association region**. It is also known as the **ellipse (or ellipsoid) of probability concentration** — the region of *minimum volume* that contains a given probability mass. The semiaxes of the ellipsoid (2.3.2-2) are the square roots of the eigenvalues of γS .

The threshold γ is obtained from tables of the chi-square distribution since the quadratic form in (2.3.2-2) that defines the validation region is chi-square distributed with number of degrees of freedom equal to the dimension n_z of the measurement [BL93]. This quantity (metric), introduced as the **normalized innovation squared (NIS)** in Section 5.4 of [BL93], is also called “statistical distance squared” in the literature.

²The symbol $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, S)$ stands for the normal (Gaussian) pdf with argument the (vector) random variable \mathbf{x} , mean $\boldsymbol{\mu}$ and covariance matrix S .

2.3.2 The Validation Region

Table 2.3.2-1 gives the **gate probability**

$$P_G \triangleq P\{z(k+1) \in \mathcal{V}(k+1, \gamma)\} \quad (2.3.2-3)$$

or the “probability that the (true) measurement will fall in the gate” for various values γ and dimensions n_z of the measurement.

The square root $g = \sqrt{\gamma}$ is sometimes referred to as “number of sigmas” (standard deviations) of the gate. This, however, does not fully define the probability mass in the gate as can be seen from Table 2.3.2-1.

The **volume** V of the validation region \mathcal{V} from (2.3.2-2) corresponding to the threshold $\gamma = g^2$ (“ g -sigma” gate) is

$$V(k+1) = c_{n_z} |\gamma S(k+1)|^{1/2} = c_{n_z} g^{n_z} |S(k+1)|^{1/2} \quad (2.3.2-4)$$

where n_z is the dimension of the measurement and c_{n_z} is the volume of the unit hypersphere of this dimension ($c_1 = 2, c_2 = \pi, c_3 = 4\pi/3$, etc.). In general,

$$c_n = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)} \quad (2.3.2-5)$$

where $\Gamma(\cdot)$ is the gamma function.

γ	1	4	6.6	9	9.2	11.4	16	25
g	1	2	2.57	3	3.03	3.38	4	5
n_z								
1	.683	.954	.99	.997			.99994	1
2	.393	.865		.989	.99		.9997	1
3	.199	.739		.971		.99	.9989	.99998

Table 2.3.2-1: Gate thresholds and the probability mass P_G in the gate.

2.3.3 Measurement Validation — Summary

Selection of the measurement(s) to be incorporated into the tracking filter is done using a *validation* or *association region (gate)*.

The validation region is the region in the measurement space around the predicted measurement in which the *true measurement* of interest will lie, in view of all the uncertainties, with a certain (high) probability.

A measurement in the gate is a candidate for association to the corresponding track. Various ways of using these candidates will be discussed later.

The measurement vector should include as components all the variables that

- carry information about the state of the target
- help discriminate against undesirable measurements.

If the measurement of interest is Gaussian distributed about its predicted value then the validation region is the *ellipsoid of probability concentration* — the region of minimum volume that contains a given probability mass.

The validation region consists of the points whose *normalized distance* to the predicted measurement is below a threshold. This is equivalent to the *normalized innovation* being below this threshold.

The validation threshold for *Gaussian innovations* follows from the *chi-square* distribution tables with degrees of freedom equal to the dimension of the measurement vector.

2.4 TARGETS AND CLUTTER

2.4.1 A Single Target in Clutter

The validation procedure described in Section 2.3 limits the region in the measurement space where the information processor will “look” to find the measurement from the target of interest. In spite of this, it can happen that *more than one detection*, i.e., several measurements, will be found in the validation region.

Measurements *outside the validation region* can be ignored: they are “too far” and thus very unlikely to have originated from the target of interest. This holds if the gate probability is close to unity and *the model used to obtain the gate is correct*.

The problem of tracking a single target in clutter considers the situation where there are possibly several measurements in the validation region (gate) of a target. The set of **validated measurements** consists of:

- the correct measurement (if detected and it fell in the gate)
- the undesirable measurements: clutter or false-alarm originated.

In practice detections are obtained by thresholding the signal received by the sensor after processing it. This is the simplest (binary) way of using a target **feature** — its intensity. More sophisticated ways of using such feature information will be discussed later.

It is assumed that the measurement contains all the information that could be used to discard the undesirable measurements. Therefore, any measurement that has been validated could have originated from the target of interest.

2.4.1 A Single Target in Clutter

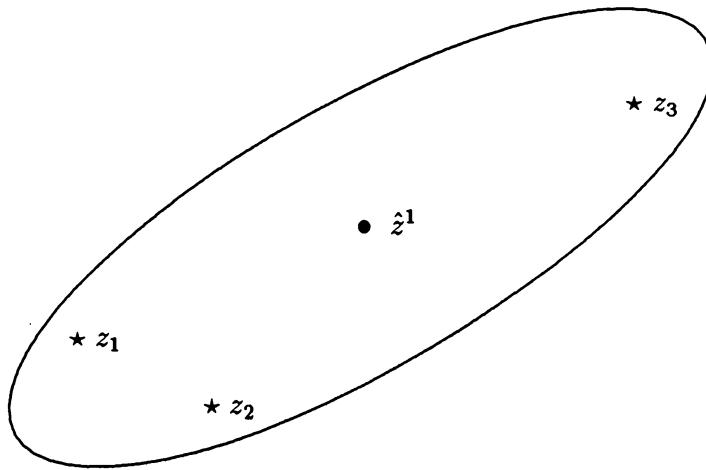


Figure 2.4.1-1: Several measurements in the validation region of a single target.

A situation with several validated measurements is depicted in Figure 2.4.1-1. The (two-dimensional) validation region is an ellipse centered at the predicted measurement $\hat{z}(k+1|k)$. The parameters of the ellipse are determined by the covariance matrix $S(k+1)$ of the innovation.

All the measurements in the validation region can be said to be *not too unlikely* to have originated from the target of interest, even though only one is assumed to be the true one.

The implication of the assumption that there is a single target is that the *undesirable measurements constitute a random interference*. The common mathematical model for such **false measurements** is that they are

- uniformly spatially distributed and
- independent across time.

This corresponds to what is known as **residual clutter** — the constant clutter, if any, has been removed.

2.4.2 Multiple Targets in Clutter

The situation where there are several targets in the same neighborhood as well as clutter (or false alarms) is more complicated.

Figure 2.4.2-1 illustrates such a case for a given time, with the predicted measurements for the two targets considered denoted as \hat{z}^1 and \hat{z}^2 .

In this figure the following measurement origins are possible:

- z_1 from target 1 or clutter
- z_2 from either target 1 or target 2 or clutter
- z_4 and z_5 from target 2 or clutter.

However, if z_2 originated from target 2 then it is quite likely that z_1 originated from target 1. This illustrates the *interdependence of the associations* in a situation where a **persistent interference** (neighboring target) is present in addition to random interference (clutter).

Up to this point it was assumed that a measurement could have originated from *one of the targets* or from *clutter*. However, in view of the fact that any signal processing system has an inherent **finite resolution** capability, an additional possibility has to be considered:

- z_2 could be the result of the **merging** of the detections from the two targets — it is an **unresolved measurement**.

This constitutes a fourth origin hypothesis for a measurement that lies in the intersection of two validation regions.

This illustrates only the difficulty of association of measurements to tracks *at one point in time*. The full problem, as will be discussed later, consists of associating measurements *across time*.

2.4.2 Multiple Targets in Clutter

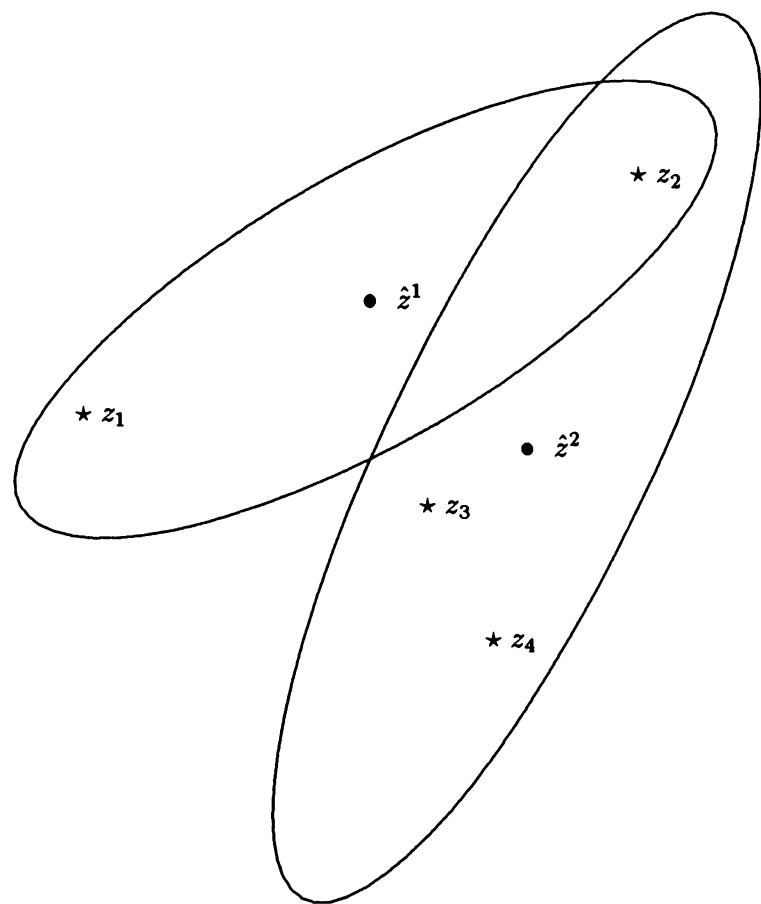


Figure 2.4.2-1: Two targets with a measurement in the intersection of their validation regions.

2.5 A CLUTTER MODEL

2.5.1 The Model for False Detections

Consider a sensor with N **resolution cells** (or **pixels**). A detection is declared in a cell if the output of the signal processor in this cell exceeds a certain threshold. If the sensor points to a region where there is no target, then detections will still occur due to sensor noise or background noise (spurious radiators/reflectors).

ASSUMPTIONS:

- The events of detection in each cell are independent of each other
- The probability of such a detection, which is a **false alarm**, is $P_{FA} = p$ in each cell.

Then, the **probability mass function (pmf)** of the number of false alarms in these N cells is given by the binomial (Bernoulli) distribution

$$P\{n_{FA} = m\} = \mu_{FA}(m) = \binom{N}{m} p^m (1-p)^{N-m} \quad (2.5.1-1)$$

Denoting by V the volume of the N cells under consideration, one has the **spatial density of the false alarms** as

$$\lambda = \frac{E[n_{FA}]}{V} = \frac{Np}{V} \quad (2.5.1-2)$$

If

$$p \ll 1 \quad (2.5.1-3)$$

then, for N large enough, such that Np is of the order of 1 (or larger), the Poisson distribution is a good approximation of (2.5.1-1) [Pap84].

The Poisson approximation for (2.5.1-1) is

$$\mu_{FA}(m) = e^{-Np} \frac{(Np)^m}{m!} \quad (2.5.1-4)$$

Since the Poisson approximation is quite accurate for $p \leq 0.1$, Eq. (2.5.1-4) can be considered as an equality for practical values of p , which are substantially smaller.

2.5.2 The Number and Location of False Alarms

2.5.2 The Number and Location of False Alarms

Using (2.5.1-2) in (2.5.1-4), one has the pmf of the **number of false alarms** or **clutter points** in the volume V , in terms of their spatial density λ , as

$$\mu_{FA}(m) = e^{-\lambda V} \frac{(\lambda V)^m}{m!} \quad (2.5.2-1)$$

This serves as the justification for the common use of the Poisson distribution for the number of false measurements in a certain volume under the two assumptions given above.

The spatial distribution of the false alarms is, based on the same two assumptions, *uniform*. Thus, neglecting the granularity due to the resolution cells, the pdf of the location of a false detection, i.e., the **pdf of a false measurement** is

$$p(z|z \text{ is a false measurement}) = \frac{1}{V} \quad (2.5.2-2)$$

where V denotes now the volume of the subspace in which the measurement is known to lie. This can be, depending on the situation considered, the sensor's surveillance region or a target's validation region.

Remarks

In practice the clutter is not always homogeneous, i.e., P_{FA} and, consequently, λ can vary within a sensor's surveillance region. Some of the algorithms to be discussed later require knowledge of the parameter λ , while others, called "nonparametric", do not require the clutter density parameter.

Another practical consideration is the existence of **persistent clutter** in the environment due to spurious energy reflectors or emitters. It should be kept in mind that the above model covers only **random clutter**. Ideally, the persistent clutter can be identified as such and then ignored. However, the real world also exhibits clutter that is somewhere between random and persistent.

2.6 TRACK FORMATION IN CLUTTER

2.6.1 A Logic-Based Track Formation Procedure

Track formation in the presence of measurement uncertainty requires **measurement-to-measurement association**.

One commonly used approach is a logic-based one that uses gates and requires a certain sequence of detections in these gates. If the requirement is satisfied, then the measurement sequence is accepted as a valid track.

The following two-stage **cascaded logic** that assumes target position measurements is considered:

1. Every unassociated detection (measurement) is an “initiator” — it yields a **tentative track**.
2. At the sampling time (**scan** or **frame**) following the detection of an initiator, a gate is set up based on the
 - assumed maximum and minimum target motion parameters
 - the measurement noise intensities

such that, if there is a target that gave rise to the initiator, the measurement from it in this second scan (if detected) will fall in the gate with nearly unity probability. Following a detection, this track becomes a **preliminary track**. If there is no detection, this track is dropped.

3. Since a preliminary track has two measurements, a Kalman filter can be initialized and used to set up a gate for the next (third) sampling time.
4. Starting from the third scan a logic of m detections out of n scans (*frames*) is used for the subsequent gates.
5. If at the end (scan $n+2$ at the latest) the logic requirement is satisfied, the track becomes a **confirmed track** or an **accepted track**. Otherwise it is dropped.

This logic can be called, in view of the above, $2/2\&m/n$.

The reason for requiring the first two detections is the following: the association gate for the second detection (following an “initiator”) uses only prior velocity information and is very large. If there is no second detection, the gate increases substantially for the next scan. Thus, in the absence of a second detection, this logic waits for a new initiator and this avoids the use of an even larger gate that opens the system to false tracks.

Performance Prediction

In this section a *non-simulation performance prediction technique* is presented for this type of cascaded logic for track formation.

This evaluates the probabilities of the following events:

- (a) acceptance of a target-originated sequence of detections — **true track detection probability** — P_{DT}
- (b) acceptance of a (purely) false-alarm-originated sequence of detections — **false track probability** — P_{FT} .

The evaluation of P_{DT} is based on the *single scan* or *single frame target detection probability* P_D , and in the absence of false alarms. The evaluation of P_{FT} is based on the false alarm probability P_{FA} , which is defined per resolution cell and the size of the “association gates” in units of resolution cells is needed.

In practice, the gate sizes are obtained from a least squares fit or Kalman filter based prediction and are, therefore, not only time-varying, but also depend on the *prior detection sequence*. The technique to be presented in the sequel accounts for the above two factors.

An approach that takes into account the effect of clutter on a logic-based formation of true tracks as well as varying gate sizes (based on maximum target maneuverability) is presented in Section 7.4.

2.6.2 The Target Model

2.6.2 The Target Model

The target is assumed to move in a 2-dimensional space and its motion is modeled as independent between coordinates. The motion in each coordinate has nearly constant velocity with piecewise constant white noise acceleration with variance $q = \sigma_v^2$ as in Subsection 6.3.2 of [BL93].

The measurements are position only, with additive white noise with variance r in each coordinate.

The state covariance for each coordinate evolves according to the usual Riccati equation (see Subsection 5.2.5 in [BL93]) modified with a “detection indicator” as follows:

$$P(k+1|k) = FP(k|k)F' + Q \quad (2.6.2-1)$$

$$P(k+1|k+1) = P(k+1|k) - \delta(k+1)P(k+1|k)H'S(k+1)^{-1}HP(k+1|k) \quad (2.6.2-2)$$

In the above F is the block-diagonal transition matrix consisting of two blocks, Q is the block-diagonal process noise covariance matrix (with the blocks given in Subsection 6.3.2 of [BL93]),

$$S(k+1) = HP(k+1|k)H' + R \quad (2.6.2-3)$$

is the innovation covariance matrix at time $k+1$, H is the measurement matrix (the measurement consists of each position coordinate), R is the measurement noise covariance matrix with diagonal elements r , and

$$\delta(k) = \begin{cases} 1 & \text{if there is a detection in the gate at } k \\ 0 & \text{otherwise} \end{cases} \quad (2.6.2-4)$$

is the **detection indicator** at time k .

2.6.3 Markov Chain Model of the Track Formation Process

2.6.3 Markov Chain Model of the Track Formation Process

The Markov chain that defines the cascaded logic 2/2&2/3 is given in Table 2.6.3-1. State i of this Markov chain is defined by the **detection sequence indicator vector** — the sequence of detections that led to state i , denoted as δ_i . The transition destinations following a detection (D) or no detection (\bar{D}) are also indicated.

State	Detection sequence indicator vector	Transitions
1	Initial (zero) state	$D \rightarrow 2; \bar{D} \rightarrow 1$
2	$\delta_2 = [1]$	$D \rightarrow 3; \bar{D} \rightarrow 1$
3	$\delta_3 = [1 1]$	$D \rightarrow 4; \bar{D} \rightarrow 6$
4	$\delta_4 = [1 1 1]$	$D \rightarrow 8a; \bar{D} \rightarrow 5$
5	$\delta_5 = [1 1 1 0]$	$D \rightarrow 8b; \bar{D} \rightarrow 1$
6	$\delta_6 = [1 1 0]$	$D \rightarrow 7; \bar{D} \rightarrow 1$
7	$\delta_7 = [1 1 0 1]$	$D \rightarrow 8c; \bar{D} \rightarrow 1$
8a	$\delta_{8a} = [1 1 1 1]$	Confirmed state
8b	$\delta_{8b} = [1 1 1 0 1]$	Confirmed state
8c	$\delta_{8c} = [1 1 0 1 1]$	Confirmed state

Table 2.6.3-1: The Markov chain for the 2/2&2/3 logic.

The three states 8a–8c are all **confirmed status** or **accepted track** states, i.e., the same.

The chain state probability vector, denoted by $\mu(k)$, has components

$$\mu_i(k) = P\{\text{the chain is in state } i \text{ at time } k\} \quad (2.6.3-1)$$

2.6.3 Markov Chain Model of the Track Formation Process

True Track Detection Probability

ASSUMPTION: The target detection event is independent across scans and has a fixed probability P_D .

The transition probabilities for the 8-state Markov chain for target-originated measurements are:

$$\begin{aligned}\pi_{11} &= \pi_{21} = \pi_{36} = \pi_{45} = \pi_{51} = \pi_{61} = \pi_{71} = 1 - P_D P_G \\ \pi_{12} &= \pi_{23} = \pi_{34} = \pi_{48} = \pi_{58} = \pi_{67} = \pi_{78} = P_D P_G \\ \pi_{88} &= 1\end{aligned}\quad (2.6.3-2)$$

where P_G is the gate probability (2.3.2-3).

To evaluate the probability of confirmation of a target-originated sequence, one starts at scan $k = 0$ with the initial chain state probability vector (of dimension 8) with

$$\mu_1(0) = 1 \quad (2.6.3-3)$$

The Markov chain probability vector is propagated according to

$$\boxed{\mu(k+1) = \Pi' \mu(k)} \quad k = 0, 1, \dots \quad (2.6.3-4)$$

where the elements of Π are given in (2.6.3-2).

The resulting $\mu_8(k)$ is the **cumulative probability mass function (cpmf)** of the **true track detection (acceptance or confirmation)** at time k , i.e., accepted at k or before

$$\boxed{P_{DT}(k) = \mu_8(k)} \quad (2.6.3-5)$$

and the corresponding pmf (probability of being accepted at k) is

$$m_8(k) = \mu_8(k) - \mu_8(k-1) \quad (2.6.3-6)$$

The **average confirmation time** of a target-originated sequence is

$$\boxed{\bar{t}_C = \sum_{k=1}^{\infty} km_8(k)} \quad (2.6.3-7)$$

2.6.3 Markov Chain Model of the Track Formation Process

False Track Acceptance Probability

ASSUMPTIONS:

- The false alarms occur with a fixed probability P_{FA} per resolution cell per scan, independently across cells and scans,
- The probability of having more than one false alarm per gate is negligible.

The Markov chain will model the sequence of events starting with *one false alarm in a resolution cell in a scan*.

The chain is the same as in Table 2.6.3-1, except that it starts at $k = 1$ with the initial state probability vector with components

$$\mu_1(1) = 1 - P_{FA} \quad \mu_2(1) = P_{FA} \quad (2.6.3-8)$$

KEY OBSERVATION: *Each resolution cell can start in each scan a sequence of detections, which can end up in a **track acceptance** (state 8) or back to the “zero state” (state 1).*

State 1 is an “absorbing state” — the potential false track has been rejected. Thus

$$\pi_{11} = 1 \quad (2.6.3-9)$$

The transition probabilities from the other states now depend on the number of cells each gate has. With the volume of the gate at scan 2 being V_2 (in resolution cell units), one has

$$\pi_{23} = V_2 P_{FA} \quad \pi_{21} = 1 - \pi_{23} \quad (2.6.3-10)$$

The volumes of the association gate corresponding to state j is given, according to (2.3.2-4), for a two-dimensional measurement, by

$$V_j = \gamma \pi |S(\delta_j)|^{1/2} \quad (2.6.3-11)$$

where $S(\delta_j)$ is the innovation covariance corresponding to the detection sequence δ_j and γ is the gate threshold.

2.6.3 Markov Chain Model of the Track Formation Process

The remaining Markov chain transition probabilities are

$$\begin{aligned}
 \pi_{34} &= V_3 P_{FA} & \pi_{36} &= 1 - \pi_{34} \\
 \pi_{48} &= V_4 P_{FA} & \pi_{45} &= 1 - \pi_{48} \\
 \pi_{58} &= V_5 P_{FA} & \pi_{51} &= 1 - \pi_{58} \\
 \pi_{67} &= V_6 P_{FA} & \pi_{61} &= 1 - \pi_{67} \\
 \pi_{78} &= V_7 P_{FA} & \pi_{71} &= 1 - \pi_{78} \\
 \pi_{88} &= 1 & &
 \end{aligned} \tag{2.6.3-12}$$

With the elements of Π given above, this chain evolves according to

$$\mu(k+1) = \Pi' \mu(k) \quad k = 0, 1, \dots \tag{2.6.3-13}$$

and reaches its final state in 4 steps.

The **false track probability**, i.e., the probability that a resolution cell yields an **accepted false track per scan** is

$$P_{FT} = \mu_8(5) \tag{2.6.3-14}$$

The **average number of accepted false tracks** in the system *per scan* follows by multiplying the above with the number of resolution cells N_c

$$E[N_{AFT}] = N_c P_{FT} \tag{2.6.3-15}$$

The **average false track length** (in sampling periods) is

$$\bar{\tau}_F = \frac{\sum_{i=2}^8 \sum_{k=1}^5 k \mu_i(k)}{\sum_{i=2}^8 \sum_{k=1}^5 \mu_i(k)} \tag{2.6.3-16}$$

The **average number of existing false tracks** in the system at a given time *prior to acceptance or elimination*, needed for **processor sizing**, is

$$E[N_{EFT}] = N_c P_{FA} \bar{\tau}_F \tag{2.6.3-17}$$

Note that this Markov chain was obtained “manually” and, for different values of the parameters m and n a different chain is needed. An automatic **Markov chain synthesis** scheme, implemented in the interactive software MULTIDATTM, is described later.

2.6.4 Numerical Examples

2.6.4 Numerical Examples

This algorithm has been implemented in the interactive software MULTIDATTM for the model described in Section 2.2 for logic parameters $1 \leq m \leq n \leq 6$.

Figures 2.6.4-1 through 2.6.4-4 show the cpmf (cumulative probability mass function) of the confirmation of a target track vs. time for a range of P_D and for several logic parameters. Table 2.6.4-1 shows the average confirmation time (in scans or sampling periods).

P_D	0.95	0.9	0.85	0.8	0.75	0.7
<i>m/n</i>						
2/3	4.29	4.65	5.13	5.73	6.5	7.51
1/2	3.22	3.48	3.8	4.18	4.65	5.24
2/4	4.27	4.58	4.95	5.42	5.99	6.73
3/4	5.37	5.89	6.58	7.52	8.76	10.47

Table 2.6.4-1: Average true track confirmation time.

For evaluation of the false-alarm-originated track probabilities, the following parameters were used:

$V_2 = 400$ (resolution cells — this follows, with motion in two dimensions, from an assumption of maximum velocity of 10 cells/scan in each direction)

$r = 1$ (in resolution cell units, in each of the two coordinates)

$\gamma = 9$ (about 99% gate probability)

The volume of the association gate at scan 3 for these parameters, in the absence of process noise, is, using Table 3.5.3-1 of [BL93],

$$V_3 = \pi \gamma^5 \sqrt{r_{xx} r_{yy}} \approx 150 \quad (2.6.4-1)$$

2.6.4 Numerical Examples

Table 2.6.4-2 presents the resulting false track probability and the average length (in scans) of a false track for various false alarm densities and logic parameters. The highest false alarm rate density ($P_{FA} = 10^{-3}$) yields an expected number of false detections of 0.4 in V_2 and 0.15 in V_3 .

The first logic is evaluated both for a model without process noise, as well as one with process noise that corresponds to a **target maneuvering index**³ of 0.3. The gate size increase due to this level of process noise does not have a significant impact on the performance.

		P_{FA}	$2 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	10^{-4}	$2 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	10^{-3}
m/n	q							
2/3	0	P_{FT}	$3.06 \cdot 10^{-12}$	$1.19 \cdot 10^{-10}$	$1.9 \cdot 10^{-9}$	$3.02 \cdot 10^{-8}$	$1.15 \cdot 10^{-6}$	$1.76 \cdot 10^{-5}$
2/3	0.1	P_{FT}	$3.34 \cdot 10^{-12}$	$1.3 \cdot 10^{-10}$	$2.1 \cdot 10^{-9}$	$3.29 \cdot 10^{-8}$	$1.25 \cdot 10^{-6}$	$1.91 \cdot 10^{-5}$
1/2	0.1	P_{FT}	$1.65 \cdot 10^{-9}$	$2.57 \cdot 10^{-8}$	$2.05 \cdot 10^{-7}$	$1.62 \cdot 10^{-6}$	$2.46 \cdot 10^{-5}$	$1.85 \cdot 10^{-4}$
2/4	0.1	P_{FT}	$1.08 \cdot 10^{-11}$	$4.17 \cdot 10^{-10}$	$6.6 \cdot 10^{-9}$	$1.03 \cdot 10^{-7}$	$3.74 \cdot 10^{-6}$	$5.26 \cdot 10^{-5}$
3/4	0.1	P_{FT}	$4.17 \cdot 10^{-15}$	$4.06 \cdot 10^{-13}$	$1.29 \cdot 10^{-11}$	$4.11 \cdot 10^{-10}$	$3.93 \cdot 10^{-8}$	$1.21 \cdot 10^{-6}$
2/3	0	$\bar{\tau}_F$	1.024	1.06	1.116	1.23	1.52	1.91
2/3	0.1	$\bar{\tau}_F$	1.024	1.06	1.116	1.23	1.52	1.91
1/2	0.1	$\bar{\tau}_F$	1.02	1.06	1.116	1.23	1.52	1.89
2/4	0.1	$\bar{\tau}_F$	1.05	1.12	1.23	1.44	1.97	2.55
3/4	0.1	$\bar{\tau}_F$	1.02	1.06	1.1176	1.23	1.52	1.92

Table 2.6.4-2: False track probability and average false track length.

³The ratio of the RMS motion uncertainty in position over one sampling period to the RMS position measurement uncertainty (1.7.3-1).

2.6.4 Numerical Examples

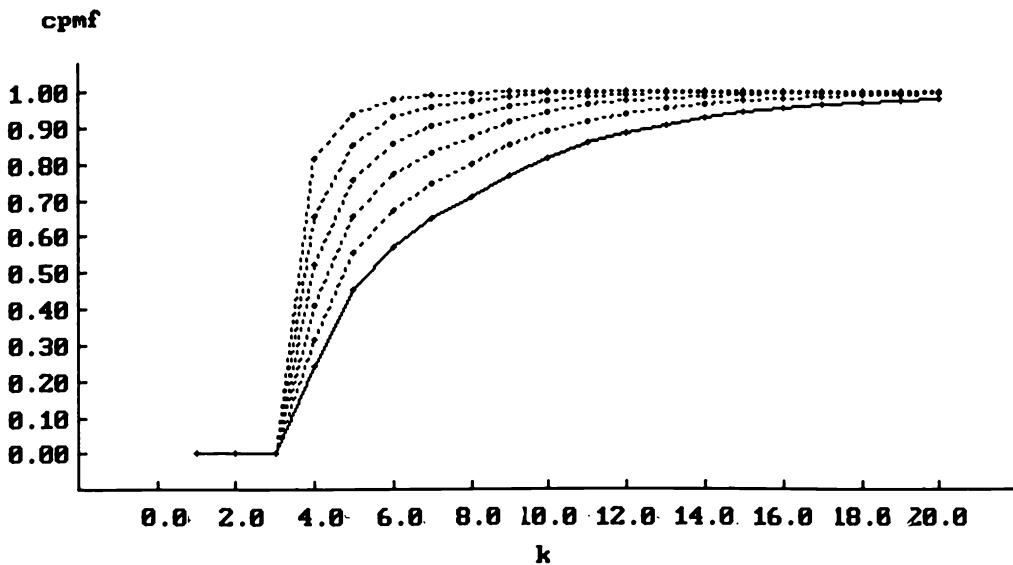


Figure 2.6.4-1: Target confirmation cpmf vs. time for $m = 2$, $n = 3$ (from top to bottom: $P_D = 0.95, 0.9, 0.85, 0.8, 0.75, 0.7$).

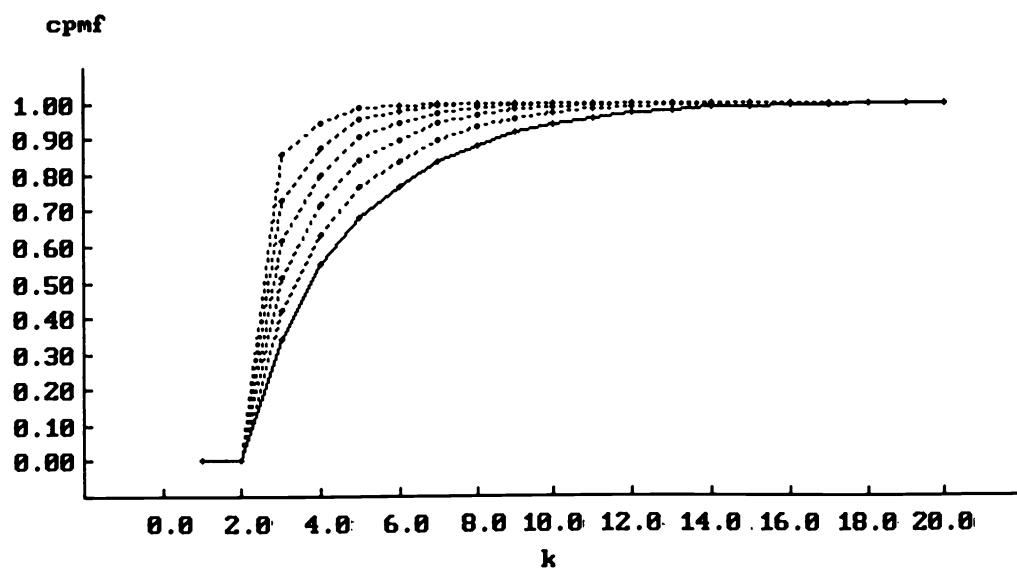


Figure 2.6.4-2: Target confirmation cpmf vs. time for $m = 1$, $n = 2$ (from top to bottom: $P_D = 0.95, 0.9, 0.85, 0.8, 0.75, 0.7$).

2.6.4 Numerical Examples

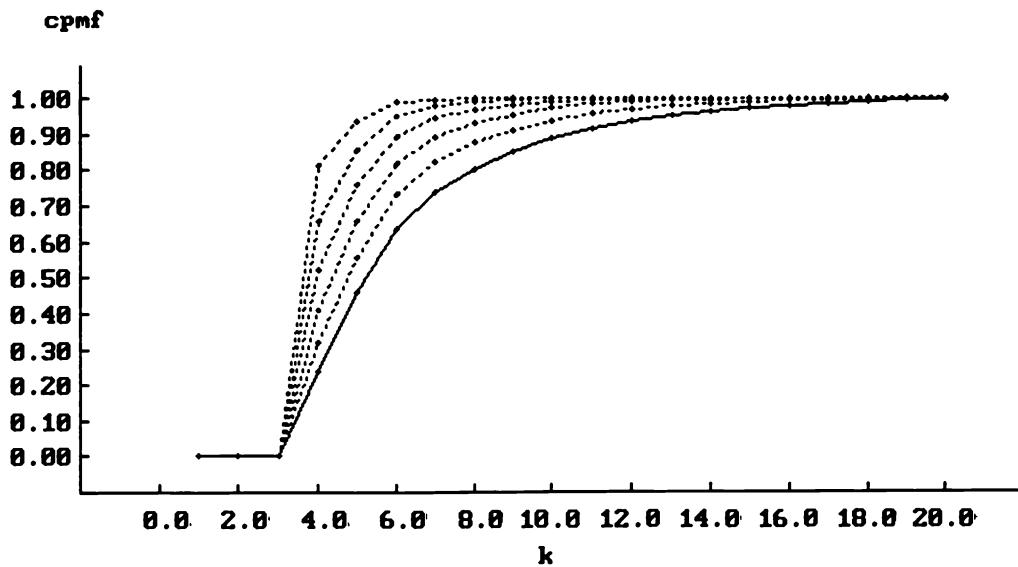


Figure 2.6.4-3: Target confirmation cpmf vs. time for $m = 2$, $n = 4$ (from top to bottom: $P_D = 0.95, 0.9, 0.85, 0.8, 0.75, 0.7$).

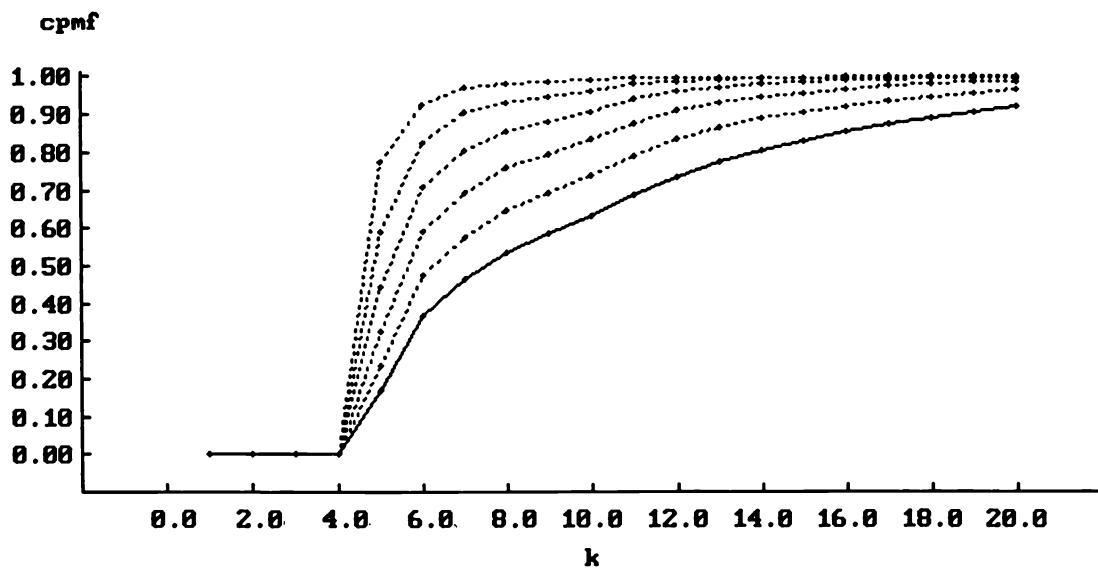


Figure 2.6.4-4: Target confirmation cpmf vs. time for $m = 3$, $n = 4$ (from top to bottom: $P_D = 0.95, 0.9, 0.85, 0.8, 0.75, 0.7$).

2.6.5 A Design Example

2.6.5 A Design Example

Assume that one has the following sensor parameters

- (i) target detection probability $P_D = 0.9$ and
- (ii) false alarm probability $P_{FA} = 2 \cdot 10^{-5}$.

The system requirements are:

- (a) $P_{DT} \geq 0.8$ by time $k = 6$, and
- (b) $P_{FT} \leq 1 \cdot 10^{-14}$.

In this case, the values $m = 2$, $n = 3$ yield $P_{DT} = 0.95$ (from Figure 2.6.4-1) and $P_{FT} = 3.3 \cdot 10^{-12}$ (from Table 2.6.4-2), i.e., requirement (b) is not met.

Considering now another logic, with $m = 3$, $n = 4$, we obtain $P_{DT} = 0.82$ (from Figure 2.6.4-4) and $P_{FT} = 4.2 \cdot 10^{-15}$ (from Table 2.6.4-2).

Thus we have to use the second set of logic parameters to satisfy both requirements. This is at the price of the average true track confirmation time being now somewhat higher (5.89 vs. 4.65, from Table 2.6.4-1).

Another way of using the tools developed in this section is

- to fix the logic parameters and
- then vary P_D and P_{FA} according to a Receiver Operating Characteristic curve (i.e., varying the detection threshold) for the assumed SNR (signal to noise ratio)

to evaluate the feasible pairs P_{DT} and P_{FT} and see if the system requirements can be met. Such an approach is used in Section 7.4.

2.6.6 Cascaded Logic Track Formation Evaluation — Summary

2.6.6 Cascaded Logic Track Formation Evaluation — Summary

A track formation procedure using a two-stage cascaded logic $2/2\&m/n$ has been described and its performance evaluated.

The performance of this track formation algorithm was given in terms of

- the probability that a target-originated sequence of detections is accepted — *true track detection probability*
- the probability that a false-alarm-originated sequence of detections is accepted — *false track probability* (per resolution cell per scan).

The evaluation of these performance measures was carried out via a Markov chain analysis, using as input parameters

- the single scan target detection probability
- the false alarm probability (per resolution cell per scan)

and

- accounting for the variation of the association gates during the track formation process.

The gate sizes were assumed to be determined from a Kalman filter based prediction, i.e., they were of minimum size.

Expressions were given for the averages (expected values) of the following:

- target track confirmation time
- number of confirmed false tracks
- false track length
- number of existing false tracks (prior to confirmation or elimination).

2.6.7 Automatic Synthesis of the Markov Chain for Track Formation Evaluation

The procedure for **automatic generation of the Markov chain** for the $2/2\&m/n$ track formation logic is described below.

Denote the number of detections and the number of misses as

$$\lambda_D(\delta) = \sum_{j=3}^k \delta_j; \quad \lambda_D = m \implies \text{"A"} \text{(acceptance)} \quad (2.6.7-1)$$

$$\lambda_M(\delta) = \sum_{j=3}^k (1 - \delta_j); \quad \lambda_M = n - m + 1 \implies \text{"R"} \text{(rejection)} \quad (2.6.7-2)$$

Table 2.6.7-1 below describes the chain for $m/n = 2/3$ (the state numbers differ somewhat from Table 2.6.3-1).

k	δ	λ_D	λ_M	Chain state index	Transition (origin, event, destination)
0	—	—	—	1	
1	0	—	—	"R"=1	1, \bar{D} , 1
	1	—	—	2	1, D , 2
2	10	—	—	"R"=1	2, \bar{D} , 1
	11	—	—	3	2, D , 3
3	110	0	1	4	3, \bar{D} , 4
	111	1	0	5	3, D , 5
4	1100	0	2	"R"=1	4, \bar{D} , 1
	1101	1	1	6	4, D , 6
	1110	1	1	7	5, \bar{D} , 7
	1111	2	0	"A"=8	5, D , 8; 8, D/\bar{D} , 8
5	11010	1	2	"R"=1	6, \bar{D} , 1
	11011	2	1	"A"=8	6, D , 8
	11100	1	2	"R"=1	7, \bar{D} , 1
	11101	2	1	"A"=8	7, D , 8

Table 2.6.7-1: Automatically generated Markov chain example.

2.7 NOTES AND PROBLEMS

2.7.1 Bibliographical Notes

The sequential updating with multisensor measurements presented in Section 2.2 can be found in, e.g., [WCD76] or [AM79]. The former also discusses the parallel vs. sequential state update procedures. A discussion of the use of multiple sensors in the presence of jamming can be found in [TL87].

The *validation* or *gating* technique of Section 2.3 can be traced back to [Sit64] with the multi-dimensional version presented in [SK71].

The issue of a single target in clutter discussed in Section 2.4 was first treated in the context of modern estimation theory in [SS71] and [Sea71]. The discussion of the multiple targets in clutter using validation regions in Subsection 2.4.2 is based on [Bar74].

A discussion of clutter in radar problems and the generation of “clutter maps” to reduce the effect of fixed clutter on tracking can be found in [FS85]. The Poisson approximation presented in Section 2.5 for the binomial distribution of the number of false measurements can be found in [Pap84] and its proof in [Par60]. More complex clutter models can be found in, e.g., [PB86].

Track initiation procedures with m/n logic were discussed in [Cas76, MO78, FS85] and curves of the probability of initiation of a track for a target (with $P_D < 1$) in the absence of clutter were presented, as well as due to clutter only (assuming a fixed probability of clutter in the gate). The evaluations of [Cas76] (also in [FS85]) and [MO78] assumed, however, *fixed gates*. Section 2.6 is based on [BSP90] and this algorithm has been implemented in the interactive software MULTIDATTM. The extension of this technique to incorporate the goodness of fit of a sequence of measurements (discussed later in Section 3.3) appears in [BL91].

Approaches that “assign” measurements to tracks based on score functions (e.g., sum of distances squared) have been in practical use [Bla86]. The general assignment algorithm of Munkres [BL71b, BL71a], can minimize such a score function. More recent work along these lines is [PDBW92].

2.7.2 Problems

- 2-1 **Gating based on maximum velocity and measurement noise.** Consider a sensor that makes position measurements in two Cartesian coordinates with zero-mean white and mutually independent additive Gaussian noises with variance σ^2 . A target can have a velocity of magnitude $|v| \leq v_{\max}$ with the heading uniformly distributed in all directions. Given a first measurement, at $t = 0$, at location (z_1, z_2) , find the region in the measurement space where the second measurement, at $t = T$, is expected to be found with a high probability.

Chapter 3

ALGORITHMS FOR TRACKING A SINGLE TARGET IN CLUTTER

3.1 INTRODUCTION

3.1.1 Outline

This chapter deals with the state estimation of a single target using measurements of uncertain origin. The target state is observed with detection probability possibly less than unity and in the presence of clutter, which gives rise to false measurements.

Tracks are assumed initialized, i.e., the problem is **track maintenance**.

At every time a validation or association region is set up and one or more measurements can be found in it.

The simplest possible approach is to use the measurement nearest to the predicted measurement as if it were the correct one. This approach, called **Nearest Neighbor Standard Filter (NNSF)**, is described in Section 3.2. Assuming that signal intensity information is available, an alternative is to select the *strongest measurement* among the validated ones. This yields the **Strongest Neighbor Standard Filter (SNSF)**, also discussed in Section 3.2.

Since any of the validated measurements could have originated from the target, this suggests that *all the measurements from the validation region* should be used in some fashion.

One approach is to split the track every time more than one measurement is found in the validation region. This **track split** approach, discussed in Section 3.3 uses a likelihood function based “pruning” technique to reduce the number of false tracks.

3.1.1 Outline

A Bayesian approach, called **Probabilistic Data Association (PDA)**, discussed in Section 3.4, associates probabilistically all the “neighbors” — each validated measurement at *the current time* — to the target of interest.¹ This probabilistic information is used in a tracking filter, called **PDA filter (PDAF)**, that accounts for the measurement origin uncertainty.

The incorporation of **feature measurements** in addition to the usual kinematic measurements into the PDA technique is also discussed.

The optimal Bayesian filter, presented in Section 3.5, computes association probabilities for each **sequence of measurements** or **history**. This optimal approach amounts to track splitting and a probability calculation for each split track.

Section 3.6 deals with the specialized problem of target tracking in the presence of a **wake**.

The use of the PDA in conjunction with an ML approach for **track formation** based on a batch of bearing and frequency measurements (i.e., no direct position observations), typical of a narrow band passive sonar operating in a low SNR environment (in the presence of false alarms), is presented in Section 3.7.

¹At MIT, a number of years ago, PDA was outlawed: there was a prominent sign in the dorms “NO PDA ALLOWED” (Public Display of Affection).

3.1.2 Summary of Objectives

3.1.2 Summary of Objectives

Present algorithms for tracking a target in the presence of false measurements

- Non-Bayesian association techniques
 - Nearest Neighbor Standard Filter
 - Strongest Neighbor Standard Filter
 - Track-Split approach
- Bayesian association techniques
 - Probabilistic Data Association Filter
 - Optimal Bayesian approach (multiple hypotheses).

Examine performance limits of the PDAF.

Incorporate feature information into the PDAF.

Develop an ML/PDA track formation technique for low SNR passive sonar measurements.

3.1.3 The Model for Tracking a Single Target in Clutter

3.1.3 The Model for Tracking a Single Target in Clutter

The state and measurement equations are assumed linear with additive zero-mean white noises with known covariances.

The state of the target of interest, of dimension n_x , is assumed to evolve in time according to the equation

$$x(k+1) = F(k)x(k) + v(k) \quad (3.1.3-1)$$

with the **true measurement**, of dimension n_z , given by

$$z(k) = H(k)x(k) + w(k) \quad (3.1.3-2)$$

where $v(k)$ and $w(k)$ are zero-mean mutually independent white Gaussian noise sequences with known covariance matrices $Q(k)$ and $R(k)$, respectively.

The model for the **false measurements** will be as in Section 2.5, i.e., uniformly distributed in the measurement space.

Tracks are assumed initialized — an initial state estimate and associated covariance are available.

A validation region is set up at every sampling time around the predicted measurement and possibly several measurements fall in it. The various approaches differ in how these measurements are used (or not) in the estimation of the state of the target.

All the variables that carry information useful to differentiate the correct measurement from the incorrect ones are assumed to be included in the measurement vector.

Remark

Even though a linear model is considered, the techniques to be discussed in the sequel can be also used on nonlinear models by carrying out linearization.

3.2 USE OF THE STANDARD FILTER

There are two variants of this approach, which consists of the following steps:

1. Validation of the measurements using (2.3.2-2)
2. Selection of *one of the validated measurements*
3. Update of the target state with this measurement *as if it were the correct one*, i.e., in the *standard* Kalman filter manner.

Step 2 has the following options:

- 2A. Select the *nearest measurement* to the predicted measurement according to the distance measure (norm of the innovation squared).

$$D(z) = [z - \hat{z}(k+1|k)]' S(k+1)^{-1} [z - \hat{z}(k+1|k)] = \nu' S(k+1)^{-1} \nu \quad (3.2.0-3)$$

where S is the covariance matrix of the innovation (of the true measurement). This is the **Nearest Neighbor Standard Filter (NNSF)**.

- 2B. Select the *strongest measurement* (in terms of signal intensity) among the validated ones — this assumes that signal intensity information is available. This is the **Strongest Neighbor Standard Filter (SNSF)**.

The problem with choosing the nearest or strongest neighbor is that, with some probability, it is not the correct measurement. Therefore, the filter will use (sometimes) incorrect measurements while “believing” that they are correct. This amounts to “overconfidence” and, even with moderate clutter density, it can lead to loss of the target as it will be shown later.

The NNSF is common for radar trackers, while the SNSF is used in sonar trackers.

3.3 THE TRACK SPLIT FILTER — A MARGINAL LIKELIHOOD FUNCTION APPROACH

3.3.1 The Track Splitting Approach

The *track split* procedure consists of the following steps:

1. After the initialization (at $k = 0$), for every measurement at $k = 1$ that falls in the validation region around the location $\hat{z}(1|0)$ where the measurement is expected, the track is split.
2. For each measurement an updated state is computed via the (standard) Kalman Filter equations [BL93], using the state model (3.1.3-1), (3.1.3-2), and propagated forward to yield another validation region at $k = 2$.
3. For each new validation region at $k = 2$ the procedure is repeated.
4. Since the number of branches into which the track is split can grow exponentially, the *likelihood function* of each split track is computed and the unlikely ones are discarded.

Remark

The likelihood functions are evaluated for each track *separately*, and, thus, this method utilizes a *marginal likelihood function approach*.

The better (and more costly) approach would be to evaluate *jointly* the likelihood of all the possible tracks — this will be discussed later.

3.3.2 The Likelihood Function of a Track

3.3.2 The Likelihood Function of a Track

Denote the l -th sequence of measurements up to time k as

$$Z^{k,l} = \{z_{i,j,l}(j)\}_{j=1}^k \quad (3.3.2-1)$$

where $z_{i,j,l}(j)$ is the measurement at time j that belongs to sequence l .

The **track likelihood function** defined as the likelihood function of this sequence of measurements being a true (correct) track, i.e., conditioned upon their having *originated from the same target*

$$\theta^{k,l} = \{Z^{k,l} \text{ is a true track}\} \quad (3.3.2-2)$$

is the probability density function

$$\boxed{\Lambda(\theta^{k,l}) = p[z(1), \dots, z(k) | \theta^{k,l}]} \quad (3.3.2-3)$$

where the subscripts of the measurements have been dropped for simplicity. The track initialization information (initial estimate) is subsumed in (3.3.2-3).

Using the notation

$$Z^j = \{z(i)\}_{i=1}^j \quad (3.3.2-4)$$

the joint pdf of the measurements can be rewritten as the following product (see [BL93]):

$$\Lambda(\theta^{k,l}) = \prod_{j=1}^k p[z(j) | Z^{j-1}, \theta^{k,l}] \quad (3.3.2-5)$$

Assumptions

- The system is linear-Gaussian (or it can be so approximated)
- The target probability of detection is unity.

Under the linear-Gaussian assumptions, from [BL93] one has

$$p[z(j) | Z^{j-1}, \theta^{k,l}] = \mathcal{N}[z(j); \hat{z}(j|j-1), S(j)] = \mathcal{N}[\nu(j); 0, S(j)] \quad (3.3.2-6)$$

From (3.3.2-5) and (3.3.2-6) the track likelihood function is

$$\Lambda(\theta^{k,l}) = c_k e^{-\frac{1}{2} \sum_{j=1}^k \nu(j)' S(j)^{-1} \nu(j)} \quad (3.3.2-7)$$

where c_k is the normalization constant. The above assumes implicitly that the target probability of detection is unity.

3.3.2 The Likelihood Function of a Track

The negative **log-likelihood function** (multiplied by 2 and without the additive constant) is, with the sequence index omitted for simplicity,

$$\lambda^k \triangleq -2 \ln \frac{\Lambda(\theta^{k,l})}{c_k} = \sum_{j=1}^k \nu(j)' S(j)^{-1} \nu(j) \quad (3.3.2-8)$$

and can be also computed recursively as follows

$$\boxed{\lambda^k = \lambda^{k-1} + \nu(k)' S(k)^{-1} \nu(k)} \quad (3.3.2-9)$$

The last term above has a chi-square density with n_z (dimension of the measurement vector) degrees of freedom and, since the innovations are independent, the log-likelihood function at time k is chi-squared distributed with kn_z degrees of freedom.

Note that (3.3.2-9) is a measure of the *goodness of fit* of the measurements to the assumed target model.

Statistical Test for Keeping a Track

A track is not discarded if its log-likelihood function is below a threshold

$$\lambda^k \leq \lambda_{max}^k \quad (3.3.2-10)$$

This threshold follows from the chi-square tables for kn_z degrees of freedom as

$$\lambda_{max}^k = \chi_{kn_z}^2(1 - \alpha) \quad (3.3.2-11)$$

where the **tail probability** α is the probability that a true track will be rejected and it is taken, typically, as 0.01.

3.3.2 The Likelihood Function of a Track

Extension to Detection Probability Less Than Unity

The assumption of unity probability of detection ($P_D = 1$) implies that only “full” sequences of measurements are to be considered. If $P_D < 1$ then *incomplete sequences*, i.e., with missing measurements, should also be considered.

In this case a track is kept if it satisfies the requirements

- (a) it has, say, m detections out of the last n sampling times, and
- (b) its goodness of fit is acceptable in the sense of (3.3.2-10)

i.e., a combination of the technique of Section 2.6 and the present one.

Remarks

1. This algorithm is of the sequential type and can be used in an environment where the number of targets is unknown (but it is not very powerful).
2. This (likelihood function) approach does not yield a probability that a sequence is correct — a general characteristic of the non-Bayesian approaches.
3. This approach considers potential tracks on an individual basis since it evaluates only the (marginal) likelihood function of a track. Thus it ignores the “competition” for the same measurements between several tracks. This problem is taken care of in the “Joint Maximum Likelihood” approach to be described in Section 7.2.
4. The main problem with this approach is that its computational and memory requirements can grow with time and saturate even large computing systems.

3.3.3 The Track Splitting Approach — Summary

3.3.3 The Track Splitting Approach — Summary

At every sampling time when there is more than one measurement in the validation region the track is split.

A standard filter is used to update the state of the target.

The likelihood function of each split track is evaluated in order to eliminate unlikely tracks.

The likelihood function measures the *goodness of fit* of the measurements to the target model.

The *negative log-likelihood function* can be calculated recursively as the sum of the normalized square innovations and has to be below a threshold obtained from the chi-square distribution.

The algorithm can be used for the case of unity probability of detection ($P_D = 1$) as well as for the (realistic) case when $P_D < 1$. In the latter case it is to be combined with a detection logic technique as the one of Section 2.6.

This approach does not yield a probability that a sequence is correct — it makes only keep/discard decisions.

The limited power of discrimination of this algorithm against false tracks, combined with its computational and memory requirements that can grow with time and saturate even large computing systems, makes it of limited interest.

3.4 A SUBOPTIMAL BAYESIAN ALGORITHM: THE PROBABILISTIC DATA ASSOCIATION FILTER

3.4.1 Introduction to Probabilistic Data Association

The **Probabilistic Data Association (PDA)** algorithm calculates the association probabilities for each validated measurement at the *current time* to the target of interest. This probabilistic (Bayesian) information is used in a tracking filter, called **PDA filter (PDAF)**, that accounts for the measurement origin uncertainty.

Assumptions

- There is only one target of interest, modeled by (3.1.3-1), (3.1.3-2)
- The track has been initialized
- The past information about the target is summarized approximately by

$$p[x(k)|Z^{k-1}] = \mathcal{N}[x(k); \hat{x}(k|k-1), P(k|k-1)] \quad (3.4.1-1)$$

The above *basic assumption of the PDAF* is similar to the *GPB1 approach* where a single “lumped” state estimate is a *quasi-sufficient statistic*

- At each time a validation region as in (2.3.2-2) is set up
- Among the possibly several validated measurements, at most one of them can be target-originated according to (3.1.3-2) — if the target was detected and the corresponding measurement fell into the validation region
- The remaining measurements are assumed due to false alarm or clutter and are modeled as *i.i.d. (independent identically distributed)* with uniform spatial distribution
- The target detections occur independently over time with known probability P_D .

These assumptions make it possible to obtain a state estimation scheme that is almost as simple as the KF, but much more effective in clutter.

The PDAF Approach

The PDAF uses a decomposition of the estimation w.r.t. the origin of each element of the *latest set of validated measurements*, denoted as

$$Z(k) = \{z_i(k)\}_{i=1}^{m(k)} \quad (3.4.1-2)$$

where $z_i(k)$ is the *i-th validated measurement* and $m(k)$ is the number of measurements in the validation region at time k .

The cumulative set (sequence) of measurements² is

$$Z^k = \{Z(j)\}_{j=1}^k \quad (3.4.1-3)$$

The optimal algorithm, to be presented in Section 3.5, carries out a decomposition with respect to each *sequence of measurements* from the set (3.4.1-3). In order to carry out a decomposition w.r.t. the latest measurements only, the assumption given in (3.4.1-1) was made.

Measurement Validation

Following (3.4.1-1), the validation region is the elliptical region

$$\boxed{\mathcal{V}(k, \gamma) = \{z : [z - \hat{z}(k|k-1)]' S(k)^{-1} [z - \hat{z}(k|k-1)] \leq \gamma\}} \quad (3.4.1-4)$$

where γ is the *gate threshold* and

$$S(k) = H(k)P(k|k-1)H(k)' + R(k) \quad (3.4.1-5)$$

is the covariance of the innovation corresponding to the true measurement. The **volume of the validation region** (3.4.1-4) is

$$V(k) = c_{n_z} |\gamma S(k)|^{1/2} = c_{n_z} \gamma^{\frac{n_z}{2}} |S(k)|^{1/2} \quad (3.4.1-6)$$

where the coefficient c_{n_z} depends on the dimension of the measurement (it is the volume of the n_z -dimensional unit hypersphere: $c_1 = 2$, $c_2 = \pi$, $c_3 = 4\pi/3$, etc.). Its general expression is given in (2.3.2-5).

²When the running index is a time argument, one has a sequence, otherwise it is a set where the order is not relevant. The context should make it clear which is the case.

3.4.2 The State Estimation

In view of the assumptions listed, the *association events*

$$\theta_i(k) = \begin{cases} \{z_i(k)\text{ is the target originated measurement}\} & i = 1, \dots, m(k) \\ \{\text{none of the measurements is target originated}\} & i = 0 \end{cases} \quad (3.4.2-1)$$

are *mutually exclusive and exhaustive* for $m(k) \geq 1$.

Using the total probability theorem [BL93] w.r.t. the above events, the conditional mean of the state at time k can be written as

$$\begin{aligned} \hat{x}(k|k) &= E[x(k)|Z^k] \\ &= \sum_{i=0}^{m(k)} E[x(k)|\theta_i(k), Z^k] P\{\theta_i(k)|Z^k\} \\ &= \sum_{i=0}^{m(k)} \hat{x}_i(k|k) \beta_i(k) \end{aligned} \quad (3.4.2-2)$$

where $\hat{x}_i(k|k)$ is the updated state *conditioned on the event that the i -th validated measurement is correct* and

$$\beta_i(k) \triangleq P\{\theta_i(k)|Z^k\} \quad (3.4.2-3)$$

is the conditional probability of this event — the **association probability**, obtained from the PDA procedure, presented in the next subsection.

The estimate conditioned on measurement i being correct is

$$\hat{x}_i(k|k) = \hat{x}(k|k-1) + W(k)\nu_i(k) \quad i = 1, \dots, m(k) \quad (3.4.2-4)$$

where the corresponding innovation is

$$\nu_i(k) = z_i(k) - \hat{z}(k|k-1) \quad (3.4.2-5)$$

The gain $W(k)$ is the same as in the standard filter

$$W(k) = P(k|k-1)H(k)'S(k)^{-1} \quad (3.4.2-6)$$

since, conditioned on $\theta_i(k)$, there is no measurement origin uncertainty.

For $i = 0$, i.e., if none of the measurements is correct, or, if there is no validated measurement (i.e., $m(k) = 0$), one has

$$\hat{x}_0(k|k) = \hat{x}(k|k-1) \quad (3.4.2-7)$$

3.4.2 The State Estimation

The State and Covariance Update

Combining (3.4.2-4) and (3.4.2-7) into (3.4.2-2) yields the state update equation of the PDAF

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)\nu(k) \quad (3.4.2-8)$$

where the **combined innovation** is

$$\nu(k) = \sum_{i=1}^{m(k)} \beta_i(k) \nu_i(k) \quad (3.4.2-9)$$

The covariance associated with the updated state is (proof in Subsection 3.4.9)

$$P(k|k) = \beta_0(k)P(k|k-1) + [1 - \beta_0(k)]P^c(k|k) + \tilde{P}(k) \quad (3.4.2-10)$$

where the covariance of the state updated with the *correct measurement* is

$$P^c(k|k) = P(k|k-1) - W(k)S(k)W(k)' \quad (3.4.2-11)$$

and the **spread of the innovations** term (similar to the spread of the means term in a mixture [BL93]) is

$$\tilde{P}(k) \triangleq W(k) \left[\sum_{i=1}^{m(k)} \beta_i(k) \nu_i(k) \nu_i(k)' - \nu(k) \nu(k)' \right] W(k)' \quad (3.4.2-12)$$

Remarks on the Covariance Update Equation

1. With probability $\beta_0(k)$ none of the measurements is correct: the prediction covariance $P(k|k-1)$ (“no update”) appears with this weighting.
2. With probability $1 - \beta_0(k)$ the correct measurement is available and the updated covariance $P^c(k|k)$ appears with this weighting.
3. Since it is not known which of the $m(k)$ validated measurements is correct, the term \tilde{P} , which is *positive semidefinite* increases the covariance of the updated state — this is the effect of the measurement origin uncertainty.
4. Note the *data dependence of the estimation accuracy*, typical of nonlinear estimators. The estimate update in Eq. (3.4.2-8) appears linear, but it is nonlinear due to the association probabilities $\beta_i(k)$ that depend on the innovations.

3.4.2 The State Estimation

The Prediction Equations

The prediction of the state and measurement to $k + 1$ is done as in the standard filter, i.e.,

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) \quad (3.4.2-13)$$

$$\hat{z}(k+1|k) = H(k+1)\hat{x}(k+1|k) \quad (3.4.2-14)$$

The covariance of the predicted state is, similarly,

$$P(k+1|k) = F(k)P(k|k)F(k)' + Q(k) \quad (3.4.2-15)$$

where $P(k|k)$ is given by (3.4.2-10).

The innovation covariance (for the correct measurement) is, again, as in the standard filter

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1) \quad (3.4.2-16)$$

Remark

Note that (3.4.2-16) is the *covariance of the innovation corresponding to the correct measurement*, even though it is not known which is the correct one. This is a direct consequence of (3.4.1-1), which “lumps” all the data into a single state estimate for the target.

The covariance of the *combined innovation* (3.4.2-9) is different (actually smaller than S given in (3.4.2-16)) and is discussed further in Section 3.6 and derived in Subsection 5.2.

3.4.3 The Probabilistic Data Association

3.4.3 The Probabilistic Data Association

To evaluate the association probabilities, the conditioning is broken down into the *past data* Z^{k-1} and the *latest data* $Z(k)$.

Since a probabilistic inference can be made on both the *number* of measurements in the validation region (from the clutter density, if known) as well as on their *location*, this is written out explicitly as follows:

$$\beta_i(k) = P\{\theta_i(k)|Z^k\} = P\{\theta_i(k)|Z(k), m(k), Z^{k-1}\} \quad (3.4.3-1)$$

Using Bayes' formula, the above is rewritten as

$$\beta_i(k) = \frac{1}{c} p[Z(k)|\theta_i(k), m(k), Z^{k-1}] P\{\theta_i(k)|m(k), Z^{k-1}\} \quad i = 0, 1, \dots, m(k) \quad (3.4.3-2)$$

The joint density of the validated measurements conditioned on $\theta_i(k)$, $i \neq 0$, is the product of

- the (assumed) *Gaussian* pdf of the correct (target originated) measurements, and
- the pdf of the incorrect measurements, assumed *uniform* in the validation region whose volume $V(k)$ is given in (3.4.1-6).

The pdf of the correct measurement (with the P_G factor that accounts for restricting the normal density to the validation gate) is

$$p[z_i(k)|\theta_i(k), m(k), Z^{k-1}] = P_G^{-1} \mathcal{N}[z_i(k); \hat{z}(k|k-1), S(k)] = P_G^{-1} \mathcal{N}[\nu_i(k); 0, S(k)] \quad (3.4.3-3)$$

The pdf from (3.4.3-2) is then

$$p[Z(k)|\theta_i(k), m(k), Z^{k-1}] = \begin{cases} V(k)^{-m(k)+1} P_G^{-1} \mathcal{N}[\nu_i(k); 0, S(k)] & i = 1, \dots, m(k) \\ V(k)^{-m(k)} & i = 0 \end{cases} \quad (3.4.3-4)$$

The probabilities of the association events conditioned only on the number of validated measurements are (derivations in Subsection 3.4.10)

$$\begin{aligned} \gamma_i[m(k)] &\triangleq P\{\theta_i(k)|m(k), Z^{k-1}\} = P\{\theta_i(k)|m(k)\} \\ &= \begin{cases} \frac{1}{m(k)} P_D P_G \left[P_D P_G + (1 - P_D P_G) \frac{\mu_F[m(k)]}{\mu_F[m(k)-1]} \right]^{-1} & i = 1, \dots, m(k) \\ (1 - P_D P_G) \frac{\mu_F[m(k)]}{\mu_F[m(k)-1]} \left[P_D P_G + (1 - P_D P_G) \frac{\mu_F[m(k)]}{\mu_F[m(k)-1]} \right]^{-1} & i = 0 \end{cases} \end{aligned} \quad (3.4.3-5)$$

where $\mu_F(m)$ is the probability mass function (pmf) of the number of false measurements (clutter) in the validation region.

3.4.3 The Probabilistic Data Association

Two models can be used for the pmf $\mu_F(m)$ in a volume of interest V :

(i) a **Poisson model** with a certain spatial density λ

$$\mu_F(m) = e^{-\lambda V} \frac{(\lambda V)^m}{m!} \quad (3.4.3-6)$$

(ii) a **diffuse prior model** (discussed in [BL93])

$$\mu_F(m) = \mu_F(m-1) = \delta \quad (3.4.3-7)$$

where the constant δ is irrelevant since it cancels out.

Using the (*parametric*) Poisson model in (3.4.3-5) yields

$$\gamma_i[m(k)] = \begin{cases} P_D P_G [P_D P_G m(k) + (1 - P_D P_G) \lambda V(k)]^{-1} & i = 1, \dots, m(k) \\ (1 - P_D P_G) \lambda V(k) [P_D P_G m(k) + (1 - P_D P_G) \lambda V(k)]^{-1} & i = 0 \end{cases} \quad (3.4.3-8)$$

The (*nonparametric*) diffuse prior (3.4.3-7) yields

$$\gamma_i[m(k)] = \begin{cases} \frac{1}{m(k)} P_D P_G & i = 1, \dots, m(k) \\ 1 - P_D P_G & i = 0 \end{cases} \quad (3.4.3-9)$$

The nonparametric model (3.4.3-9) can be obtained from (3.4.3-8) by setting

$$\lambda = \frac{m(k)}{V(k)} \quad (3.4.3-10)$$

i.e., replacing the Poisson parameter with the **sample spatial density** of the validated measurements. The volume $V(k)$ of the elliptical (i.e., Gaussian based) validation region is given in (3.4.1-6).

3.4.3 The Probabilistic Data Association

The Parametric PDA

Using (3.4.3-8) and (3.4.3-4) with the explicit expression of the Gaussian pdf into (3.4.3-2) yields, after some cancellations, the final equations of the **parametric PDA** with the *Poisson clutter model*

$$\beta_i(k) = \begin{cases} \frac{e_i}{b + \sum_{j=1}^{m(k)} e_j} & i = 1, \dots, m(k) \\ \frac{b}{b + \sum_{j=1}^{m(k)} e_j} & i = 0 \end{cases} \quad (3.4.3-11)$$

where

$$e_i \triangleq e^{-\frac{1}{2}\nu_i(k)'S(k)^{-1}\nu_i(k)} \quad (3.4.3-12)$$

$$b \triangleq \lambda |2\pi S(k)|^{1/2} \frac{1 - P_D P_G}{P_D} \quad (3.4.3-13)$$

The last expression above can be rewritten as

$$b = \left(\frac{2\pi}{\gamma} \right)^{\frac{n_x}{2}} \lambda V(k) c_{n_x}^{-1} \frac{1 - P_D P_G}{P_D} \quad (3.4.3-14)$$

The Nonparametric PDA

The **nonparametric PDA** is the same as above except for replacing $\lambda V(k)$ in (3.4.3-14) by $m(k)$ — this obviates the need to know λ .

Remarks

The “discrimination” capability of the PDA relies on the difference between the *Gaussian* and *uniform* densities.³

The PDA yields association probabilities that depend on where the corresponding innovation lies on the Gaussian pdf’s exponential (3.4.3-12) relative to the other innovations.

The value of the gate probability P_G , if close to unity, is irrelevant: the possible additional validated measurements if P_G is increased will have large innovations and thus low association probabilities.

Comparing to the nearest neighbor standard filter (Section 3.2), the PDAF is an “**all neighbors**” modified filter.

³Additional discrimination capability can be obtained by using feature variables (see Subsection 3.4.7).

3.4.4 Summary of the Algorithm

Figure 3.4.4-1 summarizes one cycle of the PDAF.

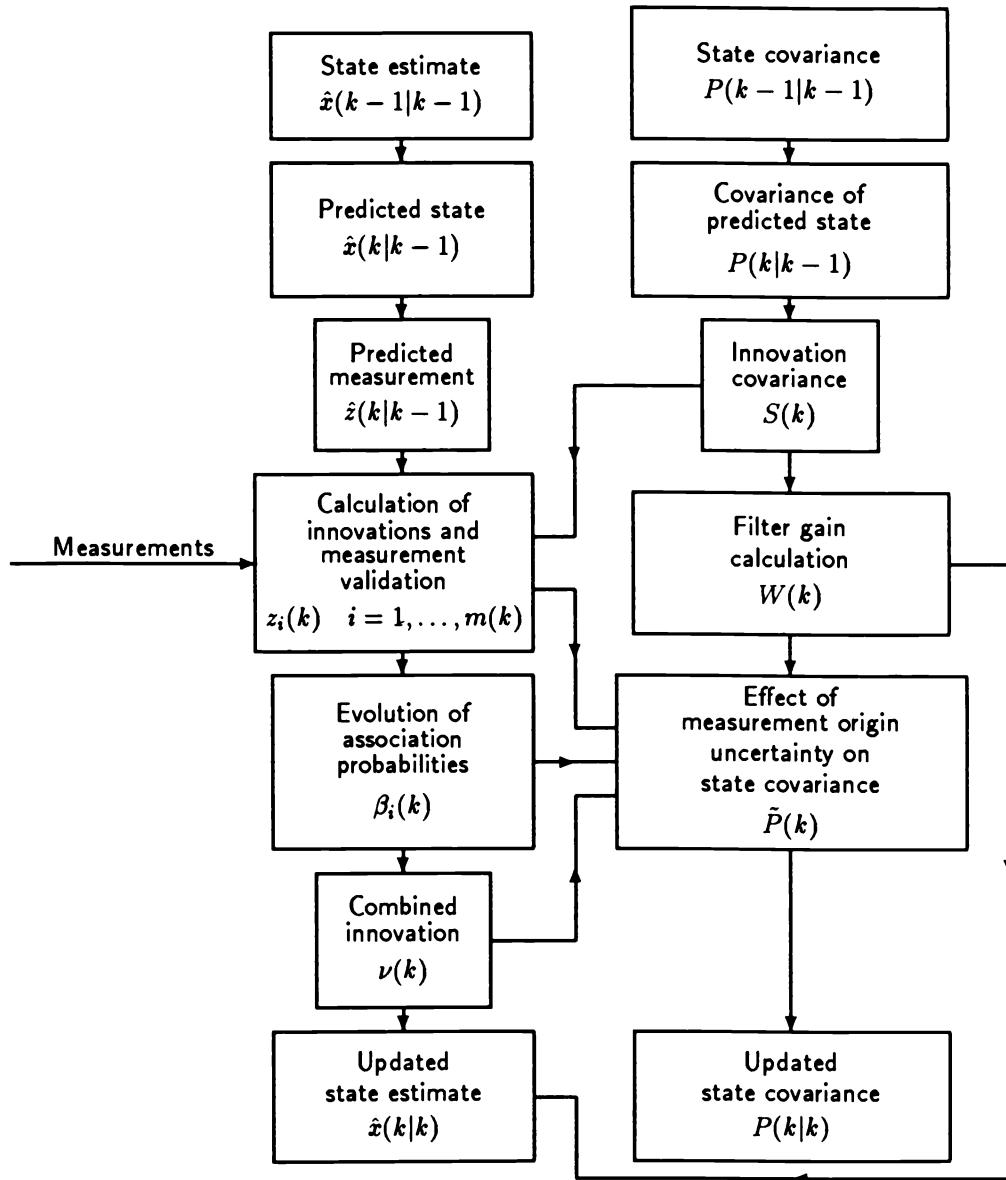


Figure 3.4.4-1: One cycle of the PDAF.

3.4.5 A Nonlinear Multisensor Tracking in Clutter Example

3.4.5 A Nonlinear Multisensor Tracking in Clutter Example

The target considered is a missile moving in a 3-dimensional space with axial thrust. The state in earth-centered inertial Cartesian coordinates is

$$x = [\xi \ \eta \ \zeta \ \dot{\xi} \ \dot{\eta} \ \dot{\zeta} \ \alpha]' \quad (3.4.5-1)$$

where α is the net specific thrust (thrust less drag per unit mass)

$$\alpha = \frac{T - D}{M} \quad (3.4.5-2)$$

modeled by the differential equation

$$\dot{\alpha}(t) = \frac{\alpha(t)^2}{c} \quad (3.4.5-3)$$

with $c = 5km/s$ and $\alpha(0) = 20m/s^2$. The above equation describes the axial acceleration of the missile during the burn period.

The continuous-time dynamic equation of the target is

$$\dot{x}(t) = f[x(t)] \quad (3.4.5-4)$$

where

$$f_1(x) = x_4 = \dot{\xi} \quad (3.4.5-5)$$

$$f_2(x) = x_5 = \dot{\eta} \quad (3.4.5-6)$$

$$f_3(x) = x_6 = \dot{\zeta} \quad (3.4.5-7)$$

$$f_4(x) = \alpha \frac{\dot{\xi} + \Omega\eta}{v_R} - \mu \frac{\xi}{R^3} \quad (3.4.5-8)$$

$$f_5(x) = \alpha \frac{\dot{\eta} - \Omega\xi}{v_R} - \mu \frac{\eta}{R^3} \quad (3.4.5-9)$$

$$f_6(x) = \alpha \frac{\dot{\zeta}}{v_R} - \mu \frac{\zeta}{R^3} \quad (3.4.5-10)$$

with μ the gravitation constant, Ω the spin rate of the earth, and

$$v_R^2 = (\dot{\xi} + \Omega\eta)^2 + (\dot{\eta} - \Omega\xi)^2 + \dot{\zeta}^2 \quad (3.4.5-11)$$

$$R^2 = \xi^2 + \eta^2 + \zeta^2 \quad (3.4.5-12)$$

The target trajectory was generated without process noise.

3.4.5 A Nonlinear Multisensor Tracking in Clutter Example

The measurements were obtained from two passive sensors that measured the azimuth and elevation angles of the line of sight to the target with standard deviation 0.1 mrad. The sensors were located on moving platforms.

The sampling period was 1s and 100 frames were available from each sensor.

An EKF was used to track this target in a “clean” (clutterless) environment. The state equation was integrated numerically between sampling times for prediction of the state. First order linearization of both dynamic and measurement equations was used for covariance calculations. In order to make the resulting filter “consistent,” a process noise equal to a fraction (0.01) of the updated state covariance was used in the EKF. The measurements from the two sensors were processed sequentially according to the algorithm presented in Subsection 2.2.2.

The EKF reached steady-state after about ten measurements. The resulting 99% validation region for the measurements (2 dimensional from each sensor, with the corresponding gating threshold $\gamma = 9.2$ or $g \approx 3$) was called the “standard gate” and used as the basis to define the clutter density. In the simulations false measurements (the “clutter points”) were generated, starting at $k = 10$, with uniform pdf for their location with various spatial densities. The parameter of the study was \bar{r} , the *expected number of false measurements* in the standard gate defined above.

The PDAF used in this example was the *non-parametric* one, i.e., no knowledge of the spatial density of false measurements was needed — a diffuse prior modeled the number of false measurements in the validation region.

The purpose of the simulations was to examine the *tracking capability* of the PDAF in comparison with the NNSF. The *percentage of lost tracks* was estimated from Monte Carlo simulations.

Figure 3.4.5-1 presents the percentage of lost tracks from 50 Monte Carlo runs for the PDAF and the NNSF. A track was considered lost when the correct measurement was not in the 99% validation region of at least one of the sensors for at least the last 20 sampling times. This is a reasonable definition (even though not the only one) of a lost track since in this case the final errors were very large.

The PDAF could track reliably up to a clutter density for which the expected number of false measurements in the standard gate was $\bar{r} = 2$. The average number of false measurements in the PDAF gate was larger because its covariance was increased according to (3.4.2-10). The NNSF had a substantially higher percentage of lost tracks.

3.4.5 A Nonlinear Multisensor Tracking in Clutter Example

Figures 3.4.5-2 and 3.4.5-3 show the average position and velocity estimation errors for the PDAF, respectively, from 50 runs, for two values of \bar{r} , compared to the average error in the clean environment. It is notable that for $\bar{r} = 0.75$ the performance of the PDAF was the same as in the clean environment and is not shown separately. For $\bar{r} = 1.5$ there is a slight degradation and even for $\bar{r} = 2.25$ it is still not very large. On the other hand, the NNSF, for $\bar{r} = 2.25$ has already 40% track loss as indicated in Figure 3.4.5-1.

Therefore, the PDAF can significantly extend the tracking capability into the region of high clutter density where the NNSF becomes unreliable because of its high probability of track loss.

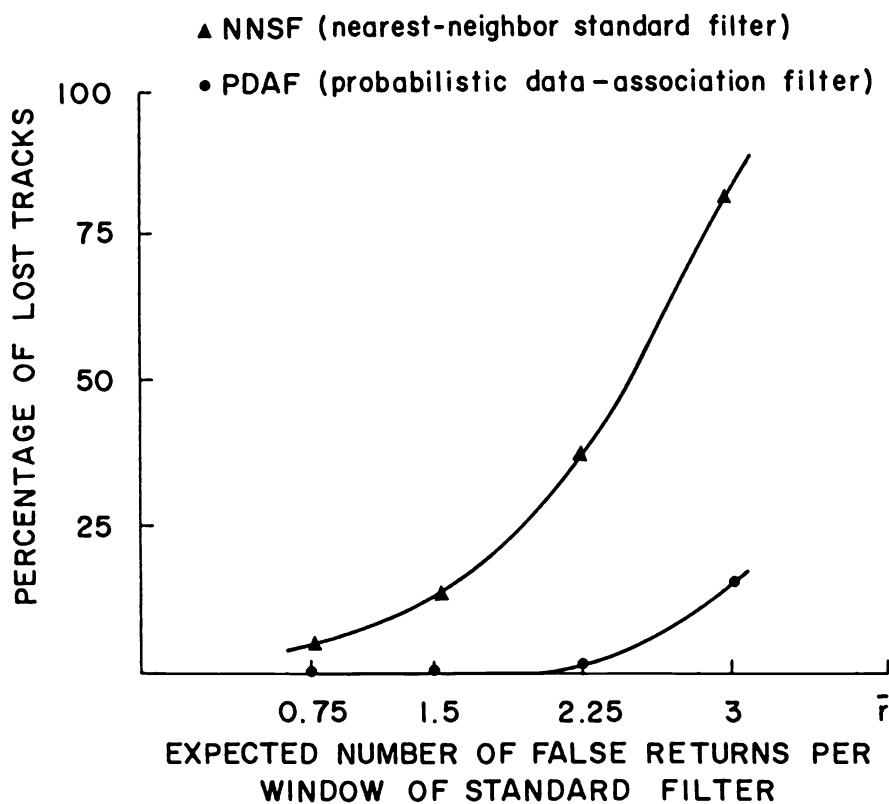


Figure 3.4.5-1: Comparison of tracking capability.

3.4.5 A Nonlinear Multisensor Tracking in Clutter Example

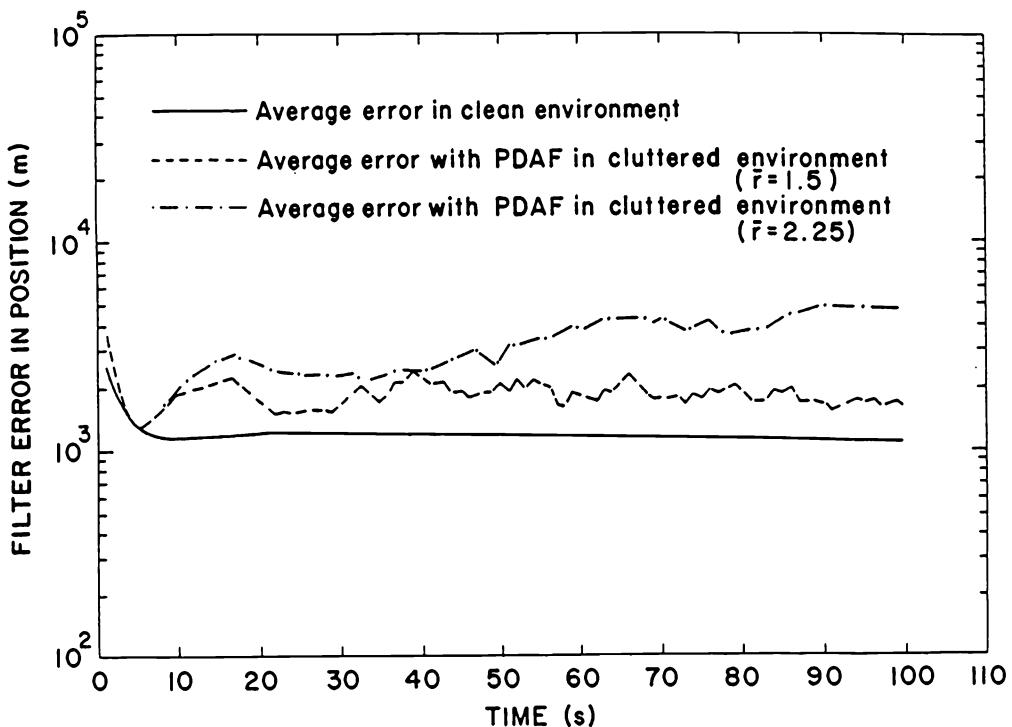


Figure 3.4.5-2: Position estimation errors for the PDAF.

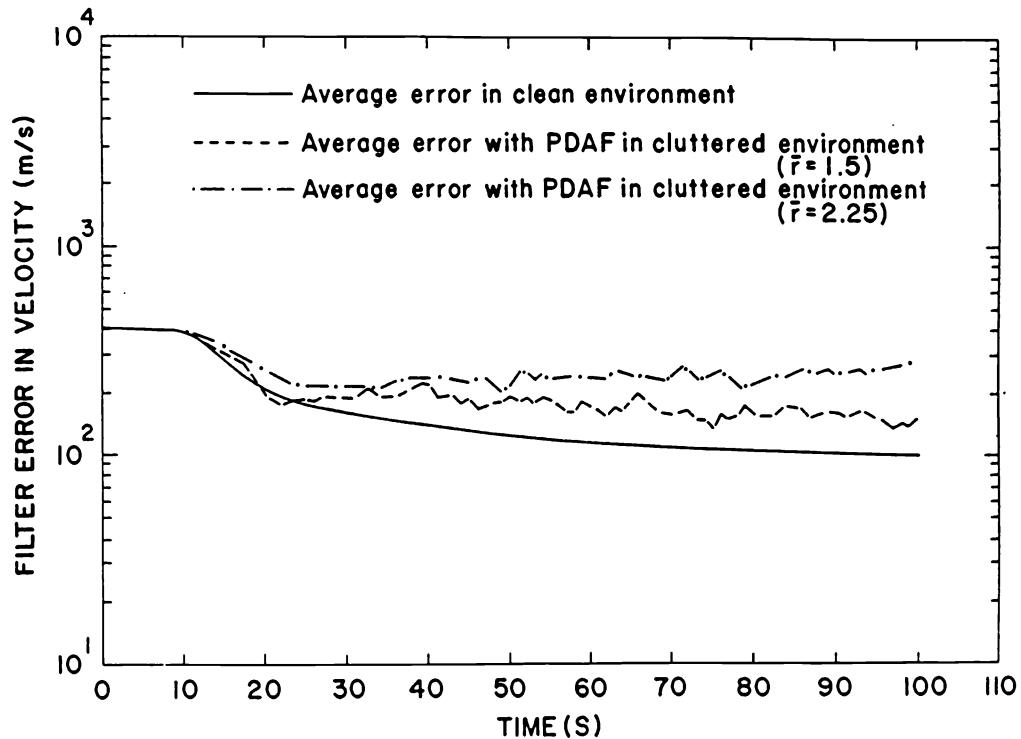


Figure 3.4.5-3: Velocity estimation errors for the PDAF.

3.4.6 Consistency and Robustness Evaluation of a Parametric and Nonparametric PDAF

3.4.6 Consistency and Robustness Evaluation of a Parametric and Nonparametric PDAF

The three consistency criteria for a filter, introduced in [BL93], are

- (a) The normalized state estimation errors should be within a certain region
- (b) The normalized innovations should be within a certain region
- (c) The magnitude of the autocorrelation of the innovations should be below a threshold — the innovations should be ideally white.

The target in track was modeled as a constant velocity object in the plane. The plant equation for this model, with sampling intervals T , is

$$x(k+1) = Fx(k) \quad (3.4.6-1)$$

where, using Cartesian coordinates

$$x = [\xi \ \dot{\xi} \ \eta \ \dot{\eta}] \quad (3.4.6-2)$$

and

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4.6-3)$$

The correct measurements were

$$z(k) = Hx(k) + w(k) \quad (3.4.6-4)$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.4.6-5)$$

and the measurement noise zero-mean with covariance

$$E[w(k)w(j)'] = R\delta_{kj} \quad (3.4.6-6)$$

Generation of the False Measurements

The area of a validation region is, for a two-dimensional measurement

$$V(k) = \pi\gamma|S(k)|^{\frac{1}{2}} \quad (3.4.6-7)$$

where $S(k)$ is the innovation covariance as in (3.4.1-5). In order to obtain a Poisson distributed number of false measurements with λ per unit area,

$$N_F = [10V(k)\lambda + 1]^- \quad (3.4.6-8)$$

false measurements were generated independently and uniformly distributed in a square centered at the correct measurement, known for simulation. The notation $[\cdot]^-$ stands for “rounded down to the nearest integer.” The area of the square was

$$V = \frac{N_F}{\lambda} \quad (3.4.6-9)$$

This large number of false measurements in the area $V \approx 10V(k)$ yielded the actual number of false measurements falling in $V(k)$ approximately Poisson with expected number $V(k)\lambda$ as discussed in Section 2.5. A recently developed exact and efficient technique for doing this is described in [Li92].

Simulation Parameters

The following numerical values were used in the simulation: sampling time $T = 1$, total number of samples $K = 100$, measurement noise covariance $R_{11} = R_{22} = 200m^2$, $R_{12} = 0$. The initial condition was $x(0) = [200m, 0, 10^4m, -15m/s]'$. Initialization was done (assuming a clean environment) with two-point differencing [BL93] and clutter was introduced in the system at $t = 10s$. The target detection probability was assumed $P_D = 1$.

A Monte Carlo simulation with $N = 50$ runs was carried out for a wide range of λ corresponding from low to very dense cluttered environment. The parametric PDAF “knew” the value of the spatial clutter density λ while the nonparametric PDAF *does not need this knowledge*.

The Normalized Estimation Error Squared

State estimation consistency was tested not with a single expression for the entire state as in [BL93] but separately for position and velocity.

The **normalized position error squared** in run i is

$$\epsilon_p^i(k) = \tilde{x}_p^i(k|k)'[P_p^i(k|k)]^{-1}\tilde{x}_p^i(k|k) \quad (3.4.6-10)$$

where x_p is the position subvector from x and P_p the corresponding submatrix from the covariance matrix P .

The N -run average was considered

$$\bar{\epsilon}_p(k) = \frac{1}{N} \sum_{i=1}^N \epsilon_p^i(k) \quad (3.4.6-11)$$

and a similar expression for the **normalized velocity error squared**.

Since the dimension of x_p is 2, $N\bar{\epsilon}_p(k) \sim \chi_{2N}^2$, i.e., chi-square distributed with $2N$ degrees of freedom. With $N = 50$, the 95% probability region for a 100 degrees of freedom chi-square random variable is [75, 130], which, for the average (3.4.6-11), becomes [1.5, 2.6].

The Normalized Innovation Squared

The norm squared of the innovation in run i

$$\epsilon_\nu^i(k) = \nu^i(k)'[S^i(k)]^{-1}\nu^i(k) \quad (3.4.6-12)$$

is, if $\nu^i(k) \sim \mathcal{N}(0, S^i(k))$, chi-square distributed with n_z degrees of freedom. An N -run average was then taken as in (3.4.6-11).

Note, however, that in the PDAF one does not have the “correct” innovation but only the **combined innovation** (3.4.2-9). The matrix $S(k)$ is the *covariance of the correct innovation*.

As shown in Subsection 5.2, the **combined innovation’s covariance** is $q_2 S(k)$, where $0 < q_2 < 1$. The factor q_2 depends on

- 1) P_D , the target’s probability of detection, and
- 2) the expected number of false measurements in the validation gate (of a given size γ), $\lambda V(k)$.

As the number of false measurements increases, the factor q_2 decreases. This is because the smaller innovations receive larger weighting and thus the combined innovation is dominated by small innovations.

In this example the average of the **normalized innovation squared** was taken over the $N = 50$ Monte Carlo runs as well as over the 90 time steps when clutter was present, i.e.,

$$\bar{\epsilon}_\nu = \frac{1}{90} \frac{1}{50} \sum_{i=1}^{50} \sum_{k=11}^{100} \epsilon_\nu^i(k) \quad (3.4.6-13)$$

The quantity $4500\bar{\epsilon}_\nu/q_2$ has a chi-square distribution with $n = 4500n_z = 9000$ degrees of freedom⁴. The standard deviation of a chi-square random variable with n degrees of freedom being $\sqrt{2n}$ [BL93], it follows that the standard deviation of $\bar{\epsilon}_\nu/q_2$ is $\sqrt{18000}/4500 = 0.03$ while its mean is $n_z = 2$.

Thus the 95% probability region for (3.4.6-13) is, based on the normal distribution (because of the very large number of terms averaged, the normal approximation is very good), two standard deviations around the mean. In relative terms this becomes $\pm 0.06/2 = \pm 3\%$.

⁴The factor q_2 is assumed for simplicity time-invariant; this is a reasonable approximation.

The Innovations' Autocorrelation

For test (c) the sample innovation autocorrelation [BL93] was used for $l = 1$, i.e.,

$$\bar{\rho}(k, k+1) = \sum_{i=1}^N \nu^i(k)' \nu^i(k+1) \left[\sum_{i=1}^N \nu^i(k)' \nu^i(k) \sum_{i=1}^N \nu^i(k+1)' \nu^i(k+1) \right]^{-\frac{1}{2}} \quad (3.4.6-14)$$

Tables 3.4.6-1 and 3.4.6-2 provide the results for the two PDAF versions.

Line 2 shows the observed average number κ of false measurements in the PDAF's validation gate with threshold $\gamma = 16$ for various values of the spatial clutter density λ . The difference in the gate sizes for the two filters and, hence, in the corresponding value of κ , was negligible.

Line 3 shows the normalized innovations (3.4.6-13) and they fit their theoretical average (line 4) within 2%. The values of q_2 are based on Figure 5.2.2-1. Line 5 shows that the parametric PDAF works up to about an average of 6.75 false measurements per gate $\gamma = 16$ while the non-parametric PDAF's upper limit is about 6; beyond these limits both filters' performance deteriorated drastically. The corresponding density in a γ' gate is obtained by multiplying this with γ'/γ (because the dimension of the measurement is 2). Thus in the $\gamma' = 9$ (99%) gate both filters worked up to about 3.5 false measurements — a quite heavy clutter environment.

Figure 3.4.6-1 shows the average normalized position error for three ranges of clutter — low, medium and high. Figure 3.4.6-2 shows the average normalized velocity error versus time for these clutter ranges. The normalized errors stay within their 95% probability regions delineated in the figures by the region lower and upper bound lines.

Figure 3.4.6-3 shows the autocorrelation $\bar{\rho}(k, k+1)$ for the combined innovations' whiteness test. Since the autocorrelation is well within the 95% probability region the innovations can be accepted as white.

Conclusion

Based on these results, the non-parametric PDAF, which exhibits robustness in a wide range of clutter, is the preferred one.

3.4.6 Consistency and Robustness Evaluation of a Parametric and Nonparametric PDAF

	1	2	5	10	20	30	40	45	50
Spatial density of false measurements [$\lambda = (\cdot) \times 10^{-5}$]									
Sample average κ of number of false measurements in gate $\gamma = 16$.11	.22	.56	1.14	2.36	3.65	5.08	5.96	6.76
Sample average of normalized innovation $\bar{\epsilon}_\nu$	1.93	1.88	1.74	1.50	1.14	.90	.73	.65	.59
Theoretical value of normalized innovation ($2q_2$)	1.92	1.84	1.73	1.50	1.14	.92	.74		
Percentage of lost tracks (out of 50)	2	2	0	0	0	0	0	4	2

Table 3.4.6-1: Results for the parametric PDAF.

	1	2	5	10	20	30	40	45	50
Spatial density of false measurements [$\lambda = (\cdot) \times 10^{-5}$]									
Sample average κ of number of false measurements in gate $\gamma = 16$.11	.22	.56	1.14	2.36	3.67	5.07	5.94	6.69
Sample average of normalized innovation $\bar{\epsilon}_\nu$	1.92	1.87	1.71	1.47	1.14	.91	.74	.64	-
Theoretical value of normalized innovation ($2q_2$)	1.92	1.84	1.73	1.50	1.14	.92	.74		
Percentage of lost tracks (out of 50)	2	2	0	0	0	0	2	0	-

Table 3.4.6-2: Results for the nonparametric PDAF.

3.4.6 Consistency and Robustness Evaluation of a Parametric and Nonparametric PDAF

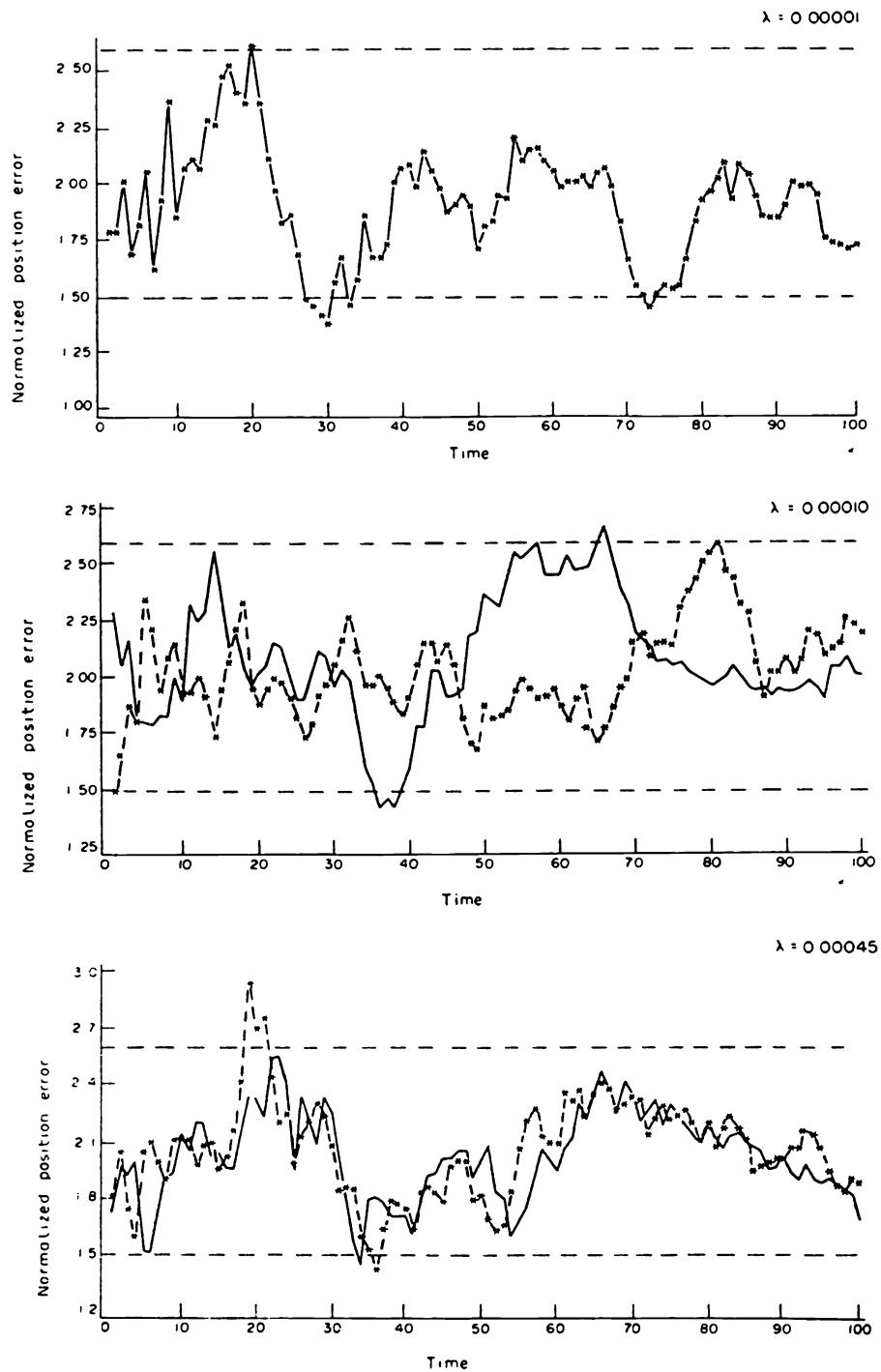


Figure 3.4.6-1: Average normalized position error squared from 50 runs (Nonparametric PDAF:
-*-- Parametric PDAF — 95% probability region bounds - - -).

3.4.6 Consistency and Robustness Evaluation of a Parametric and Nonparametric PDAF

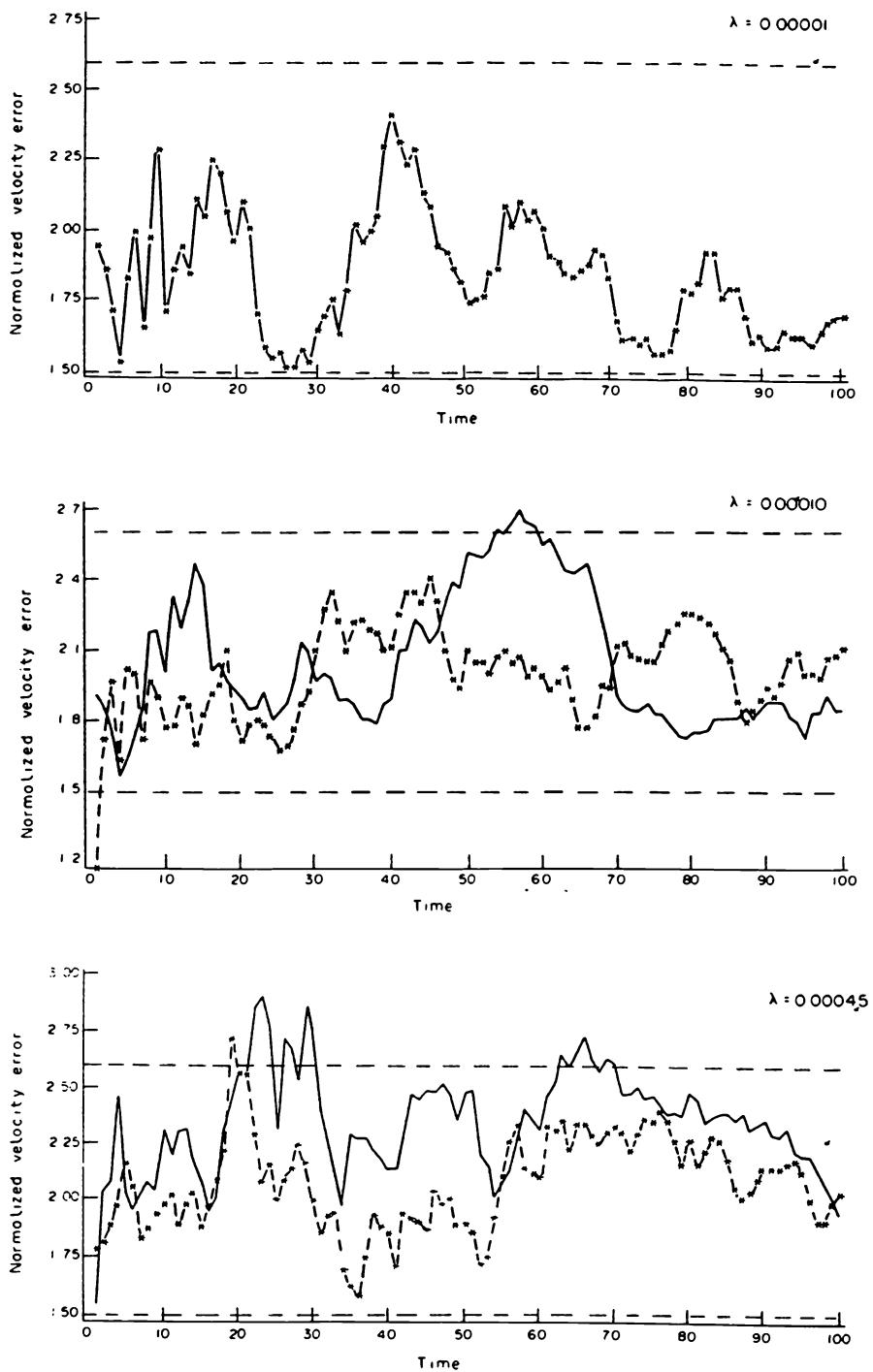


Figure 3.4.6-2: Average normalized velocity error squared from 50 runs (Nonparametric PDAF: $-*--*$ Parametric PDAF — 95% probability region bounds $---$).

3.4.6 Consistency and Robustness Evaluation of a Parametric and Nonparametric PDAF

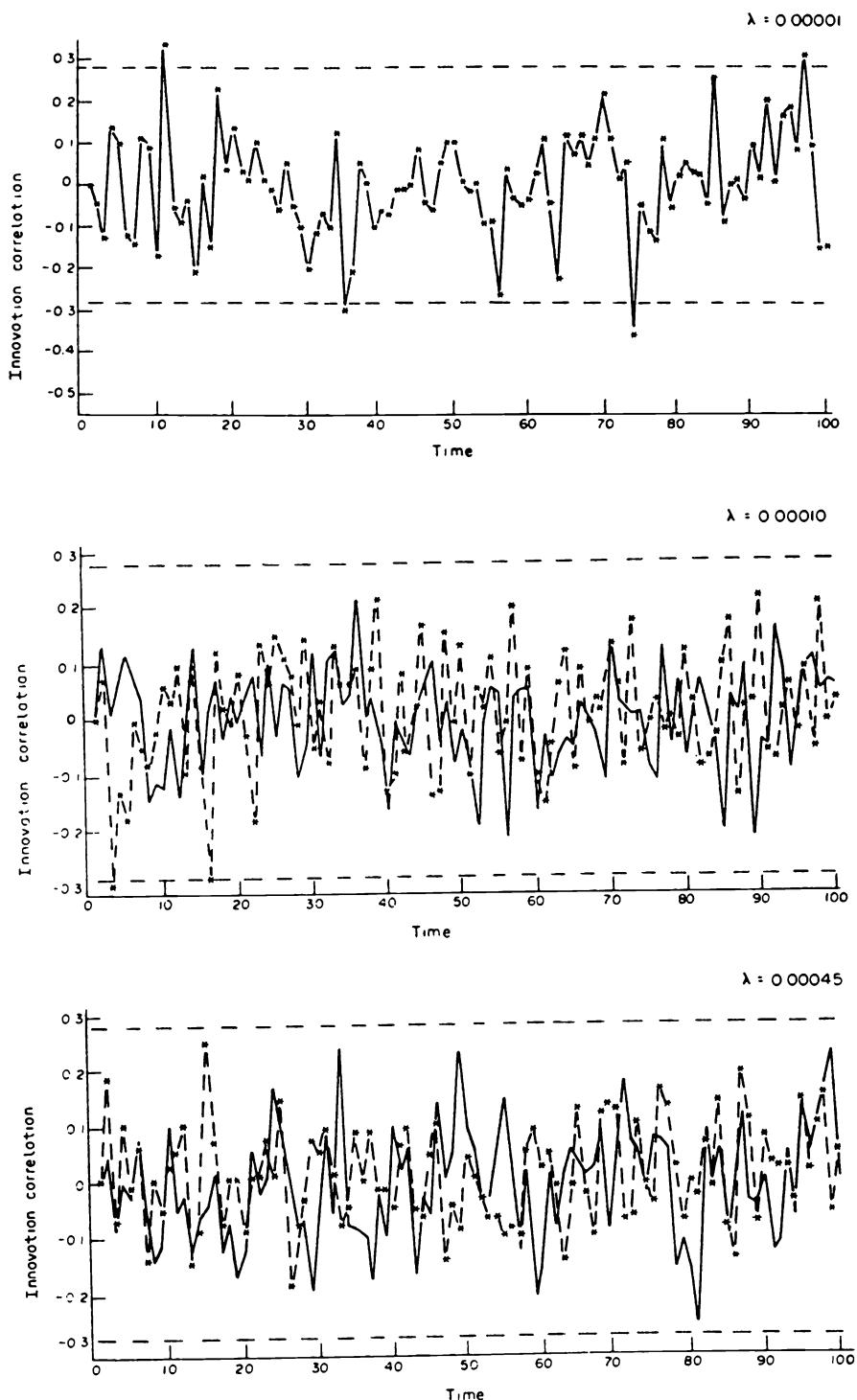


Figure 3.4.6-3: Autocorrelation of combined innovations from 50 runs (Nonparametric PDAF: $-*--*$ Parametric PDAF — 95% probability region bounds $--$).

3.4.7 Augmented Probabilistic Data Association with Feature Measurements

Assume that in addition to the usual measurements of kinematic components (position, radial velocity) one has measurements of a **feature** of the target, which, while not related to its state, can be used to discriminate against the undesired measurements.

The most natural feature variable is the target intensity (or radar cross-section). This is used implicitly in all systems in a simple way: thresholding at the output of the detector amounts to a “binarization” of the measured target intensity. In the PDA this can be used in a more elaborate way.

Define the augmented measurement vector $z_a(k)$ at time k consisting of the kinematic variable z and the feature variable r

$$z_a(k) = \begin{bmatrix} z(k) \\ r(k) \end{bmatrix} \quad (3.4.7-1)$$

The **augmented PDA** procedure is the same as described in Subsection 3.4.4 with the following modifications that incorporate the *probabilistic description of the feature component(s) of the measurement vector*.

Assume one has the **feature pdf** (or pmf, if discrete)

$$p(r|\theta) = \begin{cases} p_C(r) & \text{if } \theta = \theta_C \\ p_T(r) & \text{if } \theta = \theta_T \end{cases} \quad (3.4.7-2)$$

where θ_C denotes clutter origin and θ_T denotes target origin.

The pdf of a correct measurement, which was (3.4.3-3), is now

$$\begin{aligned} p[z_{a_i}(k)|\theta_i(k), m(k), Z^{k-1}] &= p[z_i(k), r_i(k)|\theta_i(k), m(k), Z^{k-1}] \\ &= P_G^{-1} \mathcal{N}[\nu_i(k); 0, S(k)] p_T[r_i(k)] \end{aligned} \quad (3.4.7-3)$$

The pdf of a false measurement, which was $V(k)^{-1}$, is now

$$\begin{aligned} p[z_{a_i}(k)|\theta_0(k), m(k), Z^{k-1}] &= p[z_i(k), r_i(k)|\theta_0(k), m(k), Z^{k-1}] \\ &= V(k)^{-1} p_C[r_i(k)] \end{aligned} \quad (3.4.7-4)$$

The Augmented PDA

Denote the **target to clutter feature likelihood ratio**

$$L_{T/C}(r) \triangleq \frac{p_T(r)}{p_C(r)} \quad (3.4.7-5)$$

The association probabilities are the same as in (3.4.3-11), except for the exponential (3.4.3-12) being replaced by

$$e_i \triangleq e^{-\frac{1}{2}\nu_i(k)'S(k)^{-1}\nu_i(k)} L_{T/C}[r_i(k)] \quad (3.4.7-6)$$

Remarks

The two versions of the PDA — parametric and nonparametric — carry over. As before, the nonparametric one is preferred since it requires less information.

The discrimination capability of the augmented PDA is based on the difference between

- the *Gaussian* location and *target feature pdf* vs.
- the *uniform* location and *clutter (false measurement) feature pdf*.

The “sharper” the difference between the feature pdfs, the more powerful the augmented PDA becomes.

The key in using the augmented PDA is the modeling of the feature via the two pdfs required in the likelihood ratio (3.4.7-5).

If the feature is the signal intensity, then the average SNR is needed to specify the two feature pdfs.

Applications of the PDA procedure augmented with various features can be found in Sections 4.2, 4.6.

3.4.8 The PDAF — Summary

Assumptions of the PDAF:

- There is a *single* target of interest to be tracked in the presence of *false measurements*
- The track of this target has been initialized
- A validation region about the predicted measurement is set up at each sampling time
- At most one of the validated measurements can be target-originated (if the target is detected)
- The detection of the target occurs independently from sample to sample with a known probability, which can be time-varying
- The measurements that did not originate from the target (false alarms or “residual” clutter) are independently and uniformly distributed in the measurement space
- The conditional pdf of the state of the target given the *past measurements* is assumed *Gaussian with given mean and covariance* — the predicted state and the associated covariance are the *approximate sufficient statistic* summarizing the past.

The algorithm’s capability of distinguishing (probabilistically) between the correct measurement and the false ones resides in the former having a *Gaussian* pdf while the latter are *uniformly distributed*. The algorithm consists of the following:

- The probability of each *current* measurement in the validation region having originated from the target is calculated (PDA)
- The probability of a measurement being correct is:
 - the ratio of the “height” on the predicted measurement’s Gaussian pdf of its innovation and
 - the sum of such terms for all the validated measurements plus a constant (which accounts for the possibility that none of the validated measurements is correct)
- The state estimate is updated with *all the validated measurements* weighted by their probabilities of having originated from the target — a *combined innovation* is used in the filter (PDAF)
- The covariance update equation contains an extra term, which is positive semidefinite, that accounts for the additional *measurement uncertainty*
- The additional term in the covariance update equation is *measurement dependent* — the estimation accuracy is not precomputable as in the linear filter case

3.4.8 The PDAF — Summary

- The state and measurement prediction and the filter gain calculation are done as in the standard filter — the only difference is in the use of the combined innovation in the state update and the increased covariance of the updated state.

There are two versions of the PDA:

- *Nonparametric* — a “diffuse prior” is used for the number of false measurements — this requires no knowledge about the spatial density of false measurements
- *Parametric* — a Poisson pmf is used for the number of false measurements — this requires the spatial density (expected number per unit volume) of the false measurements.

The PDAF can work in an environment with up to about 5 false measurements in a $g = 4$ validation region.

The computational requirements of the PDAF are up to about double compared to the standard filter.

The Augmented PDA

If

- feature information can be obtained from the sensor in addition to the kinematic measurements, and
- a probabilistic model is available to distinguish between the target and clutter originated feature measurements (e.g., via the SNR)

one can enhance the PDA with the use of this information via the *target to clutter feature likelihood ratio* which modifies the association probabilities.

3.4.9 The Covariance Update in the PDAF

3.4.9 The Covariance Update in the PDAF

The covariance of the updated state (3.4.2-8) is

$$\begin{aligned} P(k|k) &= E\{[x(k) - \hat{x}(k|k)][x(k) - \hat{x}(k|k)]' | Z^k\} \\ &= \sum_{i=0}^{m(k)} E\{[x(k) - \hat{x}(k|k)][x(k) - \hat{x}(k|k)]' | \theta_i(k), Z^k\} \beta_i(k) \\ &= \bar{P}(k|k) + \tilde{P}(k) \end{aligned} \quad (3.4.9-1)$$

where

$$\bar{P}(k|k) \triangleq \sum_{i=0}^{m(k)} \beta_i(k) P_i(k|k) \quad (3.4.9-2)$$

$$\tilde{P}(k) \triangleq \sum_{i=0}^{m(k)} \beta_i(k) \hat{x}_i(k|k) \hat{x}_i(k|k)' - \hat{x}(k|k) \hat{x}(k|k)' \quad (3.4.9-3)$$

For $i = 0$ one has the prediction covariance while for $i \neq 0$ one has the same updated covariance:

$$P_0(k|k) = P(k|k-1) \quad (3.4.9-4)$$

$$P_i(k|k) = P^c(k|k) = P(k|k-1) - W(k)S(k)^{-1}W(k)' \quad i = 1, \dots, m(k) \quad (3.4.9-5)$$

Using (3.4.9-2), (3.4.9-3) and the identity

$$\beta_0(k) = 1 - \sum_{i=1}^{m(k)} \beta_i(k) \quad (3.4.9-6)$$

yields for the term (3.4.9-2)

$$\bar{P}(k|k) = \beta_0(k)P(k|k-1) + [1 - \beta_0(k)]P^c(k|k) \quad (3.4.9-7)$$

The “spread of the means” term (3.4.9-3) can be rewritten as follows

$$\begin{aligned} \tilde{P}(k) &= \sum_{i=0}^{m(k)} \beta_i(k) \hat{x}_i(k|k) \hat{x}_i(k|k)' - \hat{x}(k|k) \hat{x}(k|k)' \\ &= \sum_{i=0}^{m(k)} \beta_i(k) [\hat{x}(k|k-1) + W(k)\nu_i(k)] [\hat{x}(k|k-1) + W(k)\nu_i(k)]' \\ &\quad - [\hat{x}(k|k-1) + W(k)\nu(k)] [\hat{x}(k|k-1) + W(k)\nu(k)]' \end{aligned} \quad (3.4.9-8)$$

3.4.9 The Covariance Update in the PDAF

Use will be made of the identity (3.4.9-6) rewritten as

$$\sum_{i=0}^{m(k)} \beta_i(k) = 1 \quad (3.4.9-9)$$

and the fact that

$$\sum_{i=0}^{m(k)} \beta_i(k) \nu_i(k) = \sum_{i=1}^{m(k)} \beta_i(k) \nu_i(k) = \nu(k) \quad (3.4.9-10)$$

since, by definition,

$$\nu_0(k) = 0 \quad (3.4.9-11)$$

With this, (3.4.9-8) becomes

$$\begin{aligned} \tilde{P}(k) &= \hat{x}(k|k-1)\hat{x}(k|k-1)' + \hat{x}(k|k-1)[W(k)\nu(k)]' + W(k)\nu(k)\hat{x}(k|k-1)' \\ &\quad + \sum_{i=0}^{m(k)} \beta_i(k)W(k)\nu_i(k)[W(k)\nu_i(k)]' - \hat{x}(k|k-1)\hat{x}(k|k-1)' \\ &\quad - \hat{x}(k|k-1)[W(k)\nu(k)]' - W(k)\nu(k)\hat{x}(k|k-1)' - W(k)\nu(k)[W(k)\nu(k)]' \end{aligned} \quad (3.4.9-12)$$

After cancellations, the above becomes

$$\tilde{P}(k) = W(k) \left[\sum_{i=0}^{m(k)} \beta_i(k) \nu_i(k) \nu_i(k)' - \nu(k) \nu(k)' \right] W(k)' \quad (3.4.9-13)$$

Finally, inserting (3.4.9-7) and (3.4.9-13) into (3.4.9-1) yields (3.4.2-10).

Note that the (weighted) **spread of the innovations** term \tilde{P} given by (3.4.9-3), with the alternative expression (3.4.9-13), is the equivalent of the spread of the means from the covariance of a mixture [BL93]. The only difference is that here the innovations are weighted by the filter gain.

Since one has a sum of dyads with positive weightings, it is *positive semidefinite*, i.e., it increases the covariance to reflect the uncertainty in the origin of the measurements.

3.4.10 Probability Evaluations for the PDA

The probability $\gamma_i(m)$ of event θ_i (time index k is dropped) conditioned on the total number of validated measurements $\mathbf{m} = m$ will be evaluated. In this notation \mathbf{m} is the random variable and m its realization. Denoting by ϕ the number of false measurements, one has

$$\begin{aligned}
 \gamma_i(m) &= P\{\theta_i | \mathbf{m} = m\} \\
 &= P\{\theta_i | \phi = m - 1, \mathbf{m} = m\} P\{\phi = m - 1 | \mathbf{m} = m\} \\
 &\quad + P\{\theta_i | \phi = m, \mathbf{m} = m\} P\{\phi = m | \mathbf{m} = m\} \\
 &= \begin{cases} (1/m)P\{\phi = m - 1 | \mathbf{m} = m\} + (0)P\{\phi = m | \mathbf{m} = m\} & i = 1, \dots, m \\ (0)P\{\phi = m - 1 | \mathbf{m} = m\} + (1)P\{\phi = m | \mathbf{m} = m\} & i = 0 \end{cases} \tag{3.4.10-1}
 \end{aligned}$$

because ϕ must be either $m - 1$ (if the target has been detected and its measurement fell in the validation gate) or m .

Using Bayes' formula one has

$$P\{\phi = m - 1 | \mathbf{m} = m\} = \frac{P\{\mathbf{m} = m | \phi = m - 1\} P\{\phi = m - 1\}}{P\{\mathbf{m} = m\}} = \frac{P_D P_G \mu_F(m - 1)}{P\{\mathbf{m} = m\}} \tag{3.4.10-2}$$

where μ_F is the probability mass function (pmf) of the number of false measurements and $P_D P_G$ is the probability that the target has been detected and its measurements fell in the gate. Similarly,

$$P\{\phi = m | \mathbf{m} = m\} = \frac{P\{\mathbf{m} = m | \phi = m\} P\{\phi = m\}}{P\{\mathbf{m} = m\}} = \frac{(1 - P_D P_G) \mu_F(m)}{P\{\mathbf{m} = m\}} \tag{3.4.10-3}$$

The common denominator in (3.4.10-2) and (3.4.10-3) is

$$P\{\mathbf{m} = m\} = P_D P_G \mu_F(m - 1) + (1 - P_D P_G) \mu_F(m) \tag{3.4.10-4}$$

Combining (3.4.10-2) through (3.4.10-4) into (3.4.10-1) yields (3.4.3-5).

3.4.11 Probabilistic Data Association Filter Overview

3.4.11 Probabilistic Data Association Filter Overview

Modeling Assumptions of the Probabilistic Data Association Filter (PDAF)

- There is only one target of interest modeled by white-noise driven state equation as in the KF/EKF
- Its track has been initialized
- At each time a validation region (gate) to select the measurement(s) to be used for state update is set up
- At most one of the measurements can be target-originated
- The target is detected independently at each scan with probability P_D
- The remaining measurements are assumed due to false alarm or clutter and are modeled as i.i.d. random variables with uniform spatial distribution
- The number of false measurements obeys a diffuse pmf — any number is equally likely (this yields the non-parametric PDA; a Poisson pmf with known mean yields the parametric PDA)
- The past at time k is summarized by a single “lumped” state estimate for the target $\hat{x}(k-1|k-1)$ and the associated covariance $P(k-1|k-1)$, which serve as a *quasi-sufficient statistic*
- The pdf of the target-originated measurement is Gaussian with mean $\hat{z}(k|k-1)$ and covariance $S(k)$ (the innovation covariance) based on the above.

The PDAF Algorithm

PREDICTION

Prediction of the state and measurement

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) \quad (3.4.11-1)$$

$$\hat{z}(k+1|k) = H(k+1)\hat{x}(k+1|k) \quad (3.4.11-2)$$

Covariance of the predicted state

$$P(k+1|k) = F(k)P(k|k)F(k)' + Q(k) \quad (3.4.11-3)$$

Innovation covariance corresponding to the true measurement

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1) \quad (3.4.11-4)$$

All the above are exactly the same as in the standard Kalman filter.

3.4.11 Probabilistic Data Association Filter Overview

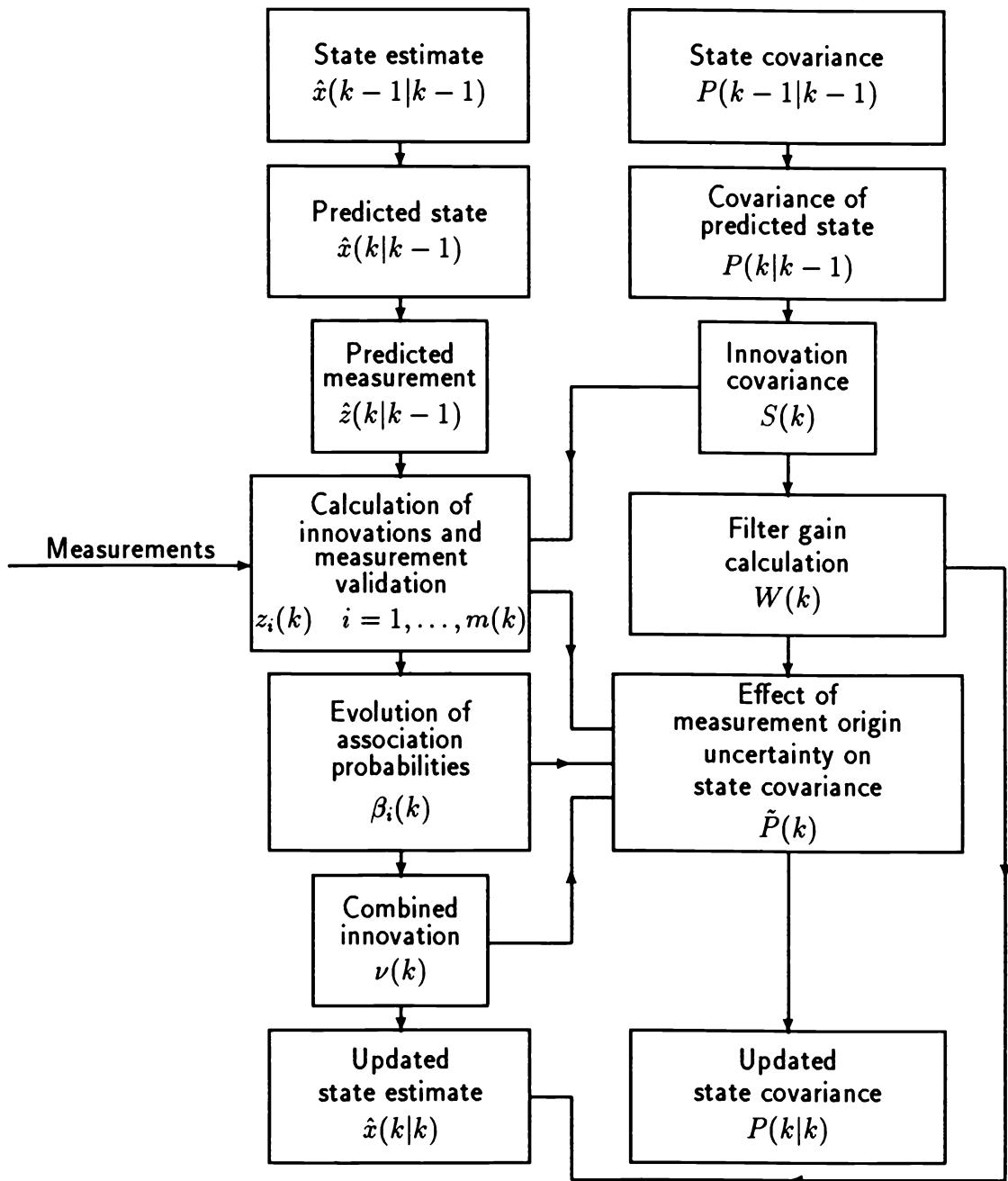


Figure 3.4.11-1: One cycle of the PDAF.

3.4.11 Probabilistic Data Association Filter Overview

MEASUREMENT SELECTION FOR UPDATE

The validation region (gate) — the ellipsoid

$$\mathcal{V}(k+1) = \{z : [z - \hat{z}(k+1|k)]' S(k+1)^{-1} [z - \hat{z}(k+1|k)] \leq \gamma\} \quad (3.4.11-5)$$

where γ is the gate threshold determined by the chosen *gate probability* P_G .

Validated measurements

$$\{z_i(k+1), i = 1, \dots, m(k+1)\} \quad (3.4.11-6)$$

Innovation corresponding to the i -th validated measurement

$$\nu_i(k+1) = z_i(k+1) - \hat{z}(k+1|k) \quad i = 1, \dots, m(k+1) \quad (3.4.11-7)$$

Volume of the validation region

$$V(k+1) = c_{n_z} |\gamma S(k+1)|^{1/2} = c_{n_z} \gamma^{\frac{n_z}{2}} |S(k+1)|^{1/2} \quad (3.4.11-8)$$

where c_{n_z} is the volume of the n_z -dimensional unit hypersphere ($c_1 = 2$, $c_2 = \pi$, $c_3 = 4\pi/3$, etc.).

PROBABILISTIC DATA ASSOCIATION (PDA) — NONPARAMETRIC VERSION

Probability that the i -th validated measurement is the correct one

$$\beta_i(k+1) = \begin{cases} \frac{e_i}{b + \sum_{j=1}^{m(k+1)} e_j} & i = 1, \dots, m(k+1) \\ \frac{b}{b + \sum_{j=1}^{m(k+1)} e_j} & i = 0 \end{cases} \quad (3.4.11-9)$$

where $\beta_0(k+1)$ is the probability that none of the measurements is correct,

$$e_i \triangleq e^{-\frac{1}{2} \nu_i(k+1)' S(k+1)^{-1} \nu_i(k+1)} \quad (3.4.11-10)$$

$$b \triangleq \left(\frac{2\pi}{\gamma}\right)^{\frac{n_z}{2}} m(k+1) c_{n_z}^{-1} \frac{1 - P_D P_G}{P_D} \quad (3.4.11-11)$$

3.4.11 Probabilistic Data Association Filter Overview

UPDATE

State update

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)\nu(k+1) \quad (3.4.11-12)$$

with the *combined innovation*

$$\nu(k+1) \triangleq \sum_{i=1}^{m(k+1)} \beta_i(k+1)\nu_i(k+1) \quad (3.4.11-13)$$

Covariance associated with the updated state

$$P(k+1|k+1) = P(k+1|k) - [1 - \beta_0(k+1)]W(k+1)S(k+1)W(k+1)' + \tilde{P}(k+1) \quad (3.4.11-14)$$

where the (weighted) *spread of the innovations* term is

$$\tilde{P}(k+1) \triangleq W(k+1) \left[\sum_{i=1}^{m(k+1)} \beta_i(k+1)\nu_i(k+1)\nu_i(k+1)' - \nu(k+1)\nu(k+1)' \right] W(k+1)' \quad (3.4.11-15)$$

Remarks

Since it is not known which of the $m(k+1)$ validated measurements is correct, the spread of the innovations term (3.4.11-15), which is *positive semidefinite*, increases the covariance of the updated state — this is the effect of the measurement origin uncertainty.

The estimation accuracy is *data dependent* (typical of nonlinear filters).

In the parametric PDA, which assumes the spatial density of the false measurements to be λ , the number $m(k+1)$ in (3.4.11-11) is replaced by $\lambda V(k+1)$.

Computational requirements

About 50% more than the KF.

3.5 THE OPTIMAL BAYESIAN APPROACH

3.5.1 The Optimal Approach

The main difference between the optimal method and the PDAF is:

- the decomposition of the state estimate (according to the total probability theorem) is done w.r.t *all combinations of measurements from the initial to the present time* rather than only the latest set of measurements.

As in the PDAF, it is assumed that

- there is only one target of interest, modeled by (3.1.3-1), (3.1.3-2),
- the environment contains an arbitrary number of observations due to random clutter or false alarms.

A sequence of measurements or **measurement history** up to time k is denoted as

$$Z^{k,l} = \{Z^{k-1,s}, z_i(k)\} \quad (3.5.1-1)$$

where $Z^{k-1,s}$ is the “prior part” of history $Z^{k,l}$ — the **ancestor** of $z_i(k)$ in $Z^{k,l}$. The most recent element, $z_i(k)$, is the “offspring” or “continuation” of $Z^{k-1,s}$ within $Z^{k,l}$.

The total number of measurement histories at time k is

$$L(k) = \prod_{j=1}^k [1 + m(j)] \quad (3.5.1-2)$$

where $m(j)$ is the number of measurements at time j . The possibility that none of the measurements is correct is also accounted for in (3.5.1-2).

3.5.2 The Estimation

3.5.2 The Estimation

The event that the l -th history at time k is the correct sequence of measurements is denoted as $\theta^{k,l}$ and its probability conditioned on all the data up to time k as

$$\beta^{k,l} = P\{\theta^{k,l}|Z^k\} \quad (3.5.2-1)$$

Since the events $\theta^{k,l}$, $l = 1, \dots, L(k)$, are mutually exclusive and exhaustive, the conditional mean of the state at time k can be expressed as

$$\hat{x}(k|k) = E[x(k)|Z^k] = \sum_{l=1}^{L(k)} E[x(k)|\theta^{k,l}, Z^k] P\{\theta^{k,l}|Z^k\} \quad (3.5.2-2)$$

or, with $\hat{x}^l(k|k)$ denoting the **history-conditioned estimate**,

$$\boxed{\hat{x}(k|k) = \sum_{l=1}^{L(k)} \hat{x}^l(k|k) \beta^{k,l}} \quad (3.5.2-3)$$

For each history a *standard filter* provides this estimate:

$$\boxed{\hat{x}^l(k|k) = \hat{x}^s(k|k-1) + W^l(k)[z_i(k) - \hat{z}^s(k|k-1)]} \quad (3.5.2-4)$$

where $z_i(k)$ is the latest element of sequence l and $\hat{z}^s(k|k-1)$ is the predicted measurement corresponding to its ancestor $\theta^{k-1,s}$, with covariance $S^s(k)$.

The gain in (3.5.2-4) is

$$\boxed{W^l(k) = P^s(k|k-1)H(k)[S^s(k)]^{-1}} \quad (3.5.2-5)$$

The covariance of the history-conditioned updated state (3.5.2-4)

$$\boxed{P^l(k|k) = E\{[x(k) - \hat{x}^l(k|k)][x(k) - \hat{x}^l(k|k)]'|\theta^{k,l}, Z^k\}} \quad (3.5.2-6)$$

is given by the standard equation

$$\boxed{P^l(k|k) = P^s(k|k-1) - W^l(k)S^s(k)W^l(k)'} \quad (3.5.2-7)$$

The covariance associated with the **combined estimate** (3.5.2-3) is, according to [BL93],

$$\boxed{P(k|k) = \sum_{l=1}^{L(k)} \beta^{k,l} P^l(k|k) + \sum_{l=1}^{L(k)} \beta^{k,l} \mu^l(k) [\mu^l(k)]'} \quad (3.5.2-8)$$

where

$$\mu^l(k) \triangleq \hat{x}^l(k|k) - \hat{x}(k|k) \quad (3.5.2-9)$$

3.5.2 The Estimation

Remark

The number of histories considered in the estimation equation (3.5.2-3), $L(k)$, given in (3.5.1-2), indicates that the memory and computation requirements *increase with time* — this is the price for optimality in the present problem. Suboptimal versions that are intermediate between this and the PDAF from Section 3.4 will be discussed later.

Measurement Validation

At time k , prior to obtaining the latest set of measurements, $Z(k)$, one has the predicted state $\hat{x}(k|k-1)$, the associated covariance $P(k|k-1)$, as well as the predicted measurement $\hat{z}(k|k-1)$ obtained using the standard prediction equations.

The latest set of validated measurements, $Z(k)$, consists of the measurements falling in the validation region set up around $\hat{z}(k|k-1)$.

The set of histories through time k consists of *all the combinations* of the histories through $k-1$ with the latest validated measurements.

The evaluation of the probabilities of each sequence of validated measurements having originated from the target of interest is presented next.

3.5.3 Evaluation of the Probabilities of Histories

The vector whose elements are the number of validated measurements at each time through k is denoted as

$$m^k = [m(1) \quad \cdots \quad m(k)]' \quad (3.5.3-1)$$

The computation of the probability $\beta^{k,l}$, that the l -th measurement history through time k is correct, with the conditioning on m^k written out explicitly,

$$\beta^{k,l} = P\{\theta^{k,l}|Z^k, m^k\} = P\{\theta_i(k), \theta^{k-1,s}|Z(k), m(k), Z^{k-1}, m^{k-1}\} \quad (3.5.3-2)$$

is obtained using Bayes' formula as

$$\boxed{\beta^{k,l} = \frac{1}{c} p[Z(k)|\theta_i(k), m(k), \theta^{k-1,s}, Z^{k-1}] P\{\theta_i(k)|m(k), \theta^{k-1}, Z^{k-1}, m^{k-1}\} \beta^{k-1,s}} \quad (3.5.3-3)$$

where c is the normalization constant.

Similarly to (3.4.3-4), the joint pdf of the validated measurements at k is

$$\begin{aligned} p[Z(k)|\theta_i(k), m(k), \theta^{k-1,s}, Z^{k-1}, m^{k-1}] \\ = \begin{cases} V(k)^{-m(k)+1} P_G^{-1} \mathcal{N}[z_i(k); \hat{z}^s(k|k-1), S^s(k)] & i = 1, \dots, m(k) \\ V(k)^{-m(k)} & i = 0 \end{cases} \end{aligned} \quad (3.5.3-4)$$

The second term on the r.h.s. of (3.5.3-3) has the same expression as (3.4.3-5). The last term in (3.5.3-3) is the probability of the ancestor part of the history under consideration, available from the previous sampling time.

3.5.4 Suboptimal Versions

The problem of increasing memory and computation requirements is a consequence of the exponentially increasing number of measurement histories as indicated in (3.5.1-2).

This problem can be alleviated by the following procedure:

- All tracks that have identical histories for the last n scans before the current one are combined: separate estimates are kept *only* for the tracks that differ in the interval $[k - N, k - 1]$.

Computational Load

The expected number of histories to be stored in this “ N scan back” filter is (approximately)

$$n_h = \prod_{i=1}^n \{1 + E[m(i)]\} \quad (3.5.4-1)$$

The above expression is approximate since the random variables $m(i)$ are not independent: if at a given time m is large, this causes a larger uncertainty that will be reflected in a larger validation region, which in turn will lead to a larger number of validated measurements.

The PDAF algorithm of Section 3.4 corresponds to $N = 0$: no separate tracks (and, thus, estimates and covariances) are kept from the past — all the current measurements are combined immediately into a single state estimate.

The “no scan back” approach of the PDAF is equivalent to the first order Generalized Pseudo-Bayesian (GPB1) approach. The “one scan back” approach is, equivalent to the GPB2 algorithm.

3.5.5 Optimal Bayesian Approach — Summary

3.5.5 Optimal Bayesian Approach — Summary

The optimal procedure differs from the (suboptimal) PDAF in the following:

- The probabilities of *histories (sequences) of measurements*, rather than only the latest ones, being correct are computed.

This is equivalent to calculating the exact conditional pdf of the state, which in the linear-Gaussian case is a *Gaussian mixture* with an exponentially increasing number of terms.

The algorithm consists of the following:

- The history-conditioned estimates are obtained from standard filters
- The probability of each history of measurements is evaluated
- The history-conditioned estimates are
 - propagated to the next sampling time
 - combined (for output only).

The combination of the history-conditioned estimates is done with the mixture equations using as weightings the corresponding probabilities.

The covariance of the resulting combined estimate is given by the expression of the covariance of a mixture.

The number of measurement histories increases exponentially because the optimal algorithm carries histories from the initial time to the current time — both memory and computation requirements increase accordingly.

To limit these requirements, histories have to be limited within a *sliding window* that covers the current and the previous n sampling times.

The PDAF corresponds to $N = 0$, i.e., no past histories are carried along — the past is lumped into a single quasi-sufficient statistic.

3.6 TRACKING OF A TARGET IN THE PRESENCE OF WAKE

3.6.1 The Wake Problem

An algorithm conceptually similar to the one of Section 3.4 is presented for the tracking of a target in the presence of the **wake phenomenon**.

This phenomenon occurs

- In the reentry stage of ballistic vehicles: the target's radar return exhibits multiple peaks due to complex effects of a shield of ionized particles which surround the target and extend behind it forming a wake;
- For small fast ships: the wake behind it can cause spurious returns.

Assumption

- A single measurement is extracted from the return signal (after using the standard validation procedure discussed in Section 2.2): a position — range, azimuth (and elevation) — and is based on the estimated peaks or leading edge of the peaks.

This single measurement can be target originated, i.e., correct, or wake originated, i.e., false. The corresponding events are denoted as

$$\theta_1(k) = \{z(k) \text{ correct}\} \quad (3.6.1-1)$$

$$\theta_0(k) = \{z(k) \text{ false}\} \quad (3.6.1-2)$$

Denote the corresponding probabilities, to be presented later, as

$$\beta_1(k) = P\{\theta_1(k)|Z^k\} \quad (3.6.1-3)$$

$$\beta_0(k) = P\{\theta_0(k)|Z^k\} = 1 - \beta_1(k) \quad (3.6.1-4)$$

The estimation is done by evaluating at each time the probability of the latest measurement being correct and using this to update the state.

3.6.2 The Estimation Algorithm

3.6.2.2 The Estimation Algorithm

The estimation algorithm is, according to (3.4.2-8),

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \beta_1(k)W(k)\nu_1(k) \quad (3.6.2-1)$$

where $\beta_1(k)$ is defined in (3.6.1-3), $W(k)$ is the filter gain and⁵

$$\nu_1(k) = z(k) - \hat{z}(k|k-1) \quad (3.6.2-2)$$

Similarly, the covariance update equation (3.4.2-10) becomes, with the covariance of the state updated with the correct measurement, P^c , defined in (3.4.2-11),

$$P(k|k) = [1 - \beta_1(k)]P(k|k-1) + \beta_1(k)P^c(k|k) + \beta_1(k)[1 - \beta_1(k)]W(k)\nu_1(k)\nu_1(k)'W(k)' \quad (3.6.2-3)$$

The probability $\beta_1(k)$ is obtained, with p_i , $i = 0, 1$, the a priori probabilities of the measurement being false and correct, respectively, as

$$\beta_1(k) = P\{\theta_1(k)|Z^k\} = \frac{p[z(k)|\theta_1(k), Z^{k-1}]p_1}{p[z(k)|\theta_1(k), Z^{k-1}]p_1 + p[z(k)|\theta_0(k), Z^{k-1}]p_0} \quad (3.6.2-4)$$

The pdf of a correct measurement is, similarly to (3.4.3-3), taken as

$$p[z(k)|\theta_1(k), Z^{k-1}] = P_G^{-1}\mathcal{N}[\nu(k); 0, S(k)] \quad (3.6.2-5)$$

Assuming that the target's wake points away from the sensor, which measures range and two angles, the range measurement ρ can be assumed to have a linear pdf in an interval of length L_W , starting at the predicted range

$$p[\rho(k)|\theta_0(k), Z^{k-1}] = \frac{2}{L_W^2}(\rho - \hat{\rho}) \quad \hat{\rho} \leq \rho \leq \hat{\rho} + L_W \quad (3.6.2-6)$$

The angle measurements a_i can be assumed as uniformly distributed in an interval of length 2α around their predicted values

$$p[a_i(k)|\theta_0(k), Z^{k-1}] = \frac{1}{2\alpha} \quad \hat{a}_i - \alpha \leq a_i \leq \hat{a}_i + \alpha \quad (3.6.2-7)$$

For other target aspect angles similar models can be devised.

This algorithm amounts to a **selected neighbor modified filter**.

⁵Note that the combined innovation (3.4.2-9) is now $\nu(k) = \beta_1(k)\nu_1(k)$.

3.6.3 Tracking in the Presence of a Wake — Summary

3.6.3 Tracking in the Presence of a Wake — Summary

When tracking a ballistic missile in the presence of a wake the tracking filter has been assumed to receive one measurement that can be

- from the target or
- from the wake.

Basic assumption of the model:

- The location of a wake-originated measurement has a pdf that is triangular while the correct measurement's pdf is Gaussian — this provides the probabilistic discrimination capability.

The algorithm consists of the following:

- At each time the past is summarized by a quasi-sufficient statistic as in the PDAF
- When a measurement is obtained the probabilities of its two possible origins are evaluated based on the above model
- The state is updated with the weighting that the measurement is correct
- The covariance update has an extra term because of the measurement origin uncertainty
- The state and measurement prediction to the next sampling time are done in the standard manner.

This algorithm can be viewed as a special 2-model GPB1 estimator (see [BL93]).

3.7 TRACK FORMATION WITH BEARING AND FREQUENCY MEASUREMENTS IN THE PRESENCE OF FALSE DETECTIONS

3.7.1 The Problem

This section considers the problem of **target motion analysis (TMA)** — estimation of the trajectory parameters of a constant velocity target — with a **passive sonar**, which does *not provide full target position measurements*.

Assumptions

- The available **narrow band** passive sonar measurements are
 - the bearing (azimuth) of the target
 - the frequency of the target emission
 - * the emitted frequency (or frequencies) is constant during the observation period
 - * the received frequency is **Doppler** shifted
- There are false detections due to background noise, occurring with known probability
- The target detection probability is less than unity and known.

The use of frequency measurements, in addition to bearing measurements, yields complete observability of the target motion parameters from a *batch* of measurements *without requirement of a platform maneuver*.

The approach uses an ML formulation combined with the PDA technique and the estimation accuracy is quantified by a special **Cramer-Rao lower bound (CRLB) in the presence of false measurements**, which is shown to be met by the estimator.

A test of track acceptance — **target detection** — is also described.

3.7.2 Notations

3.7.2.1 Notations

The sampling times are

$$t_i \in \{t_1, \dots, t_n\} \quad (3.7.2-1)$$

The vector defining the motion of the target and the emitted frequency is

$$\mathbf{x} \triangleq [\xi(t_r), \eta(t_r), \dot{\xi}, \dot{\eta}, \gamma_0]' \quad (3.7.2-2)$$

where t_r is the reference time, ξ, η denote the Cartesian position components and γ_0 is the unknown frequency emitted by the target.

The relative position components of the target at t_i w.r.t. the ownship (the sensor platform) are $\xi_{OT}(t_i, x), \eta_{OT}(t_i, x)$.

The relative velocity vector of the target is $v_{OT}(t_i, x)$.

The true (noise free) bearing from the ownship to the target is

$$\beta(i, x) \triangleq \tan^{-1}[\xi_{OT}(t_i, x)/\eta_{OT}(t_i, x)] \quad (3.7.2-3)$$

The true (Doppler shifted but noise-free) frequency at the ownship is

$$\gamma(i, x) \triangleq \gamma_0 \left[1 - \frac{D(t_i, x)' v_{OT}(t_i, x)}{c_0} \right] \quad (3.7.2-4)$$

where c_0 is the sound velocity and the target's aspect vector is

$$D(t_i, x) \triangleq [\sin \beta(i, x), \cos \beta(i, x)]' \quad (3.7.2-5)$$

The volume of the surveillance region defined by the bearing interval $[B_1, B_2]$ and the frequency interval $[F_1, F_2]$ is

$$u \triangleq u_\beta u_\gamma \triangleq (B_2 - B_1)(F_2 - F_1) \quad (3.7.2-6)$$

The set of measurements at t_i is

$$Z(i) \triangleq \{z_j(i)\}_{j=1}^{m(i)} \triangleq \{[z_\beta(j), z_\gamma(j)], j \in \{1, \dots, m(i)\}\} \quad (3.7.2-7)$$

where z_β, z_γ are the measured bearing and frequency, respectively.

3.7.3 Statistical Assumptions

3.7.3.1 Statistical Assumptions

The five assumptions on which the likelihood function will be based are:

- A1. The measurements at different sampling times are, conditioned on x , independent

$$p[Z(i_1), Z(i_2)|x] = p[Z(i_1)|x]p[Z(i_2)|x] \quad \forall i_1 \neq i_2 \quad (3.7.3-1)$$

- A2. The target-originated bearing and frequency measurements are corrupted by mutually independent additive zero-mean white Gaussian noises

$$z_\beta(i) = \beta(i, x) + w_\beta(i) \quad (3.7.3-2)$$

$$z_\gamma(i) = \gamma(i, x) + w_\gamma(i) \quad (3.7.3-3)$$

with variances σ_β^2 and σ_γ^2 , respectively.

- A3. The false measurements (the clutter) are uniformly distributed in the surveillance region, i.e., with pdf u^{-1} .

- A4. The number of false measurements at each time has a probability mass function (pmf) $\mu_F(\cdot)$. This can be chosen as

- Poisson with known parameter λ (expected number of false measurements per unit of volume), or
- the diffuse prior model.

- A5. There is at most one target-originated measurement at each sampling time. This event occurs with a probability equal to P_D , the target detection probability, assumed known.

3.7.4 The Likelihood Function with False Measurements

Assumption A1 allows us to express the likelihood function of x as

$$p(Z^n|x) = \prod_{i=1}^n p[Z(i)|x] \quad (3.7.4-1)$$

At each sampling time t_i , given $m(i)$ detections, one has the following mutually exclusive and exhaustive events:

$$\theta_j(i) \triangleq \begin{cases} \{\text{measurement } z_j(i) \text{ is from the target}\} & j = 1, \dots, m(i) \\ \{\text{all measurements are false}\} & j = 0 \end{cases} \quad (3.7.4-2)$$

In view of assumptions A1–A5, the likelihood function of the measurements at the i -th sampling time is the following **uniform-Gaussian mixture**:

$$\begin{aligned} p[Z(i)|x] &= u^{-m(i)}(1 - P_D)\mu_F[m(i)] + \frac{u^{1-m(i)}P_D\mu_F[m(i)-1]}{m(i)} \\ &\cdot \sum_{j=1}^{m(i)} \frac{1}{2\pi\sigma_\beta\sigma_\gamma} \exp \left[-\frac{[z_{\beta_j}(i) - \beta(i, x)]^2}{2\sigma_\beta^2} - \frac{[z_{\gamma_j}(i) - \gamma(i, x)]^2}{2\sigma_\gamma^2} \right] \end{aligned} \quad (3.7.4-3)$$

Use of a Validation Region

Since “far out” measurements get small weight in the PDA, one can consider the restriction of $Z(i)$, defined in (3.7.2-7), to the validation gate

$$\mathcal{V}_g(i, x) \triangleq \left\{ (z_\beta, z_\gamma) : \left[\frac{z_\beta - \beta(i, x)}{\sigma_\beta} \right]^2 + \left[\frac{z_\gamma - \gamma(i, x)}{\sigma_\gamma} \right]^2 \leq g^2 \right\} \quad (3.7.4-4)$$

the volume of which is denoted as V_g .

The resulting set of **validated measurements** at time i is denoted as

$$Z^*(i) \triangleq \{z_j(i)\}_{j=1}^{m^*(i)} \quad (3.7.4-5)$$

where $m^*(i)$ is the number of validated measurements at time i .

The cumulative set (sequence) of validated measurements for the n sampling times is

$$Z^{*n} \triangleq \{Z^*(i)\}_{i=1}^n \quad (3.7.4-6)$$

3.7.4 The Likelihood Function with False Measurements

The Likelihood Ratio

The MLE of the target parameter x is to be obtained by numerical search. As the estimate changes in the course of this search, the number of validated measurements changes: this leads to a *change in the physical dimension* of the likelihood function, which is a joint pdf.

To avoid this, the MLE is computed by maximizing the *dimensionless likelihood ratio* obtained by dividing (3.7.4-3) with the pdf of the measurements assumed all false (uniformly distributed in V_g)

$$\mathcal{L}[Z^{*n}, x] \triangleq \frac{p[Z^{*n}|x]}{V_g^{-\sum_{i=1}^n m^*(i)} \prod_{i=1}^n \mu_F[m^*(i)]} \quad (3.7.4-7)$$

or the log-likelihood ratio

$$\ell[Z^{*n}, x] \triangleq \ln \mathcal{L}(Z^{*n}, x) \quad (3.7.4-8)$$

The log-likelihood ratio for time i is

$$\ell_i[Z^*(i), x] = \ln \left[\frac{p[Z^*(i)|x]}{V_g^{-m^*(i)} \mu_F[m^*(i)]} \right] \quad (3.7.4-9)$$

Its expression depends on the choice of $\mu_F(\cdot)$, the pmf of the number of false measurements. Using the Poisson pmf, one obtains (parametric PDA):

$$\ell_i[Z^*(i), x] = \ln \left[1 - P_D + \frac{P_D}{\lambda} \sum_{j=1}^{m^*(i)} \frac{1}{2\pi\sigma_\beta\sigma_\gamma} e^{-\frac{[z_{\beta_j}(i) - \beta(i,x)]^2}{2\sigma_\beta^2} - \frac{[z_{\gamma_j}(i) - \gamma(i,x)]^2}{2\sigma_\gamma^2}} \right] \quad (3.7.4-10)$$

The log-likelihood ratio for the entire data interval is

$$\ell[Z^{*n}, x] = \sum_{i=1}^n \ell_i[Z^*(i), x] \quad (3.7.4-11)$$

Thus

$$\hat{x} = \arg \max_x \ell[Z^{*n}, x] \quad (3.7.4-12)$$

3.7.5 The CRLB in the Presence of False Measurements

The **Cramer-Rao Lower Bound (CRLB)** for an unbiased estimate is the inverse of the **Fisher Information Matrix (FIM)**

$$J \triangleq E \left[[\nabla_x \ln p(Z^n | x)] [\nabla_x \ln p(Z^n | x)]' \right] \Big|_{x=x_{\text{true}}} \quad (3.7.5-1)$$

(see Subsection 2.7.2 of [BL93]).

In the case of a “clean” environment (i.e. without false measurements) with n independent measurements of the bearing β and frequency γ , one has

$$J = \sum_{i=1}^n J_i \quad (3.7.5-2)$$

where (see Section 3.7 of [BL93])

$$J_i \triangleq \frac{1}{\sigma_\beta^2} [\nabla_x \beta(i, x)] [\nabla_x \beta(i, x)]' + \frac{1}{\sigma_\gamma^2} [\nabla_x \gamma(i, x)] [\nabla_x \gamma(i, x)]' \quad (3.7.5-3)$$

In the presence of false measurements (clutter), the FIM cannot be described by an expression as simple as above.

The **FIM in the presence of false measurements** is obtained using the parametric PDA as [JB90]

$$J = q_2(P_D, \lambda V_g, g) \sum_{i=1}^n J_i \quad (3.7.5-4)$$

where the expression of J_i , the FIM for time i , is the same as in (3.7.5-3). The scalar multiplier q_2 , called the **information reduction factor** is discussed in Section 5.2.

The factor q_2 accounts for the reduction of the information available from the measurements due to the presence of the clutter. Its value, which depends on the three arguments indicated, is between 0 and 1.

3.7.6 Test for Track Acceptance

3.7.6.1 Test for Track Acceptance

If an estimate \hat{x} is obtained, one has to decide whether it is statistically acceptable. This is done with a test between the hypotheses

$$H_1 = \{\text{There is one track and } \hat{x} \text{ is the global maximum}\} \quad (3.7.6-1)$$

$$H_0 = \{\text{There is no track or } \hat{x} \text{ is not the global maximum}\} \quad (3.7.6-2)$$

According to the Neyman-Pearson lemma, the most powerful test of H_1 against H_0 would be to consider the (dimensionless) likelihood ratio $\mathcal{L}(Z^n, x)$ defined in (3.7.4-4) or, equivalently, the log-likelihood ratio $\ell(Z^n, x)$ defined in (3.7.4-5).

As discussed before, it is practically sufficient to consider the set of “validated” measurements according to (3.7.4-9), i.e., the log-likelihood ratios using $Z^*(i)$, rather than $Z(i)$, that is, $\ell_i[Z^*(i), x]$.

In view of this the test will be based on the m_i^* measurements in the validation gate $\mathcal{V}_g(i, x)$ at time i rather than the m_i measurements at time i in the whole surveillance region, $i = 1, \dots, n$.

The cumulative set (sequence) of validated measurements for the n sampling times is

$$Z^{*n} \triangleq \{Z^*(i)\}_{i=1}^n \quad (3.7.6-3)$$

The test statistic will be

$$T_{H_1/H_0}(Z^{*n}, \hat{x}) \triangleq \frac{\sum_{i=1}^n [\ell_i[Z^*(i), \hat{x}] - E\{\ell_i[Z^*(i), x_{\text{true}}] | H_1\}]}{\sqrt{\sum_{i=1}^n \text{var}\{\ell_i[Z^*(i), x_{\text{true}}] | H_1\}}} \quad (3.7.6-4)$$

Define the quantities

$$\mu_1 \triangleq E\{\ell_i[Z^*(i), x_{\text{true}}] | H_1\} \quad \sigma_1^2 \triangleq \text{var}\{\ell_i[Z^*(i), x_{\text{true}}] | H_1\} \quad (3.7.6-5)$$

The statistic (3.7.6-4) can be reduced to:

$$T_{H_1/H_0}(Z^{*n}) = \frac{\sum_{i=1}^n \ell_i[Z^*(i), \hat{x}] - n\mu_1}{\sqrt{n}\sigma_1} \quad (3.7.6-6)$$

3.7.6 Test for Track Acceptance

Furthermore, since the exact pdf of the likelihood ratio is not available, the following simplification is made:

- the Central Limit Theorem is assumed to hold and, thus, the statistic (3.7.6-6) has approximately the standard Gaussian distribution (this was verified experimentally via simulations).

Hence, the test with **miss probability** α is

$$\boxed{\text{if } T_{H_1/H_0}(Z^{*n}, \hat{x}) \geq c_\alpha \text{ then } H_1 \text{ is accepted}} \quad (3.7.6-7)$$

where c_α is based on the one-sided tail of the standard Gaussian, i.e.,

$$c_\alpha = \mathcal{G}(1 - \alpha) \quad (3.7.6-8)$$

where \mathcal{G} is the inverse Gaussian cumulative distribution function; $\mathcal{G}(1 - \alpha)$ has probability mass $1 - \alpha$ to its left and α to its right (see Subsection 1.5.4 of [BL93]).

The derivation of the first two moments (3.7.6-5) and (3.7.6-6) of $\ell_i[Z^*(i), x]$ under the hypothesis H_1 is presented in [JB90]. In practice the evaluation of these moments has to be done at \hat{x} rather than x_{true} , which is not available. In this case one assumes

$$E\{\ell_i[Z^*(i), \hat{x}]|H_1\} \approx E\{\ell_i[Z^*(i), x_{\text{true}}]|H_1\} \quad (3.7.6-9)$$

$$\text{var}\{\ell_i[Z^*(i), \hat{x}]|H_1\} \approx \text{var}\{\ell_i[Z^*(i), x_{\text{true}}]|H_1\} \quad (3.7.6-10)$$

since under H_1 , one has $\hat{x} \approx x_{\text{true}}$.

3.7.7 Numerical Results

3.7.7.1 Numerical Results

The Signal Processing Parameters

The standard scheme of signal processing in a passive narrow-band sonar is:

- The received signal is filtered in several octaves, then sampled in each octave at the Nyquist rate (after band shifting)
- An FFT and a beamforming in each octave are performed
- The target angle is estimated.

As a consequence, the measurement data consist of

- Frequency measurements, quantized according to the cell size C_γ , imposed by the sampling period of the signal processing, and
- Azimuth measurements, which can be considered as a realization of a continuous random variable in each preformed beam.

The relationship between the probability of false alarm and the expected number of false measurements λ per unit volume is

$$P_{FA} = \lambda C_\gamma C_\beta \quad (3.7.7-1)$$

where C_β is the beam width, considered here as constant.

With the octave [250Hz, 500Hz] and a 2048-point FFT, the frequency cell size is

$$C_\gamma = 250/2048 = 0.122\text{Hz} \quad (3.7.7-2)$$

The frequency data have been quantized according to this cell size.

Assuming that 32 equal beams have been formed, the beam width is

$$C_\beta = 180^\circ/32 = 5.625^\circ \quad (3.7.7-3)$$

The surveillance region was the rectangle [295Hz, 303Hz] \times [-35°, 15°].

The probability of false alarm per cell was $P_{FA} = 10^{-2}$, yielding an expected number of 7 false measurements in the surveillance region at each time.

The target detection probability was $P_D = 0.8$.

The standard deviations of the azimuth and frequency measurements were 1° and 0.3Hz, respectively. The average number of false measurements in the gate $g = 5$ was 0.4.

3.7.7 Numerical Results

The Scenario

The number of sampling times was 30 and the sampling period was 60s.

A scenario for target parameter $x = [1000m, 30000m, -10m/s, 0m/s, 300Hz]'$ is shown in Figure 3.7.7-1 together with the initial and final position estimates in 100 runs and their 95% confidence ellipses based on the CRLB.

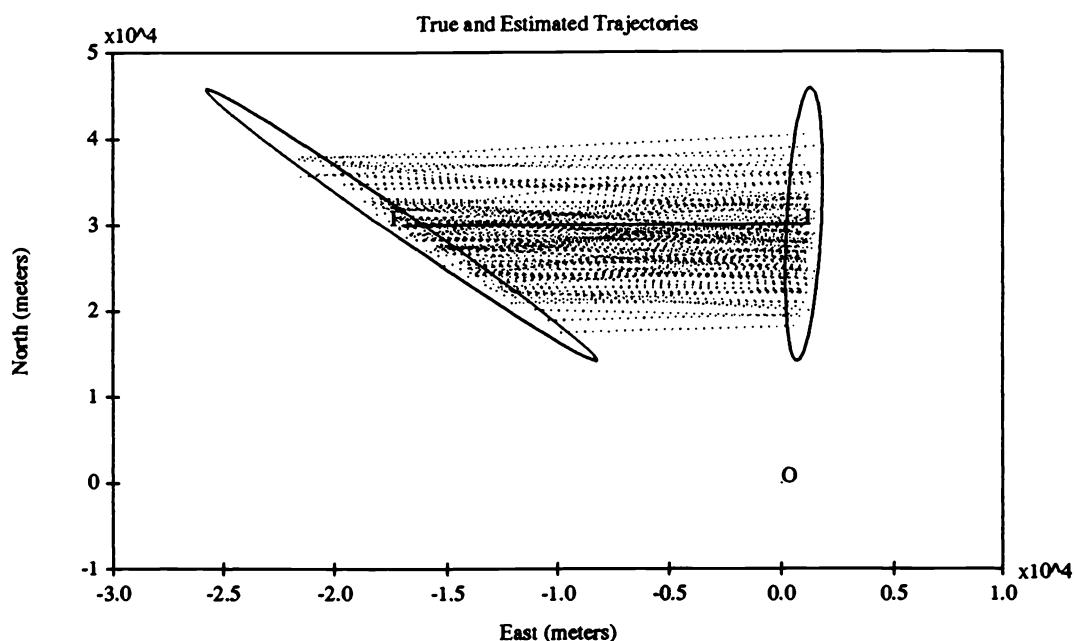


Figure 3.7.7-1: Scenario and estimates of initial target position from 100 runs (o — platform location; I — target initial position); F — target final position).

3.7.7 Numerical Results

Figure 3.7.7-2 displays the sets of measurements from one of the runs.

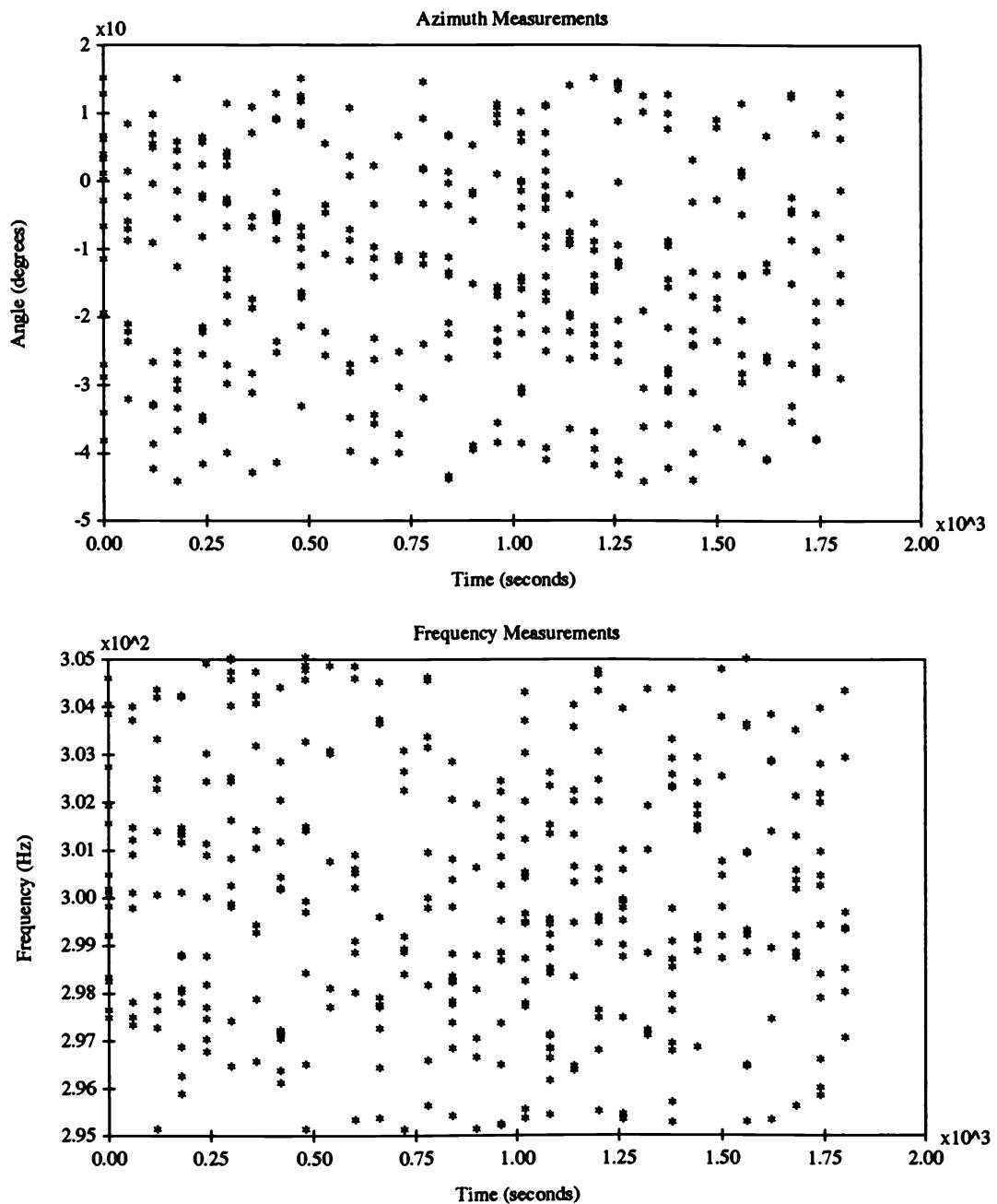


Figure 3.7.7-2: Measurements in one run.

3.7.7 Numerical Results

Statistical Properties of the Estimates

Table 3.7.7-1 shows the mean \bar{x} of the estimates, the empirical and CRLB-based standard deviations, $\hat{\sigma}$ and σ_{CRLB} , respectively. The initial state x_{init} used to initialize the search for the MLE, via the Newton-Raphson algorithm is also shown.

Units	x_{true}	x_{init}	\bar{x}	σ_{CRLB}	$\hat{\sigma}$
m	1000	2300	993	344	395
m	30000	50000	29908	6757	7200
m/s	-10	-15	-9.93	2.19	2.27
m/s	0	-3	-0.07	1.58	1.64
Hz	300	298	299.973	0.346	0.348

Table 3.7.7-1: Results of 100 Monte Carlo runs.

The normalized estimation error squared (NEES) was 4.9 — close to the theoretical mean of 5 and within the 95% confidence region [4.4, 5.6].

This indicates that the estimator is **efficient** — the CRLB is met.

With the test set for 5% rejection probability, the number of acceptances of H_1 (“target present”) was 97 and the number of rejections was 3.

Scenario Without Target

In this case, whatever the initial vector, the final estimate obtained by the algorithm was always rejected by the test.

The test proposed here is, hence, very powerful, which is in agreement with the theory.

The SNR for the Scenario

The parameters of the scenario considered ($P_{FA} = 10^{-2}$, $P_D = 0.8$) correspond, based on the Rayleigh detection model (4.6.2-5), to a **cell signal-to-noise ratio** $\text{SNR}_C = 12 \text{ dB}$ (approximately). The SNR per 1 Hz bandwidth is, with the frequency cell of $\frac{1}{8} \text{ Hz}$, $\text{SNR}_1 = \text{SNR}_C - 10 \log 8 \approx 12 - 9 = 3 \text{ dB}$.

Note

These results were obtained with the interactive software **BEARDAT™**. The above problem parameters indicate that the algorithm behaves very well in low SNR situations.

3.7.8 Low SNR Track Formation with a Passive Sonar — Summary

3.7.8 Low SNR Track Formation with a Passive Sonar — Summary

The TMA problem with a batch of

- bearing and
- frequency measurements

was presented and solved

- in the presence of heavy clutter or false alarms,

which is a realistic situation for a narrow-band passive sonar operating in a low SNR environment.

The target's velocity and frequency are assumed to be constant during the observation (batch) period.

The complete observability of the target parameter *without platform maneuver* is obtained from the use of the Doppler shifted frequency measurements.

The ML estimate of the target parameter vector — initial position, its velocity and the frequency it emits — are obtained via a numerical search.

The proposed ML/PDA estimator is statistically efficient.

A test of track acceptance — *target detection* — was also described.

The same approach can be carried out for the conventional bearings-only TMA, in which case the platform has to maneuver (see Section 3.7 of [BL93]).

The proposed estimator can be used for the initialization of a recursive filter.

Note on the SNR

The Rayleigh fluctuating amplitude model (4.6.2-5) requires, for the same P_D , P_{FA} operating point, a higher SNR than the Receiver Operating Characteristic in Figure 5.4.1-2 (which is based on a Rician model). The Rayleigh model is believed to be more appropriate for shallow water passive sonar, while the Rician model for the deep water situation. In practice, the Rayleigh model is somewhat conservative.

3.8 NOTES AND PROBLEMS

3.8.1 Bibliographical Notes

The NNSF, as described in Section 3.2, was the state of art in tracking until the early 1970s and is still in use in many less sophisticated systems. One application of this approach was reported in [FP78].

The track splitting algorithm from Section 3.3, which is another engineering common-sense approach, was described in [SB75a].

The first modification of the Kalman filter to account for the fact that the nearest neighbor might not be the correct measurement was proposed in [SS71], which eventually became [SS73]. It involved the computation of the prior probability of the nearest neighbor measurement being correct and a suitable modification of the filter gain and state estimate covariance. The evaluation of the posterior probability of the nearest neighbor was first proposed in [JB72] and refined in [Li93]. Recent advances in performance prediction for standard and modified nearest neighbor filters include [Rog91, LB94d].

The PDAF algorithm (Section 3.4) that uses all the neighbors or validated measurements with their posterior probabilities was first suggested in [BJ72, BT73] and finalized in [BT75]. The parametric and non-parametric versions of it are from [BB82]. One of the practical implementations of the PDAF, with an extension that includes “observable” and “unobservable” target situations, has been reported in [CDA86]. Other applications of the PDAF can be found in [Bar90, Bar92].

A smoothing version of the PDAF has been presented in [MPG86].

The optimal Bayesian filter for a single target, presented in Section 3.5, is from [SSH74].

The application of the probabilistic association (“editing”) of the “tallest neighbor” to tracking a target with a wake (Section 3.6) is based on [AWG77], which follows the methodology of [JB72].

Section 3.7 is based on [JB90], which contains more details about the ML/PDA algorithm. This algorithm is available in the interactive software BEARDATTM, which also includes the C source code.

3.8.2 Problems

3-1 Measurement validation example. Given the scalar system

$$x(k+1) = x(k) + v(k)$$

with the correct measurement

$$z(k) = x(k) + w(k)$$

where $v \sim \mathcal{N}\left(0, \frac{1}{3}\right)$ and $w \sim \mathcal{N}\left(0, \frac{1}{3}\right)$. Let $\hat{x}(0|0) = 1$ and $P(0|0) = \frac{1}{3}$.

- (i) Find the variance of the correct innovation at $k = 1$.
- (ii) Specify the 99% validation gate at $k = 1$.
- (iii) Let the measurements at $k = 1$ be located at $-1, 2, 4$. Indicate the validated measurements.
- (iv) Compute $\hat{x}(1|1)$ and $P(1|1)$, with $P_D = 0.8$ using the nonparametric PDAF.
- (v) Redo items (ii)–(iv) for a “ 4σ ” gate ($\gamma = 16$; $P_G = 1$ practically in this case).

3.8.2 Problems

- (vi) Using again a 99% gate redo (iii) and (iv) if the measurements were $-2.5, -2, -1.5, -1, -0.5, 0, 1, 1.5, 2, 2.5, 3, 3.5$.

3-2 **Single scan misassociation probability.** A “good (true) measurement” (from a target) has a “normalized distance” from the origin ξ with pdf

$$p(\xi) = \frac{1}{2}e^{-\frac{\xi}{2}} \quad \xi > 0$$

(this is a chi-square density with 2 degrees of freedom).

“False measurements” arrive according to a Poisson process with rate λ per unit of distance from the origin. A gate is set up as the interval $[0, \gamma]$: a measurement is an association candidate only if its distance is less than γ .

- (i) Find the probability P_G that the “good measurement” falls in the gate.
- (ii) Find the probability that a “misassociation” occurs. This event, denoted as M , occurs if
 - a) ξ is outside the gate and there is a false measurement in the gate, or
 - b) ξ is inside and there is a false measurement closer to the origin than ξ .

Assume in the sequel that the true measurement is detected with probability

$$P\{D\} = 1 - P\{\bar{D}\} = P_D < 1.$$

- (iii) Find the probability of a misassociation (denoted now as M').
- (iv) Find the probability of a correct association. This event, denoted as A , occurs if the target is detected, falls in the gate and there is no false measurement with a smaller distance.
- (v) Find the probability of “no association.” This event, denoted as \bar{A} , occurs if the gate is empty.
- (vi) Check that the probabilities of the events M' , A and \bar{A} sum up to unity.

Chapter 4

MANEUVERING TARGET IN CLUTTER

4.1 INTRODUCTION

4.1.1 Outline

Tracking a maneuvering target in the presence of clutter or false measurements involves two problems:

- the unknown inputs into the target state equation or the change of target state model that reflect the maneuver,
- the presence of false measurements.

When a target maneuvers, the measurement of a kinematic variable from it can deviate significantly from its expected location. While such a deviation has been used for maneuver detection, it would be discounted by the PDA procedure: the measurements lying farther away from their expected location, i.e., with large innovations, were considered less likely to be correct.

Due to this *conflict between data association and maneuver detection* when the measurements consist only of kinematic variables, the availability of **feature variables** in the measurement vector can be helpful. Such variables can always increase the power of the data association, and they are, in general, not going to be affected by target maneuvers in the manner position measurements are.

4.1.1 Outline

A technique that uses as feature the *rank of a measurement* (number of detections from a pulse sequence) in the PDA is presented in Section 4.2 and illustrated with several realistic maneuvering target examples.

Section 4.3 presents the combination of the (static) MM approach from Subsection 11.6.2 of [BL93] with the PDAF. This approach, called MMPDAF, is illustrated with some real data examples.

The most effective approach appears to be the combination of the (dynamic) IMM algorithm, which accounts for the “mode jumping,” with the PDAF, resulting in the IMMPDAF.

A technique for formation (initiation) of tracks, based on the IMMPDAF is presented in Section 4.4.

Section 4.5 presents the multisensor IMMPDAF, which integrates the multisensor sequential updating technique from Subsection 2.2.2, the IMM approach from Subsection 1.5.4 (for more details see Subsection 11.6.6 of [BL93]) and the PDAF from Section 3.4. The resulting algorithm is illustrated on a problem of tracking a highly maneuvering target in clutter with a radar and an infrared sensor.

The IMMPDAFAI algorithm, which combines the IMM with the PDAF and also utilizes a feature (amplitude) measurement — **amplitude information (AI)** — is discussed in Section 4.6. This algorithm is shown to be suitable for

- track formation, as well as
- track maintenance in the presence of maneuvers

in a realistic situation with target detection probability less than unity and significant false alarm probability.

4.1.2 Summary of Objectives

4.1.2 Summary of Objectives

Show the utilization of a feature measurement in tracking a maneuvering target in clutter.

Present the MMPDAF and illustrate it with a real-data example.

Show the use of the IMMPDAF for “intelligent” track formation (track initiation).

Present the multisensor IMMPDAF and show its application to tracking a highly maneuvering target in clutter with a radar and an infrared sensor (a sea skimmer).

Describe the IMMPDAFAI for

- track formation
- track maintenance

and illustrate its application to an active sonar problem.

4.2 USE OF THE PDAF WITH A FEATURE MEASUREMENT FOR MANEUVERING TARGET TRACKING

4.2.1 The Feature Measurement and the Augmented PDA

In the situation to be considered next, a radar operates as follows:

- In a given scan there are n pulses transmitted and, therefore, there can be at most n detections from a target
- For every transmitted pulse a “hit” is declared in [a] cell if the return in it exceeds that of J reference cells
- The centroid of a **hit pattern** becomes a measurement (**plot**) if the number of detections in it, r , called the **rank of the measurement**, exceeds a threshold τ
- A **constant false alarm rate (CFAR)** signal detector, based on comparison to n_r reference cells, has a **single hit false alarm probability**, assuming i.i.d. clutter samples,

$$p_C = \frac{1}{n_r + 1} \quad (4.2.1-1)$$

which does not depend on the clutter distribution

- With the same signal detector, the **target single-hit detection probability**, p_T , depends on the signal and clutter distributions — it is determined by the **signal to clutter ratio (SCR)**.

The augmented measurement vector $z_a(k)$ at time (scan) k , consisting of the kinematic variable z and the feature variable r , is given by

$$z_a(k) = \begin{bmatrix} z(k) \\ r(k) \end{bmatrix} \quad (4.2.1-2)$$

The pmf of the rank of a clutter-originated hit pattern is

$$P\{r = \rho | \theta_C\} = \binom{N}{\rho} p_C^\rho (1 - p_C)^{N-\rho} \triangleq \mu_C(\rho) \quad (4.2.1-3)$$

where θ_C denotes clutter origin.

Similarly, denoting by θ_T target origin, one has

$$P\{r = \rho | \theta_T\} = \binom{N}{\rho} p_T^\rho (1 - p_T)^{N-\rho} \triangleq \mu_T(\rho) \quad (4.2.1-4)$$

4.2.1 The Feature Measurement and the Augmented PDA

Since a hit pattern is accepted as a measurement if its rank is at least τ , one has the **probability of target detection**

$$P_D = \sum_{\rho=\tau}^N \binom{N}{\rho} p_T^\rho (1-p_T)^{N-\rho} \quad (4.2.1-5)$$

and the **probability of a false alarm (false measurement)**

$$P_F = \sum_{\rho=\tau}^N \binom{N}{\rho} p_C^\rho (1-p_C)^{N-\rho} \quad (4.2.1-6)$$

The Augmented PDA

As in Subsection 3.4.7, denote the **target to clutter feature likelihood ratio**

$$\boxed{L_{T/C}(r) \triangleq \frac{\mu_T(r)}{\mu_C(r)}} \quad (4.2.1-7)$$

where the definitions (4.2.1-3) and (4.2.1-4) were used.

The association probabilities can be shown to be the same as in the standard PDA (3.4.3-11), except for the exponential (3.4.3-12) being replaced by

$$\boxed{e_i \triangleq e^{-\frac{1}{2}\nu_i(k)'S(k)^{-1}\nu_i(k)} L_{T/C}[r_i(k)]} \quad (4.2.1-8)$$

4.2.2 Examples

4.2.2 Examples

In the examples to be considered the performance measure was taken as the **mean track life** — the number of scans the algorithm can track the target from the moment clutter is introduced.

A track was deleted if

- (i) The association “score” that quantifies the spread of the association probabilities from the PDA

$$\hat{\beta}(k) \triangleq \sum_{i=1}^{m(k)} \beta_i(k)^2 \quad (4.2.2-1)$$

was below the threshold $\beta_r = 0.7$, or

- (ii) The number of consecutive missed detections reached $n_{MD} = 2$. This reflects the following “composite track maintenance logic”: at least τ out of n single-pulse detections are required to yield a centroid measurement and at least 1 out of n_{MD} measurements are required.

The state model used is the one from Subsection 6.6.3 of [BL93] with autocorrelated acceleration with the decorrelation time constant $1/\alpha = 60s$, process noise variance $\sigma_m^2 = 200(m/s^2)^2$.

The measurements were: range with standard deviation $\sigma_r = 150m$, azimuth with $\sigma_\theta = 3mr$, and radial velocity with $\sigma_v = 30m/s$.

The target single-hit probability of detection p_T was based on the Swerling 2 scintillation model with various values of the **signal to clutter ratio (SCR)**. The target measurement detection probability from $N = 15$ pulses follows from (4.2.1-5) with the threshold $\tau = 4$.

4.2.2 Examples

The first maneuvering target trajectory considered, shown in Figure 4.2.2-1, has a $0.5g$ turn and is observed with sampling period $T = 10\text{s}$. Clutter is present during the segment CE, which lasts 40 scans.

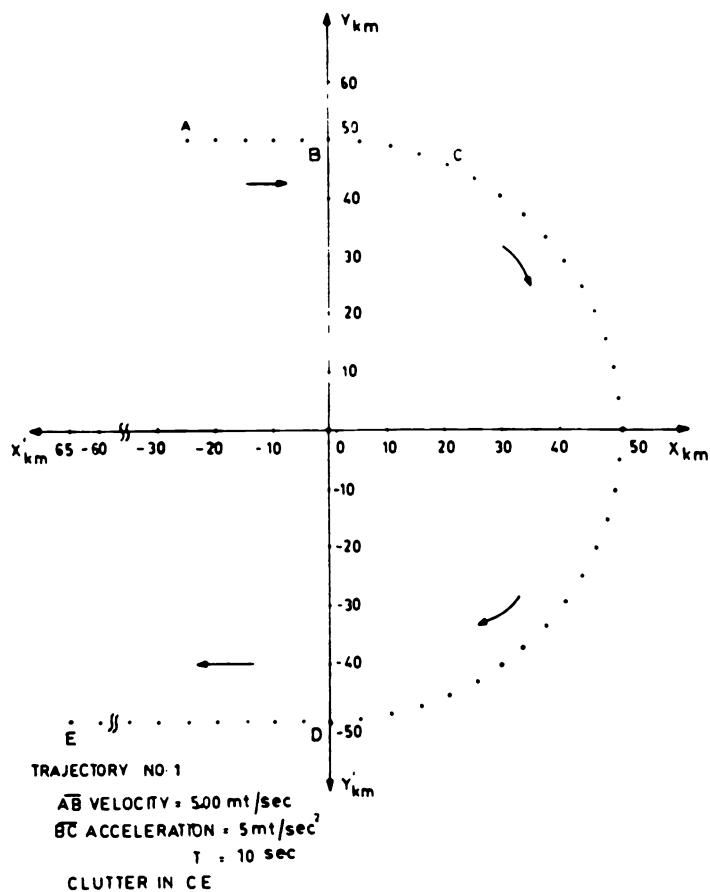


Figure 4.2.2-1: Target with $0.5g$ turn (Case 1).

4.2.2 Examples

Figure 4.2.2-2 shows the variation of the track life with the SCR, which determines P_D , for several values of \bar{m}_k , the expected number of clutter points in the 99% validation region.

The rank feature, when used in the PDA, yields a significant improvement in the performance. This is due to the fact that, in the presence of maneuvers, the measurements of kinematic variables do not carry enough information for reliable association.

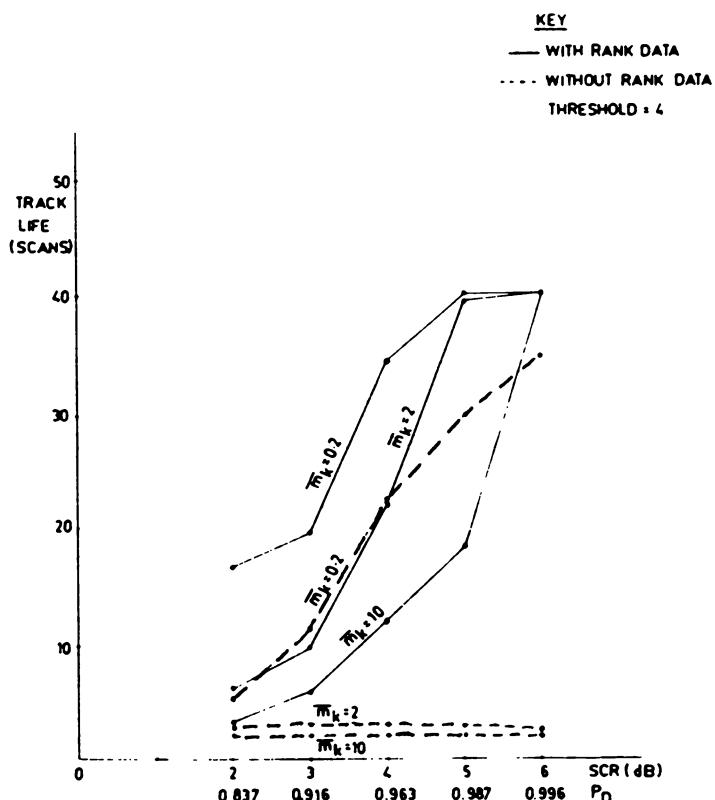


Figure 4.2.2-2: Average target track life variation with the clutter density (Case 1).

4.2.2 Examples

Figure 4.2.2-3 shows the variation of the average target track life as a function of the threshold τ for the PDAF with and without feature measurements.

In the PDAF *without feature measurements* the performance is *highest* for $\tau = 6$ (i.e., $Th = 6$ in the Figure). Above this level the target is not detected often enough, while below this level there are too many false measurements.

The performance of the PDAF *with feature measurements* is, however, *improving with lowering of the threshold τ* : the additional false measurements, which have a low number of hits, get low probabilities and, therefore do not affect adversely the performance. Lowering the threshold τ is equivalent to increasing the validation gate in the feature space.

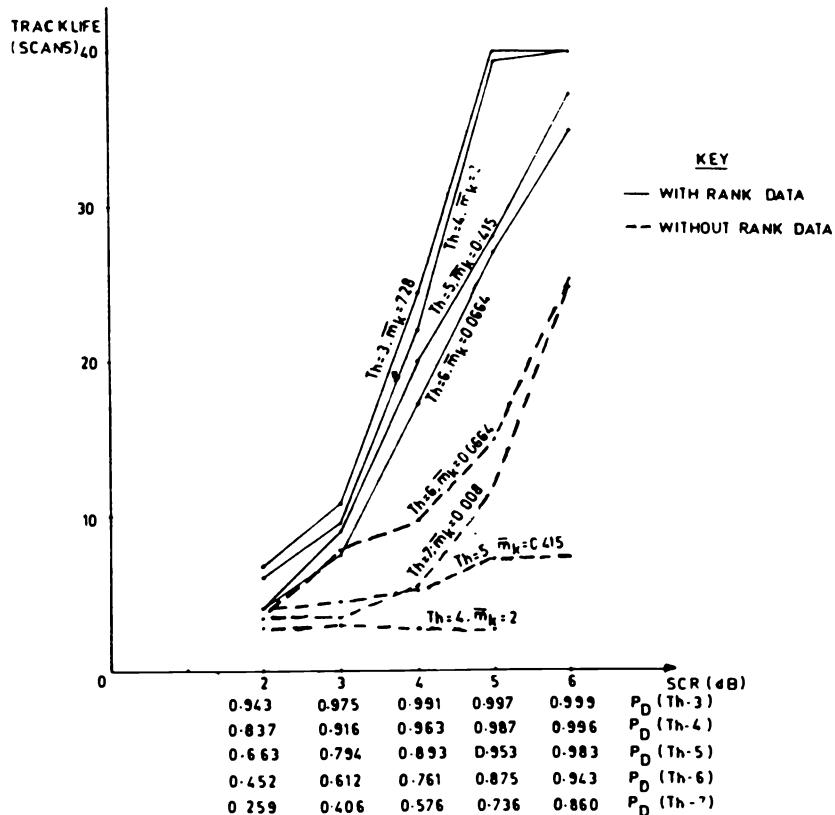


Figure 4.2.2-3: Average target track life variation with the feature threshold.

4.2.2 Examples

In Case 2 a higher ($2g$) acceleration trajectory was considered, as shown in Figure 4.2.2-4. The results, in terms of target track life as a function of the clutter density, are similar to the previous case and are illustrated in Figure 4.2.2-5. The effect of the feature threshold choice is shown in Figure 4.2.2-6.

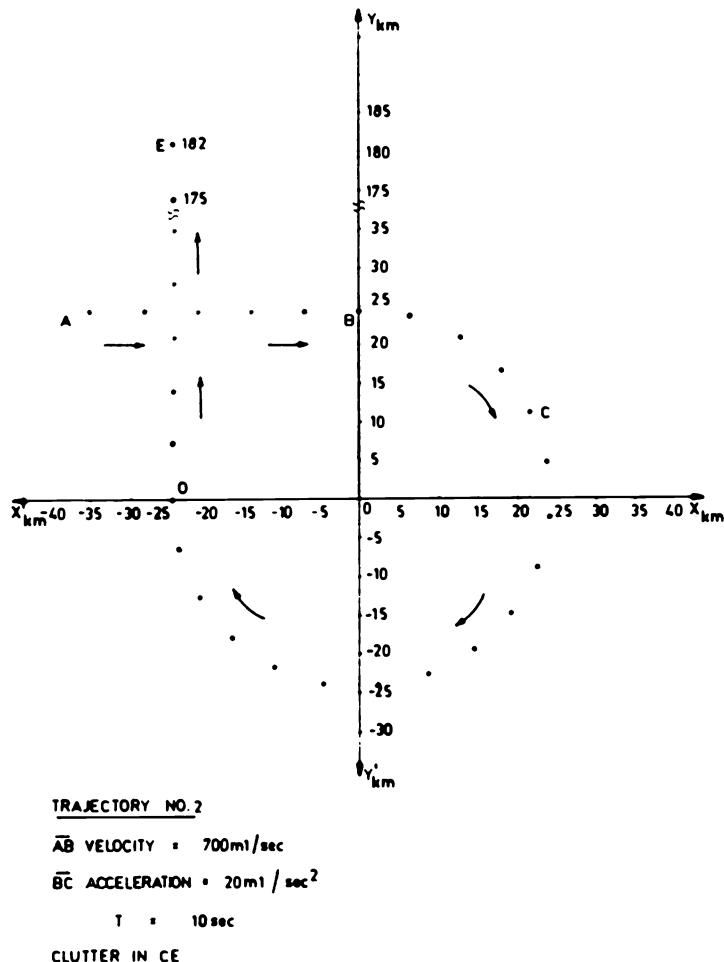


Figure 4.2.2-4: Target with $2g$ turn (Case 2).

4.2.2 Examples

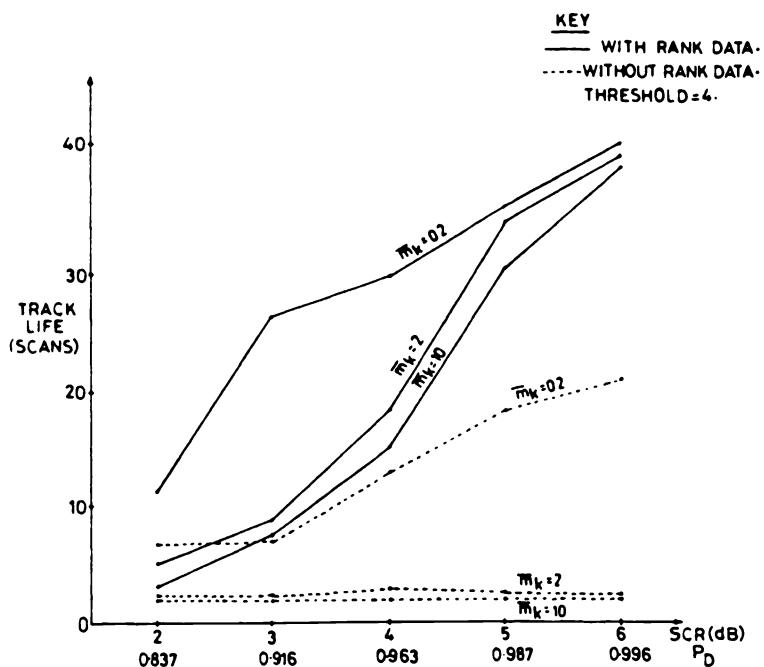


Figure 4.2.2-5: Average target track life variation with the clutter density (Case 2).

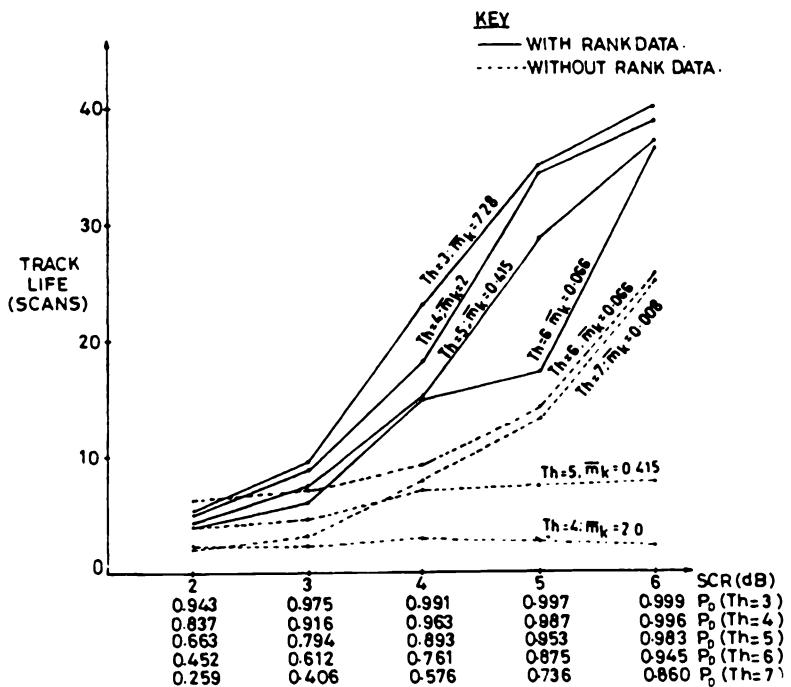


Figure 4.2.2-6: Average target track life variation with the feature threshold (Case 2).

4.2.3 PDAF with Feature Measurements — Summary

4.2.3 PDAF with Feature Measurements — Summary

Feature measurements can (and should, whenever possible) be included in the measurement vector to increase the discrimination power of the PDA against the false measurements.

This is particularly important in the case of maneuvering targets, where the measurements of kinematic variables can be too far from their predicted location and, therefore, will receive low association probabilities.

The PDA can be easily augmented to include probabilistic models for feature variables.

The key to the effective use of a feature is a suitable probabilistic model that distinguishes between

- the feature of a target
- the feature of a false measurement.

With a feature that is related to the strength of the target signal, one can lower the detection threshold without degrading the performance. This is due to the fact that

the additional (“weak”) measurements that enter into PDAF are discounted — receive low weights from the PDA.

The use of amplitude information as feature in a multiple model PDAF is discussed in Subsection 4.6.3.

4.3 THE MULTIPLE MODEL PDAF

4.3.1 Introduction

The **Multiple Model Probabilistic Data Association Filter (MMPDAF)** is a combination of

- the (nonswitching or static) MM approach of Section 11.6 of [BL93], and
- the PDAF of Section 3.4.

The result is an **adaptive estimator** that can adjust itself to the “true model” of the target while in a cluttered environment.

Assumptions

- The system obeys one out of r models
- The models do not switch in time
- The models differ only in the process and measurement noise levels.

The last two assumptions will be relaxed later.

The dynamic equation of the system is

$$x(k+1) = Fx(k) + v(k) \quad (4.3.1-1)$$

where the process noise is zero-mean white with covariance Q_j , $j = 1, \dots, r$.

The target-originated measurements are

$$z(k) = Hx(k) + w(k) \quad (4.3.1-2)$$

with the measurement noise zero-mean white, with covariance R_j , $j = 1, \dots, r$.

The remaining assumptions are the same as in the case of the standard PDAF.

The prior probability of model j is

$$P\{M_j\} = \mu_j(0) \quad (4.3.1-3)$$

4.3.2 The MMPDAF Algorithm

4.3.2 The MMPDAF Algorithm

The set of validated measurements at time k is denoted by

$$Z(k) = \{z_i(k)\}_{i=1}^{m(k)} \quad (4.3.2-1)$$

A **common validation region** for all the mode-matched filters is necessary so the mode likelihood functions are the pdf of the *same set* of measurements. The volume of this common region is denoted as V .

In the absence of model uncertainty — only additive noise and measurement origin uncertainties — the estimate of the state at time k is, with $\theta_i(k)$ denoting the event that $z_i(k)$ is the correct measurement at k ,

$$\hat{x}(k|k) = \sum_{i=0}^{m(k)} E[x(k)|\theta_i(k), Z^k] P\{\theta_i(k)|Z^k\} \quad (4.3.2-2)$$

With the additional model uncertainty, the estimate of the state is obtained by using the total probability theorem w.r.t. the measurement origin events *and* the model events M_j , $j = 1, \dots, r$, as follows

$$\begin{aligned} \hat{x}(k|k) &= \sum_{j=1}^r \sum_{i=0}^{m(k)} E[x(k)|\theta_i(k), M_j, Z^k] P\{\theta_i(k), M_j | Z^k\} \\ &= \sum_{j=1}^r \left\{ \sum_{i=0}^{m(k)} E[x(k)|\theta_i(k), M_j, Z^k] P\{\theta_i(k)|M_j, Z^k\} \right\} P\{M_j | Z^k\} \end{aligned} \quad (4.3.2-3)$$

or

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \mu_j(k)$$

(4.3.2-4)

where $\hat{x}^j(k|k)$ is the output of a *PDAF based on model j* and

$$\mu_j(k) \triangleq P\{M_j | Z^k\} \quad j = 1, \dots, r \quad (4.3.2-5)$$

is the conditional probability of mode j . Eq. (4.3.2-4) is the same as (11.6.2-5) of [BL93] and the corresponding covariance is given by (11.6.2-6) of [BL93].

4.3.2 The MMPDAF Algorithm

The Mode Probabilities

Using Bayes' formula (note the **common conditioning** Z^k for all j)

$$P\{M_j|Z^k\} = \frac{p[Z(k)|M_j, Z^{k-1}]P\{M_j|Z^{k-1}\}}{\sum_{l=1}^r p[Z(k)|M_l, Z^{k-1}]P\{M_l|Z^{k-1}\}} \quad (4.3.2-6)$$

yields the recursion for the conditional model probabilities as

$$\boxed{\mu_j(k) = \frac{\Lambda_j(k)\mu_j(k-1)}{\sum_{l=1}^r \Lambda_l(k)\mu_l(k-1)}} \quad (4.3.2-7)$$

where

$$\Lambda_j(k) \triangleq p[Z(k)|M_j, Z^{k-1}] = p[z_1(k), \dots, z_{m(k)}(k)|M_j, Z^{k-1}] \quad (4.3.2-8)$$

is the **likelihood function** of mode j at time k .

From the discussion of Subsection 3.4.3, the above likelihood function is the **uniform-Gaussian mixture**

$$\boxed{\Lambda_j(k) = V^{-m(k)}\gamma_0^j[m(k)] + V^{-m(k)+1} \sum_{i=1}^{m(k)} P_G^{-1} \mathcal{N}[\nu_i^j(k); 0, S^j(k)]\gamma_i^j[m(k)]} \quad (4.3.2-9)$$

where the prior probability of measurement i being correct is

$$\gamma_i^j[m(k)] = \begin{cases} \frac{P_D P_G}{m(k)} & i = 1, \dots, m(k) \\ 1 - P_D P_G & i = 0 \end{cases} \quad (4.3.2-10)$$

and $\nu_i^j(k)$ is the innovation corresponding to $z_i(k)$ in the PDAF based on model j , whose innovation covariance is $S^j(k)$.

The structure of the estimator is the same as that of the standard nonswitching MM estimator described in Subsection 11.6.2 of [BL93], except for the replacement of the KF modules by PDAF modules.

The resulting “bank of PDAFs” can be used for adaptive tracking of a maneuvering target. The same approach can be used in conjunction with a switching model assumption, e.g., with the IMM algorithm – this is done in the next Section.

4.3.3 Application of the MMPDAF to Track a Real Target

4.3.3 Application of the MMPDAF to Track a Real Target

In this example the sensor was a direction finder, measuring an angle — the bearing from the sensor to the target. The sampling period was $T = 0.108s$.

At every sampling time 10 detections were available from the signal processor, but only one of them originated from the target.

The maneuvering target, a ship in this case, was modelled using the autocorrelated noise approach of Subsection 8.2.3 of [BL93]. The state equation was as in (8.2.3-5) of [BL93] with the state consisting of angular position, velocity and acceleration.

The Unknown Parameters

- The covariance matrix of the process noise was given by expressions (8.2.3-10) of [BL93] but with unknown driving noise standard deviation σ_m ,
- The variance of the measurement noise, R , was also unknown.

Results with a Single-Model PDAF

A standard PDAF with “guessed values” for the parameters $\sigma_m = 0.3$ and $R = 0.1$ was run first with the results shown in Figures 4.3.3-1 and 4.3.3-2. The true trajectory of the target was known (with a somewhat coarse quantization) and is shown in Figure 4.3.3-1 for comparison with the estimated position.

The estimates are seen to be very noisy, especially for the velocity.

4.3.3 Application of the MMPDAF to Track a Real Target

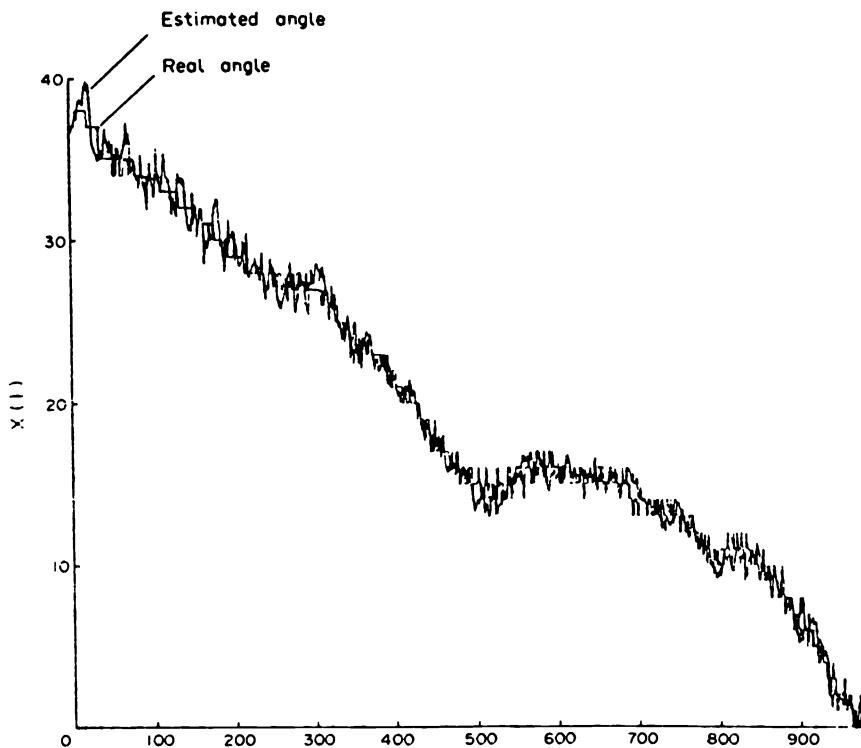


Figure 4.3.3-1: True and estimated position with a single PDAF.

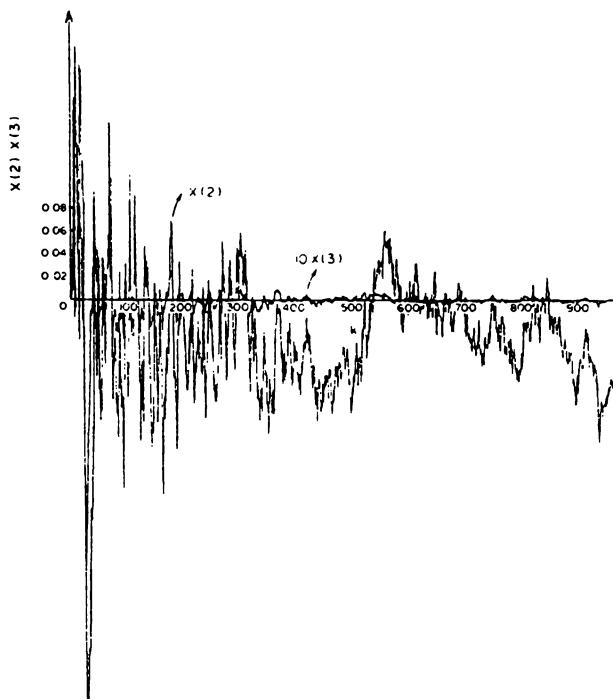


Figure 4.3.3-2: Estimated velocity and acceleration with a single PDAF.

4.3.3 Application of the MMPDAF to Track a Real Target

The Models for the MMPDAF

A grid of values was used for the bank of PDAFs. The grid points, r_1 values for σ_m and r_2 values for R , were chosen equally spaced

$$\sigma_m^s = \sigma_m^0 + \frac{s}{r_1}(\sigma_m^{r_1} - \sigma_m^0) \quad s = 1, \dots, r_1 \quad (4.3.3-1)$$

$$R^l = R^0 + \frac{l}{r_2}(R^{r_2} - R^0) \quad l = 1, \dots, r_2 \quad (4.3.3-2)$$

where σ_m^0 , $\sigma_m^{r_1}$ define the assumed range of σ_m and R^0 , R^{r_2} define the range of R . This amounts to $r = r_1 r_2$ models.

The real measurement data were run through a bank of $r = 36$ PDA filters ($r_1 = r_2 = 6$) with the noise ranges defined by $\sigma_m^0 = 0$, $\sigma_m^6 = 0.003$ and $R^0 = 0.05$, $R^6 = 0.1$.

Estimation Results with the MMPDAF

Denoting as M_{sl} the model corresponding the s -th value of σ_m and l -th value of R , the estimates of these two noise level parameters are

$$\hat{\sigma}_m(k) = \sum_{s=1}^{r_1} \sum_{l=1}^{r_2} \sigma_m^s P\{M_{sl}|Z^k\} \quad (4.3.3-3)$$

$$\hat{R}(k) = \sum_{s=1}^{r_1} \sum_{l=1}^{r_2} R^l P\{M_{sl}|Z^k\} \quad (4.3.3-4)$$

The evolution in time of these estimates is shown in Figure 4.3.3-3. The prior probabilities were chosen equal for each model. The estimate $\hat{\sigma}_m$ is visibly affected by the maneuver of the target at $k = 500$. The estimate \hat{R} is remarkably stable, except during the maneuver, but then it settles again. The corresponding position, velocity and acceleration estimates are presented in Figures 4.3.3-4 and 4.3.3-5.

4.3.3 Application of the MMPDAF to Track a Real Target

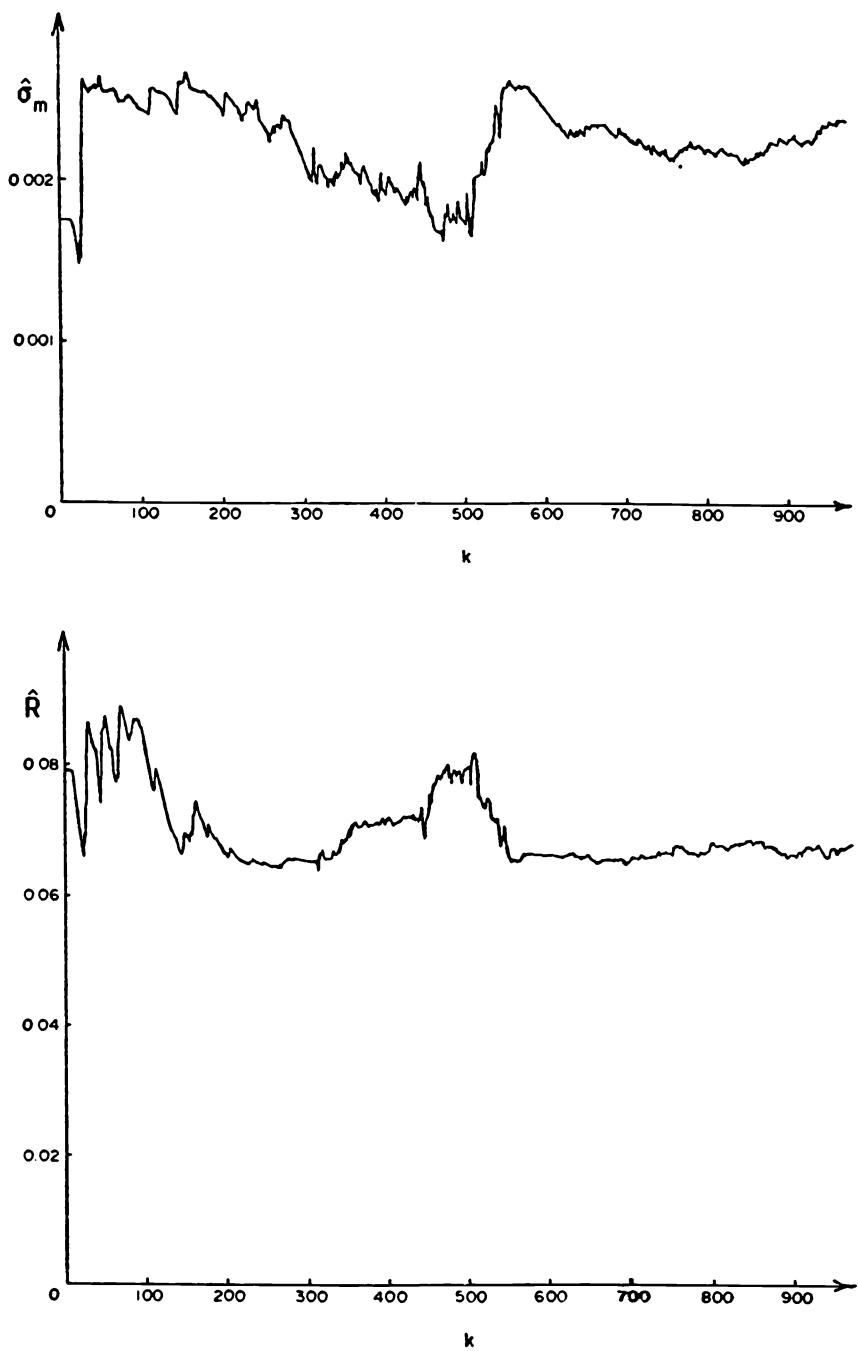


Figure 4.3.3-3: Estimates of the standard deviation of the process noise and the variance of the measurement noise.

4.3.3 Application of the MMPDAF to Track a Real Target

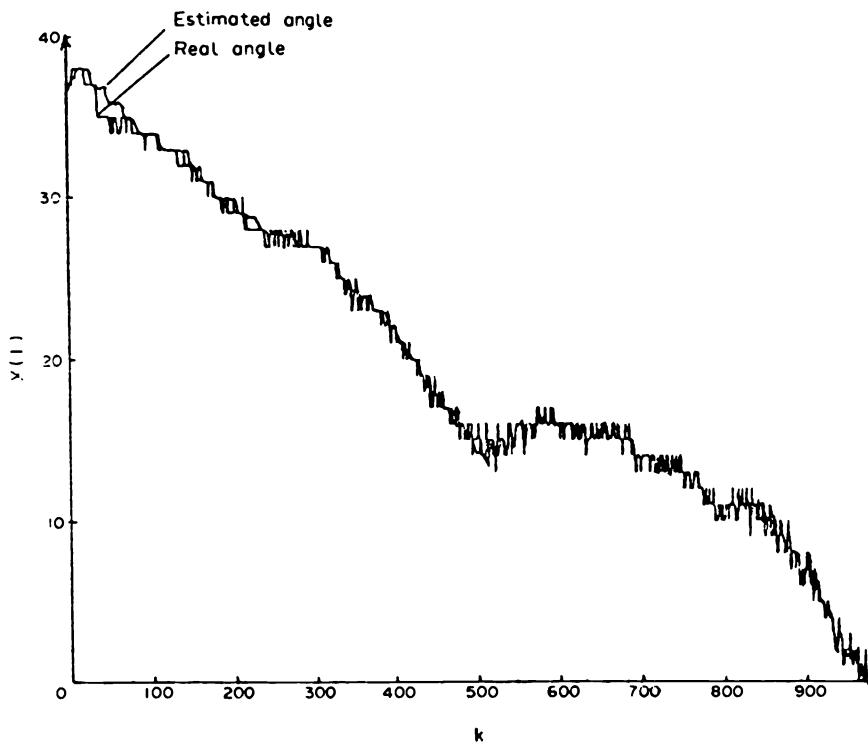


Figure 4.3.3-4: True and estimated position from the MMPDAF.

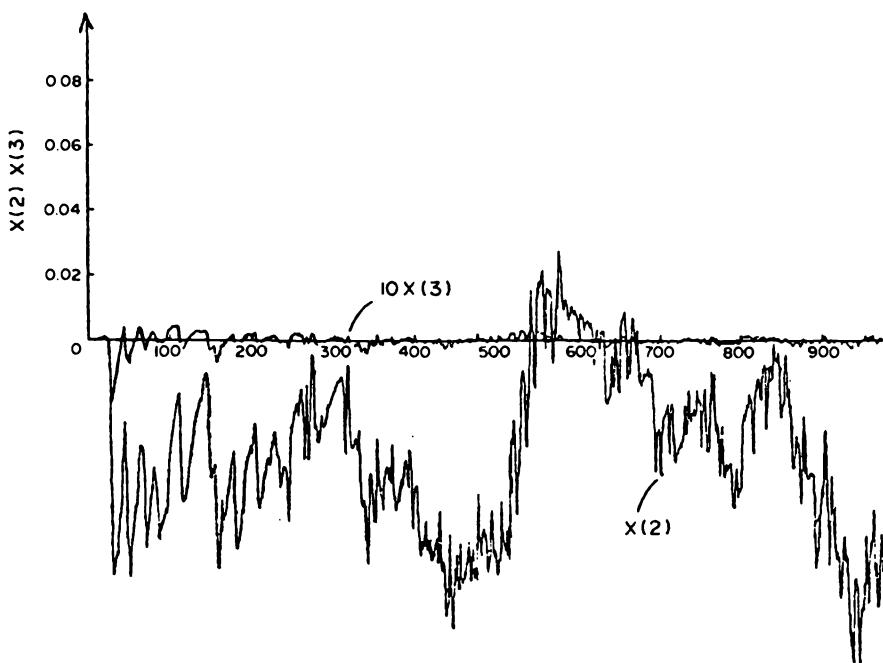


Figure 4.3.3-5: Velocity and acceleration estimates the MMPDAF.

4.3.3 Application of the MMPDAF to Track a Real Target

Adaptive Grid Approach

Since in practice one might have a less accurate idea about the range of the parameters one has to start with a preliminary set of models — a coarse grid over the space of unknown parameters.

For the problem under consideration the following initial coarse grid, also 6×6 , was run: $\sigma_m^0 = 0$, $\sigma_m^6 = 0.03$, and $R^0 = 0.05$, $R^6 = 1$.

The evolution in time of the estimates of σ_m and R is plotted for this coarse grid in Figure 4.3.3-6. The estimates rapidly settle to the lowest value in each variable, $\hat{\sigma}_m = 0.005$ and $\hat{R} = 0.21$, indicating that the “true” values are outside the range of this initial coarse grid.

Other adaptive approaches are described in [LB92].

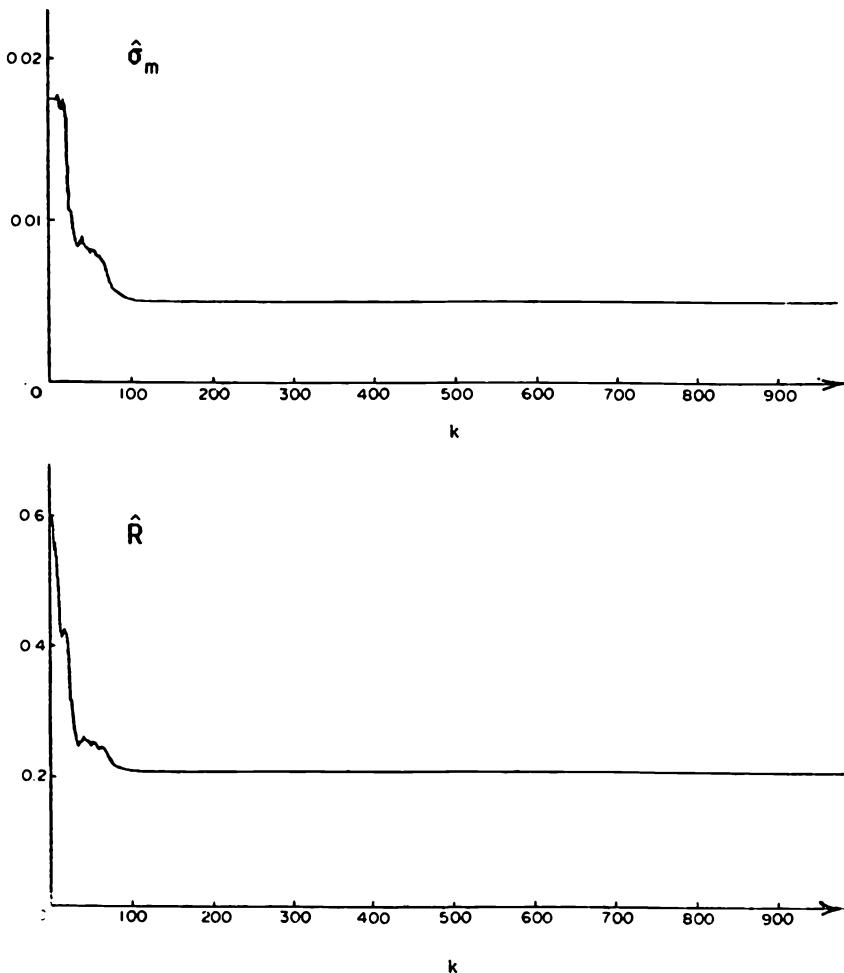


Figure 4.3.3-6: Estimates of the noise levels from the coarse grid MMPDAF.

4.3.4 The MMPDAF — Summary

4.3.4 The MMPDAF — Summary

The MMPDAF can track a target with uncertain model (because of possible maneuvers) in clutter with a number of PDAFs, based on different models, that are running in parallel.

Typically, the models differ in:

- the level of process noise and/or
- the level of measurement noise

They can also differ in

- the state models
- the dimension of the state vector

Each model has a prior probability of being the correct one.

The state estimate is obtained by evaluating

- the probability of each measurement being the correct one conditioned on a certain model, and
- the current probability of each model (target mode).

The probability of each model is updated using Bayes' formula. This uses the likelihood functions of the mode-matched PDAFs — uniform-Gaussian mixtures.

In order to allow this algorithm to be effective in a non-stationary environment (switching models) the GPB_n or IMM techniques, which assume the models to switch according to a Markov chain, can be applied in conjunction with the PDAF.

4.4 MULTIPLE MODEL TRACK FORMATION AND MAINTENANCE

4.4.1 Track Formation, Maintenance and Termination with the IMMPDAF

The ***Interacting Multiple Model PDAF (IMMPDAF)*** consists of the IMM with state estimation modules being PDAFs rather than KFs. It can be used to

- (i) form (initiate) tracks,
- (ii) carry out track maintenance for maneuvering targets,
- (iii) terminate tracks.

The transition between the above functions can be done based on the ***True Target Probability (TTP)*** that this algorithm can evaluate for a track.

This is accomplished by using a so-called ***undetectable target*** or ***no target*** model that can stand for

- (a) a true target whose detection probability has become zero,
- (b) a false (erroneously hypothesized) target, or
- (c) a target that has vanished.

These three possibilities can be modelled by the *same mathematical model*: any measurements in this case can originate only from clutter or false alarms.

Items (i)–(iii) can be accomplished with the following set of models:

M_1 : undetectable target (“no target” or false or vanished)

M_2 : nearly constant velocity (2nd order) — for uniform motion

M_3 : Wiener process acceleration (3rd order) with large noise (acceleration increment) — for the onset or end of acceleration

M_4 : Wiener process acceleration with small noise (acceleration increment) — for the nearly constant acceleration periods.

Other models, including nonlinear ones, can also be used.

Track Formation

The process of **track formation** with a **fixed window** consists of

1. **Track initiation (start-up).** Any pair of measurements from $k = 1$ and $k = 2$ within a maximum distance based on target maximum motion parameters and measurement noise variances initiates a **preliminary track**.

The initial state estimate and its covariance can be obtained from the two-point differencing technique described in Subsection 5.5.3 of [BL93].

2. **Preliminary tracking.** For $k = 3, \dots, N_w$ (the window length, e.g., $N_w = 7$) only two models, M_1 and M_2 , are used. For each track the TTP

$$\mu_2(k) = P\{M(k) = M_2 | Z^k\} \triangleq P\{M_2(k) | Z^k\} \quad (4.4.1-1)$$

is evaluated assuming an *underlying Markov chain transition matrix* between M_1 and M_2 , e.g.,

$$\begin{matrix} & M_1 & M_2 \\ M_1 & 0.98 & 0.02 \\ M_2 & 0.02 & 0.98 \end{matrix}$$

These transitions model the possibility of a target “disappearing/reappearing”¹ or, to correct an erroneous TTP.

Each model can be assumed initially equiprobable.

This technique falls in the category **track before declare**², i.e., a tracking algorithm is already used before “declaring a target” — accepting the track as a target.

3. **Track confirmation.** A track is confirmed if its TTP is above some threshold.

In the **sliding window** implementation new tracks are started with *every unassociated measurement at every time k*. Additional criteria for track confirmation, e.g., the **track age**, can also be used with the TTP.

Track Termination

Track termination can be accomplished as follows: if the TTP falls below a threshold the track is eliminated.

¹These are the mathematical models of sudden death/resurrection.

²Sometimes this is called **track before detect** where detection refers to target track detection (acceptance). Strictly speaking, track before detect refers to the case where thresholding (detection) is done only following multiple frame signal processing (see Subsection 1.4.2).

Track Maintenance

Precision tracking of a confirmed target which might maneuver — **track maintenance** — can be done with, e.g., the 4 models listed earlier.

The following Markov chain transition matrix can be used in this case:

$$\begin{array}{ccccc} & M_1 & M_2 & M_3 & M_4 \\ M_1 & 0.98 & 0.02 & 0 & 0 \\ M_2 & 0.02 & 0.88 & 0.1 & 0 \\ M_3 & 0 & 0.33 & 0.34 & 0.33 \\ M_4 & 0 & 0 & 0.2 & 0.8 \end{array}$$

The use of the “no target” model allows eventual elimination of false tracks not discarded during the track formation or recognizing a lost/disappeared target.

Likelihood Function Evaluation

The likelihood function for the mode-probability computation is to be evaluated as follows.

(a) Each PDAF will have to use the *same validation region*, equal to the union of the mode-conditioned regions. In practice, the largest of them (from M_3 in the case of the above models) is approximately equal to the union and thus can be used instead. The corresponding ellipse can be centered at the predicted measurement of the same filter.

(b) The **likelihood function** corresponding to filter matched to mode (model) j is, with $m(k)$ the number of validated measurements,

$$\Lambda_j[Z(k)] = \frac{1 - P_D^j}{V^{m(k)}} + \frac{P_D^j}{m(k)V^{m(k)-1}} \sum_{i=1}^{m(k)} \mathcal{N}[z_i(k); \hat{z}^j(k|k-1), S^j(k)] \quad (4.4.1-2)$$

where the mode-conditioned target detection probability is

$$P_D^j = \begin{cases} 0 & \text{if } j = 1 \\ P_D & \text{if } j > 1 \end{cases} \quad (4.4.1-3)$$

P_D is the target detection probability, $\hat{z}^j(k|k-1)$, $S^j(k)$ are the predicted measurement and its covariance for model j .

The state model for M_1 can be the same as for M_2 .

This algorithm has been implemented in MULTIDATTM.

4.4.2 Automatic Track Formation with the IMMPDAF

4.4.2 Automatic Track Formation with the IMMPDAF

Modeling Assumptions

The **automatic track formation (ATF)** based on the IMMPDAF consists of

- the IMM algorithm (Subsection 11.6.6 of [BL93]) with
- PDAF modules corresponding to
 - Mode M_2 : observable (true) target with $P_D^2 = P_D$ (known or hypothesized)
 - Mode M_1 : unobservable (false) target with $P_D^1 = 0$.

The *target motion* is modelled in Cartesian coordinates by a nearly constant velocity model — the same for both modes.

The *locations of the false measurements* are modelled as uniformly distributed.

The *number of the false measurements* are assumed to have a **diffuse prior** (any number of false measurements is equiprobable). This allows the use of a state estimation algorithm that does not require the spatial density of the false measurements (clutter): nonparametric PDAF — suitable for a **heterogeneous clutter** environment.

The *Markov chain model for the observable/unobservable situation* is as follows. Denoting by $M(k)$ the mode in effect during period k , the following transition or **mode switching probabilities** will be assumed:

$$P\{M(k+1) = \text{unobservable} | M(k) = \text{unobservable}\} = p_{11} = 1 - \epsilon_1 \quad (4.4.2-1)$$

$$P\{M(k+1) = \text{observable} | M(k) = \text{unobservable}\} = p_{12} = \epsilon_1 \quad (4.4.2-2)$$

$$P\{M(k+1) = \text{unobservable} | M(k) = \text{observable}\} = p_{21} = \epsilon_2 \quad (4.4.2-3)$$

$$P\{M(k+1) = \text{observable} | M(k) = \text{observable}\} = p_{22} = 1 - \epsilon_2 \quad (4.4.2-4)$$

i.e., with transitions between the modes assumed with some low probability.

4.4.2 Automatic Track Formation with the IMMPDAF

Implementation (fixed window)

*Every detection at time (scan or frame) 1 is an **initiator**.*

Assuming a 2-dimensional measurement, the **association region** at time 2 is taken as the rectangle with area

$$V(2) = [2(v_{1\max} T + 2\sqrt{R_{11}})] [2(v_{2\max} T + 2\sqrt{R_{22}})] \quad (4.4.2-5)$$

where $v_{1\max}$, $v_{2\max}$, are the maximum velocities of the target in the two measured coordinates, R_{11} , R_{22} , are the corresponding measurement noise variances and T is the time interval between the two samples.

This rectangular gate is centered at the first measurement and, as shown above, is enlarged to account for the measurement noise.

Each measurement in such a gate yields an **initiating pair**, from which a **preliminary track** is started.

Starting from time 3, a 2-model IMMPDAF, as described before, is run on each preliminary track. The state estimate initialization is done using the initiating pair from the first two times.

Each mode is assumed to have initial probability 0.5.

The IMMPDAF algorithm computes the **true target probability (TTP)** — the probability of mode 2 — of each track and discards the tracks with TTP below a certain threshold, e.g., 0.05.

Since this algorithm provides

- a state estimate, as well as
- a *quantification* of the **credibility of the track** — the TTP

it can be said that it is a (reasonably) **intelligent tracker**.

4.4.3 Simulation Results

4.4.3 Simulation Results

The results presented next were obtained using the fixed window implementation of the IMMPDAF track formation algorithm in MULTIDAT™.

Since a real-time application requires a sliding window implementation, the technique described in Subsection 4.4.6 was used to obtain the sliding window performance figures.

Parameters of the Problem

Maximum target velocity $v_{i_{\max}} = 40$, $i = 1, 2$.

Measurement noise variances $R_{11} = R_{22} = 25$.

Sampling period $T = 1$.

In the surveillance region, which had an area of 2.4E5, there was one target moving with a velocity vector of [30 20].

The target detection probability was taken as 0.9 and 0.8.

Two levels of clutter were considered: heavy and medium.

Estimator Design Parameters

Process noise: $q_j = 0.1$ for each of the two coordinates — maneuvering index 6% — for both models (different values can be taken for the two models if desired).

Mode switching probabilities:

$$p_{11} = p_{22} = 0.98 \quad p_{12} = p_{21} = 0.02$$

Minimum value of the TTP of a track in order not to be discarded: 0.05.

Figure 4.4.3-1 shows the track of the target as formed by the IMMPDAF track formation algorithm in a sample run with the heavy clutter and target $P_D = 0.9$ (in this run there was no false track).

The ellipses represent the the 2σ uncertainty for the estimated target position. The target was not detected at $k = 4$ in this run — note the increase in the uncertainty ellipse.

The track formation algorithm knew only the value of P_D , but used no information about the spatial density of the false alarms.

Table 4.4.3-1 presents the track formation performance of the IMMPDAF vs. GPB1PDAF for two levels of clutter. The evaluations are based on 300 Monte Carlo runs for the heavy clutter and 1000 runs for the lower level (medium) clutter.

4.4.3 Simulation Results

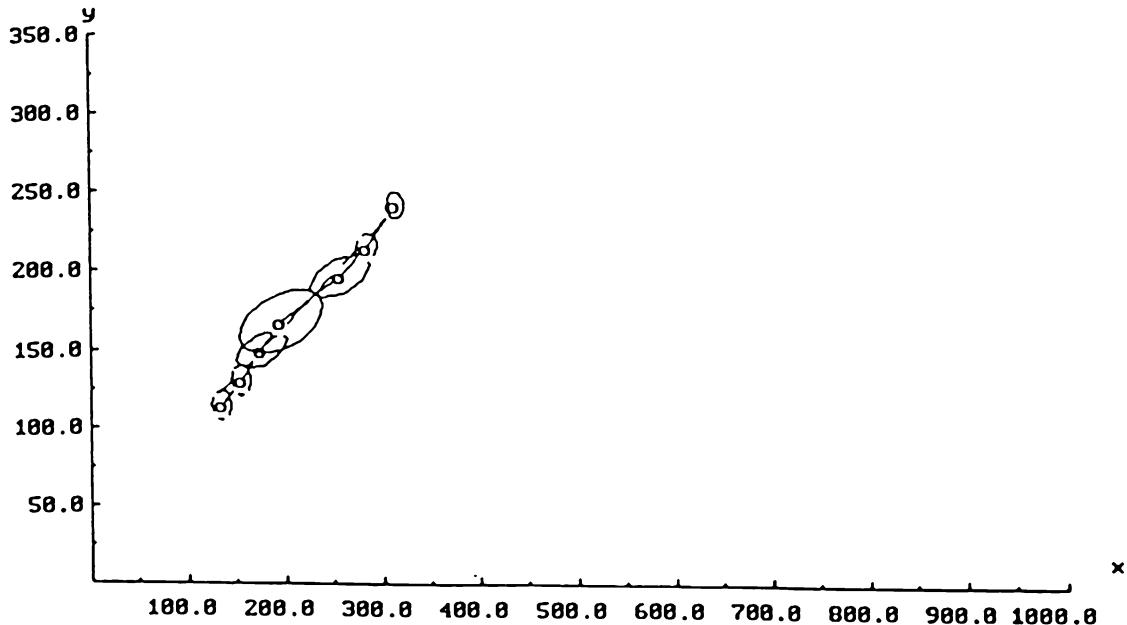


Figure 4.4.3-1: True target track formed in a sample run.

The heavy clutter corresponds to a false alarm probability per resolution cell (whose size is about $10\sqrt{R_{11}R_{22}}$) of 0.025. In this case each initiator yielded, on the average, one preliminary track.

Such a situation occurs if, due to a **stealthy target**, the detection threshold has to be set low in order to detect the target with a reasonably high probability. Such a high false alarm rate can be handled by this algorithm.

For the heavy clutter case, the density of false tracks after 7 scans with the IMMPDAF was significantly lower than with the GPB1PDAF.

The improvement ratios — false alarms to false tracks with $TTP > 0.5$ — are, for the IMMPDAF, 200 and 2,500 for the high and medium clutter densities, respectively.

The expected **confirmation time** (time by which $TTP > 0.5$) for the GPB1PDAF was slightly shorter than for the IMMPDAF. These expected confirmation times were evaluated according to the technique of Subsection 4.4.6.

Overall, the IMMPDAF is preferable to the GPB1PDAF because of the significantly lower number of false tracks shown here. The IMMPDAF can be also used to obtain maneuvering target tracking capability.

4.4.3 Simulation Results

	1E-4	1E-4	1E-5	1E-5
Average number of false alarms per unit area	1E-4	1E-4	1E-5	1E-5
Average number of false alarms in area $\sqrt{R_{11}R_{22}}$	2.5E-3	2.5E-3	2.5E-4	2.5E-4
Average number of initiators in surveillance region	24	24	2.4	2.4
Average number of false alarms in 99% gate				
in scan 2	1	1	.1	.1
in scan 3	.35	.35	.035	.035
Target detection probability	.9	.8	.9	.8
Average number of false tracks after 7 scans				
in surveillance region	I .61	2.6	1E-3	3E-3
	G 3.1	5.4	1E-3	4E-3
per unit area	I 2.6E-6	1.1E-5	4E-9	1.2E-8
	G 1.3E-5	2.3E-5	4E-9	1.7E-8
in area $\sqrt{R_{11}R_{22}}$	I 6.1E-5	2.7E-4	1E-7	3E-7
	G 3.1E-4	5.7E-4	1E-7	4E-7
Average TTP of false tracks after 7 scans	I .29	.24	.95*	.13
	G .29	.28	.50*	.27
Fraction of false tracks with TTP > .5	I .20	.09	1*	0
	G .15	.13	0*	.25
Fixed window (7 scans) target track confirmation probability				
	I .68	.66	.74	.58
	G .78	.74	.74	.58
Average TTP of target track	I			
at scan 3		.54	.50	.56
at scan 4		.65	.54	.65
at scan 5		.69	.59	.72
at scan 6		.74	.61	.77
at scan 7		.76	.63	.79
Average TTP of target track	G			
at scan 3		.55	.50	.56
at scan 4		.64	.56	.66
at scan 5		.68	.60	.76
at scan 6		.73	.62	.82
at scan 7		.74	.64	.85
Fraction of target tracks at scan 7 with TTP > .5	I .84	.64	.87	.80
	G .79	.69	.92	.81
Sliding window confirmation probability	I			
at scan 8		.75	.76	.81
at scan 9		.82	.86	.89
at scan 10		.92	.93	.95
at scan 11		.94	.96	.97
at scan 12		.96	.97	.99
Sliding window expected confirmation time	I 7.98	7.78	7.90	8.21
	G 7.47	7.48	7.90	8.21

Table 4.4.3-1: Performance of the IMMPDAF (“I”) vs. GPB1PDAF (“G”) in track formation (* statistically not significant — based on a single outcome in 1000 runs).

4.4.4 A Real-Data Electro-Optical Sensor Problem

This example illustrates the use of the **ATF** (Automatic Track Formation with the IMM-PDAF) routine from **MULTIDATTM** on a “blind” data set — generated with *unspecified parameters*.

The Scenario

Real clutter data (from an actual image) on which an unspecified number of targets were superimposed, shown in Figures 4.4.4-1 and 4.4.4-2.

Target motion parameters — unspecified.

Estimator Parameters

Sampling period (normalized) $T = 1$.

Assumed target detection probability $P_D = 0.9$.

Assumed target velocity $|\dot{x}| \leq 10 \text{ pix}/T$ and $|\dot{y}| \leq 10 \text{ pix}/T$.

Motion model: 2nd order kinematic model with acceleration process noise s.d. $\sqrt{Q} = 1 \text{ pixel}/T^2$.

Measurement noise s.d. $\sqrt{R} = 1 \text{ pixel}$.

False alarm density — unknown and *nonhomogenous*.

Two designs with different Markov chain transition matrix between the modes were considered:

- Design 1: “tight” (0.02 switching probabilities)
- Design 2: “loose” (0.10 switching probabilities).

Results

The results for the “tight” Markov chain design are presented in Tables 4.4.4-1, 4.4.4-2 and Figures 4.4.4-3 through 4.4.4-5. Table 4.4.4-3 shows how one can eliminate redundant (“piggyback”) tracks.

The results for the “loose” Markov chain design are given in Tables 4.4.4-4, 4.4.4-5 and Figure 4.4.4-6.

While Design 1 is superior to Design 2, one can see the **robustness** of the algorithm: the “looser” Markov chain yields only one extra target with a very low TTP and thus it can be discarded.

4.4.4 A Real-Data Electro-Optical Sensor Problem

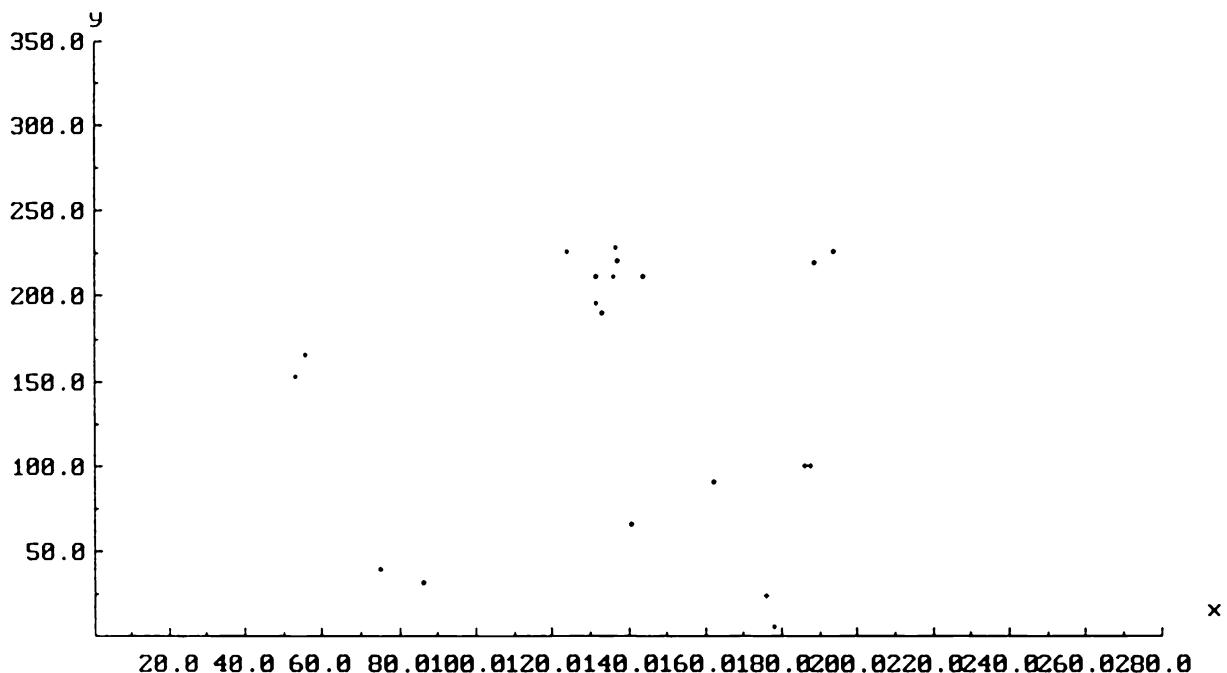


Figure 4.4.4-1: Data from frame 1 (coordinates in pixels).

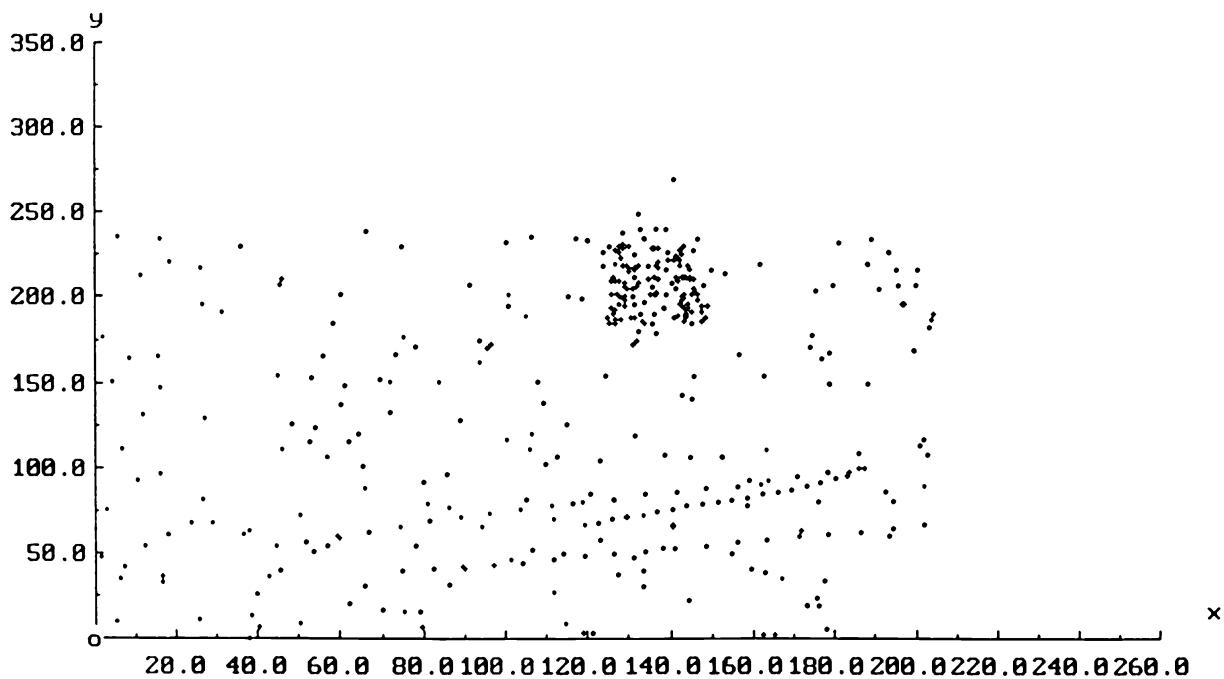


Figure 4.4.4-2: Data from 20 frames overlayed.

4.4.4 A Real-Data Electro-Optical Sensor Problem

```

C:\MDAT>runm33
AUTOMATIC TRACK FORMATION : 2 MODELS WITH PDAF

use default input data ? (y/n)
y
use the default file name deflt33.dat ? (y/n)
y
do you want to change default values ? (y/n)
y
default transition matrix between live (#2) and dead (#1) models is :
.9000E+00 .1000E+00
.1000E+00 .9000E+00
use the default transition matrix ? (y/n)
n
input new transition matrix (f)
0.98 0.02                                Best choice
0.02 0.98

the default prior process noise variances are :
model 1 (x, y) : 1.0000000 1.0000000
model 2 (x, y) : 1.0000000 1.0000000
use the default numbers ? (y/n)
y
read ground truth from file truthc.dat ? (y/n)

y
read measurements from file lxt.dat ? (y/n)
y
default measurement noise variances are 1.00 1.00
use default measurement noise variances ? (y/n)
y
default PD (target detection probability) is 9.000000E-001
use default values ? (y/n)
y
write estimated results into file newtrk.dat ? (y/n)
y
default fixed window size is 7
use default fixed window size ? (y/n)
y
default pruning threshold for track prob. is 1.000000E-001
use default threshold ? (y/n)
y
default target max velocity is 10.0000000
use default value ? (y/n)
y

```

Table 4.4.4-1: ATF routine from MULTIDATTM: The input parameters (Design 1).

4.4.4 A Real-Data Electro-Optical Sensor Problem

number of surviving tracks is	5
these are the surviving tracks	
track index, old index	(1)
track probability [9.927726E-001]	2
latest state estimate :	
.1191E+03 .7341E+01 .4875E+02 .1776E+01	
corresponding S.D. :	
.9083E+00 .1018E+01 .9094E+00 .1019E+01	
normalized state error is : 20133.3000000	
track index, old index	(2)
track probability [9.923136E-001]	3
latest state estimate :	
.1414E+03 -.7431E+01 .8644E+02 -.1895E+01	
corresponding S.D. :	
.9106E+00 .1019E+01 .9165E+00 .1021E+01	
normalized state error is : 33054.6900000	
track index, old index	(3)
track probability [9.918557E-001]	4
latest state estimate :	
.1660E+03 -.3612E+01 .8615E+02 -.1738E+01	
track index, old index	(4)
track probability [9.919762E-001]	5
latest state estimate :	
.1414E+03 -.7389E+01 .8644E+02 -.1895E+01	
corresponding S.D. :	
.9109E+00 .1020E+01 .9166E+00 .1022E+01	
normalized state error is : 33035.4100000	
track index, old index	(5)
track probability [9.921447E-001]	6
latest state estimate :	
.1660E+03 -.3571E+01 .8615E+02 -.1737E+01	
corresponding S.D. :	
.9115E+00 .1019E+01 .9193E+00 .1022E+01	
normalized state error is : 41975.5800000	

Table 4.4.4-2: Summary Output: Tracks at $k = 7$ (Design 1).

4.4.4 A Real-Data Electro-Optical Sensor Problem

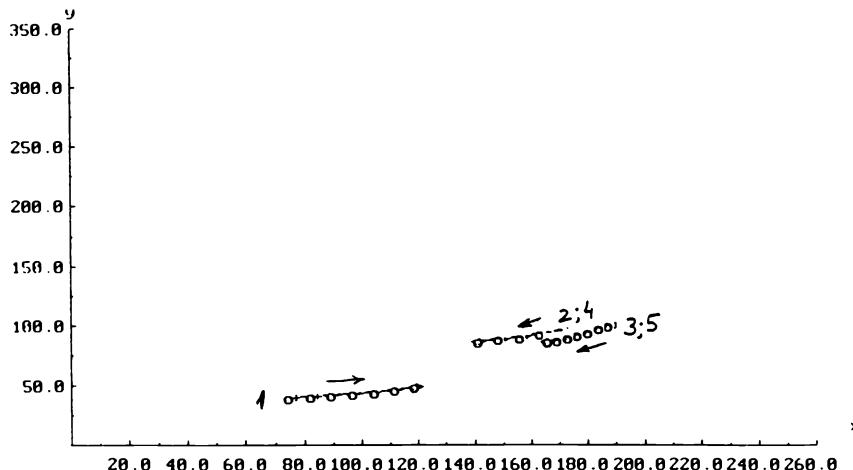


Figure 4.4.4-3: The 5 accepted tracks.

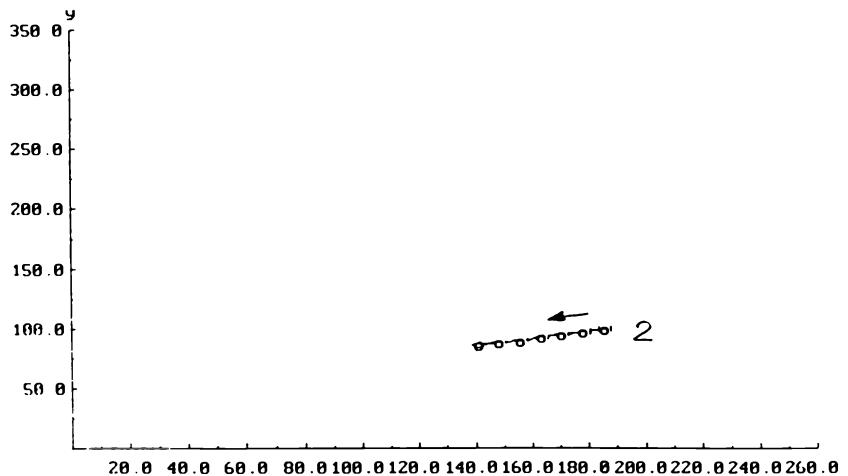


Figure 4.4.4-4: Track #2.

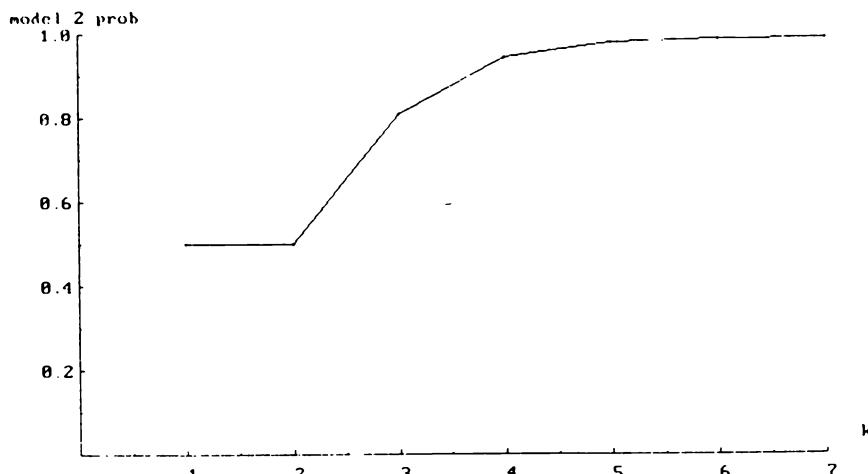


Figure 4.4.4-5: The TTP for track #1.

4.4.4 A Real-Data Electro-Optical Sensor Problem

```
plot` a single track ? (y/n)
n
more plotting ? (y/n)
n
do you want to check the similarity between tracks ? (y/n)
y
please input track numbers to be checked (2I)
2 2
the results of chi-square (4 d.f.) test is      2350.0370000
do you want to check another track pair ? (y/n)
y
please input track numbers to be checked (2I)
2 3
the results of chi-square (4 d.f.) test is      472.4305000
do you want to check another track pair ? (y/n)
y
please input track numbers to be checked (2I)
2 4
the results of chi-square (4 d.f.) test is   1.281093E-003
do you want to check another track pair ? (y/n)
y
please input track numbers to be checked (2I)
2 5
the results of chi-square (4 d.f.) test is      471.9224000
do you want to check another track pair ? (y/n)
y
please input track numbers to be checked (2I)
3 4
the results of chi-square (4 d.f.) test is      473.0249000
do you want to check another track pair ? (y/n)
y
please input track numbers to be checked (2I)
3 5
the results of chi-square (4 d.f.) test is   1.221853E-003
do you want to check another track pair ? (y/n)
y
please input track numbers to be checked (2I)
4 5
the results of chi-square (4 d.f.) test is      472.5156000
do you want to check another track pair ? (y/n)
```

Table 4.4.4-3: Similarity tests for initiated tracks.

4.4.4 A Real-Data Electro-Optical Sensor Problem

```

use the default file name deflt33.dat ? (y/n)
y
do you want to change default values ? (y/n)
y
default transition matrix between live (#2) and dead (#1) models is :
.9000E+00 .1000E+00
.1000E+00 .9000E+00
use the default transition matrix ? (y/n)
y
the default prior process noise variances are :
model 1 (x, y) : 1.0000000 1.0000000
model 2 (x, y) : 1.0000000 1.0000000
use the default numbers ? (y/n)
y
read ground truth from file truthc.dat ? (y/n)
y
read measurements from file lxt.dat ? (y/n)
y
default measurement noise variances are 1.00 1.00
use default measurement noise variances ? (y/n)

y
default PD (target detection probability) is 9.000000E-001
use default values ? (y/n)
y
write estimated results into file newtrk.dat ? (y/n)
y
default fixed window size is 7
use default fixed window size ? (y/n)
y
default pruning threshold for track prob. is 1.000000E-001
use default threshold ? (y/n)
y
default target max velocity is 10.0000000
use default value ? (y/n)
y

```

Table 4.4.4-4: ATF routine input for Design 2 (“loose” transition matrix between the modes).

4.4.4 A Real-Data Electro-Optical Sensor Problem

```

track      1 probability : 9.655476E-001
with latest estimate :
  .1191E+03  .7341E+01  .4875E+02  .1776E+01
track      2 probability : 9.633735E-001
with latest estimate :
  .1414E+03  -.7431E+01  .8644E+02  -.1895E+01
track      3 probability : 9.622643E-001
with latest estimate :
  .1660E+03  -.3612E+01  .8615E+02  -.1738E+01
track      4 probability : 9.630306E-001
with latest estimate :
  .1414E+03  -.7389E+01  .8644E+02  -.1895E+01
track      5 probability : 9.625626E-001
with latest estimate :
  .1660E+03  -.3571E+01  .8615E+02  -.1737E+01
track      6 probability : 1.417346E-001
with latest estimate :
  .1418E+03  .3530E+01  .2010E+03  .7010E+00
plot all surviving tracks simultaneously ? (y/n)
y
push enter to start plotting

```

Table 4.4.4-5: ATF routine output for Design 2 with additional weak track (TTP = 0.14) from the cloud.



Figure 4.4.4-6: Tracks formed in Design 2 with the additional track.

4.4.5 Track Formation with Multiple Models — Summary

4.4.5 Track Formation with Multiple Models — Summary

An algorithm for automatic track formation in clutter has been presented that

- consists of the IMM combined with PDAF modules, and
- relies on the key concept of “target observability” which has been put in the general framework of dynamic multiple model (hybrid systems) estimation.

Two models, differing only in the detection probability P_D , are considered in the process of the track formation: one for a target with a high P_D , and one for an “unobservable target,” defined by $P_D = 0$.

This algorithm is useful for low SNR situations where the detection threshold has to be set low in order to detect the target and this leads to a high rate of false alarms.

This technique of track formation can be naturally followed by the maneuvering target tracking technique consisting of the same algorithm with additional models for the target’s maneuvering modes.

This technique provides a systematic way, via a simple recursive algorithm, to obtain the *probability that there is a target in track* as well as *maneuvering target tracking capability in clutter*.

4.4.6 Performance Evaluation of the Automatic Track Formation Algorithm

The evaluation of the performance of the ATF algorithm operating in a *sliding window* mode, based on *fixed window simulation results*, can be obtained as follows.

Denote by D the detection event at times 1 and 2 and by A the event of acceptance of the resulting preliminary track as a confirmed track by the end of the initiation window. Define the following states of a Markov chain:

1. Initial state: $D \rightarrow 2, \bar{D} \rightarrow 1$.
2. First detection: $D \rightarrow 3, \bar{D} \rightarrow 1$.
3. Second detection: $A \rightarrow 4, \bar{A} \rightarrow 1$.
4. Confirmed target.

The final confirmation probability (for a fixed window) of a true target's track is, in terms of the above quantities,

$$P_{C_{FW}} = P_D^2 P_A \quad (4.4.6-1)$$

The transition probabilities of the Markov chain are

$$\begin{aligned} \pi_{11} &= 1 - P_D & \pi_{12} &= P_D \\ \pi_{21} &= 1 - P_D & \pi_{23} &= P_D \\ \pi_{31} &= 1 - P_A & \pi_{34} &= P_A \\ \pi_{44} &= 1 \end{aligned} \quad (4.4.6-2)$$

and the initial state is

$$\mu_1(0) = 1 \quad (4.4.6-3)$$

The Monte Carlo simulations of the ATF/IMMPDA for a fixed window, N_W , yield $P_{C_{FW}}$; then P_A follows from the knowledge of P_D .

Note that the last step in the Markov chain is $N_W - 2$ system sampling periods. Thus the probability of the last state of the chain after k chain time units is the **confirmation probability** from a sliding window after $k + N_W - 3$ system periods

$$\mu_C(k + N_W - 3) = \mu_4(k) \quad (4.4.6-4)$$

and the expected confirmation time (in system sampling periods) is

$$\bar{t}_C = \sum_{i=1}^{\infty} (i + N_W - 3) \mu_4(i) \quad (4.4.6-5)$$

The cumulative distribution function of the sliding window confirmation time is

$$P_{C_{SW}}(k + N_W - 3) = \sum_{i=1}^k \mu_4(i) \quad (4.4.6-6)$$

4.5 MULTISENSOR TRACKING OF A MANEUVERING TARGET IN CLUTTER

4.5.1 Introduction

A **multisensor IMMPDAF** is presented for track maintenance of

- a highly maneuvering target (sea skimmer with $6g$ accelerations)
- with measurements from different sensors
- in a cluttered environment.

The algorithm carries out the association of the measurements given by

- a 2-D radar (range and azimuth angle) and
- an infrared sensor (azimuth and elevation angles)

assumed collocated and synchronized.

Track initialization is assumed to have been made.

The sensors make measurements in spherical coordinates, but, the motion of the target is represented in Cartesian coordinates.

To accommodate the fact that the target can be highly maneuvering, the IMM algorithm (Subsection 1.5.4) is used on 2 or 3 models of *different dimension*.

The following models are used (in each Cartesian coordinate in a 3-D space):

Model 1 — nearly constant velocity motion (2-dimensional state)

Model 2 — Wiener process acceleration (3-dimensional state).

This is in accordance to the Variable State Dimension approach (Section 11.4 of [BL93]) that it is best to track a target using the lowest possible state dimension.

The **Multisensor PDAF (MSPDAF)**, consisting of the nonparametric PDAF (Subsection 3.4.3) with **sequential state update** between the sensors (Subsection 2.2.2), is used for the estimation modules of the IMM.

4.5.2 The Target and Sensor Models

4.5.2 The Target and Sensor Models

The target dynamics are modelled in Cartesian coordinates as

$$x(k+1) = Fx(k) + v(k) \quad k = 0, 1, \dots \quad (4.5.2-1)$$

The state of the target at time k , $x(k)$, is

- for Model 1 — a 6-dimensional vector (position and velocity in each of the 3 Cartesian coordinates)
- for Model 2 — a 9-dimensional vector (position, velocity, and acceleration in each of the 3 Cartesian coordinates)

The transition matrix F is

$$F = \begin{bmatrix} F_t & 0 & 0 \\ 0 & F_t & 0 \\ 0 & 0 & F_t \end{bmatrix} \quad t = 1, 2 \quad (4.5.2-2)$$

where, with the sampling time T

$$F_1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad F_2 = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (4.5.2-3)$$

The process noise $v(k)$ is assumed zero-mean white with known covariance

$$Q = \begin{bmatrix} u_t & 0 & 0 \\ 0 & u_t & 0 \\ 0 & 0 & u_t \end{bmatrix} \quad t = 1, 2 \quad (4.5.2-4)$$

with

$$u_1 = \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} q_1 \quad (4.5.2-5)$$

$$u_2 = \begin{bmatrix} T^4/4 & T^3/2 & T^2/2 \\ T^3/2 & T^2 & T \\ T^2/2 & T & 1 \end{bmatrix} q_2 \quad (4.5.2-6)$$

where q_1 is the variance of the process noise modeling the *acceleration* in Model 1 and q_2 is the variance of the process noise modeling the *acceleration increment over a sampling period* in Model 2.

4.5.2 The Target and Sensor Models

The measurements from sensor i are

$$z^i(k) = h^i[x(k)] + w^i(k) \quad k = 0, 1, \dots \quad i = 1, 2 \quad (4.5.2-7)$$

with the two-dimensional vector $z^i(k)$ consisting of

- range and azimuth for sensor 1 (radar)
- azimuth and elevation angles for sensor 2 (infrared);

and h^i the transformation from Cartesian to sensor coordinates.

The range, azimuth and elevation transformations are given by

$$r = (x^2 + y^2 + z^2)^{1/2} \quad a = \tan^{-1}(y/x) \quad e = \tan^{-1}[z/(x^2 + y^2)^{1/2}] \quad (4.5.2-8)$$

The measurement noise $w^i(k)$ for sensor i is assumed zero-mean white with known covariance

$$R^1 = \text{diag}[q_{a1}, q_r] \quad (4.5.2-9)$$

with q_{a1} , q_r the variances for the radar azimuth and range measurements noises, respectively, and

$$R^2 = \text{diag}[q_{a2}, q_e] \quad (4.5.2-10)$$

with q_{a2} , q_e the variances for the infrared sensor azimuth and elevation measurement noises, respectively.

Remark

These models assume decoupled motion across coordinates. The use of the PDAF, however, will introduce coupling via the “spread of the innovations” term, which will yield a full covariance matrix.

4.5.3 The Multisensor PDAF

For two sensors, the **Multisensor PDAF (MSPDAF)** with sequential implementation of the estimation across the sensors consists of the following steps:

Step 1: Prediction.

With $\hat{x}(k-1|k-1)$ and its covariance, one computes the predicted state $\hat{x}(k|k-1)$ and its covariance $P(k|k-1)$, the predicted measurement for sensor 1, $\hat{z}^1(k)$, and the corresponding covariance $S^1(k)$, using the standard prediction equations.

Step 2: Measurement association for sensor 1.

With $\hat{z}^1(k)$ and $S^1(k)$, the set of validated measurements is obtained and their association probabilities are calculated as in the standard PDA procedure.

Step 3: State estimation with the validated measurements from sensor 1.

The state estimate $\hat{x}^1(k|k)$ and its covariance $P^1(k|k)$ are computed using the standard PDAF update equations with the following modification:

Since the measurements are nonlinear, the PDAF is EKF-based rather than KF-based — it utilizes the Jacobian matrix of the measurement function, which for sensor 1 (radar) is

$$H^1(k) = \begin{bmatrix} -\frac{\sin a}{r \cos e} & 0 & 0 & \frac{\cos a}{r \cos e} & 0 & 0 & 0 & 0 \\ \cos e \cos a & 0 & 0 & \cos e \sin a & 0 & 0 & \sin e & 0 \end{bmatrix} \quad (4.5.3-1)$$

where a , e , r are the predicted azimuth, elevation and range measurements of the target, respectively.

Step 4: Measurement association for sensor 2.

With $\hat{x}^1(k|k)$ and $P^1(k|k)$ one computes the “zero-time predicted measurement” (this is a consequence of the sensor synchronicity assumption) for sensor 2 as

$$\hat{z}^2(k) = h^2[\hat{x}^1(k|k)] \quad (4.5.3-2)$$

and the covariance $S^2(k)$ associated with $\hat{z}^2(k)$ according to

$$S^2(k) = H^2(k)P^1(k|k)H^2(k)' + R^2 \quad (4.5.3-3)$$

where the Jacobian matrix of the measurement equation corresponding to sensor 2 (infrared sensor) is

$$H^2(k) = \begin{bmatrix} -\frac{\sin a}{r \cos e} & 0 & 0 & \frac{\cos a}{r \cos e} & 0 & 0 & 0 & 0 \\ -\frac{\sin e \cos a}{r} & 0 & 0 & -\frac{\sin e \sin a}{r} & 0 & 0 & \frac{\cos e}{r} & 0 \end{bmatrix} \quad (4.5.3-4)$$

4.5.3 The Multisensor PDAF

Step 5: State estimation with the validated measurements from sensor 2.

The state estimate $\hat{x}(k|k)$ and its covariance $P(k|k)$ are computed using again the standard PDAF equations with $\hat{x}^1(k|k)$ and $P^1(k|k)$ replacing $\hat{x}(k|k-1)$ and $P(k|k-1)$, respectively, and the Jacobian matrix (4.5.3-4) replacing the measurement matrix.

Remarks

For the MSPDAF for **non-synchronized sensors** one has to calculate predictions for varying time intervals. In this case the prediction equations for the state and its covariance should be available for the necessary arbitrary time intervals (see Subsection 8.7.4).

Also the Markov chain for the mode transitions has to be modified to account for the variable sampling intervals (see Subsection 1.8.3).

4.5.4 Integration of the IMM and the Multisensor PDAF

4.5.4.1 Integration of the IMM and the Multisensor PDAF

The algorithm that combines the IMM architecture with multisensor PDAF (MSPDAF) estimation modules — the **IMM/MSPDAF** — consists of the following steps:

Step 1: Mixing of the estimates from the previous time.

The mixing probabilities are

$$\mu_{s|t}(k-1|k-1) \triangleq P\{M_s(k-1)|M_t(k), Z^{k-1}\} = \frac{1}{\bar{c}_t} p_{st} \mu_s(k-1) \quad s = 1, \dots, r \quad (4.5.4-1)$$

$$\bar{c}_t \triangleq \sum_{s=1}^r p_{st} \mu_s(k-1) \quad (4.5.4-2)$$

and p_{st} is the assumed Markov mode-transition probability from mode M_s at time $k-1$ to mode M_t at time k .

With $\hat{x}^s(k-1|k-1)$ and its covariance $P^s(k-l|k-1)$, the mixed initial condition for the filter matched to model M_t is

$$\hat{x}^{0t}(k-1|k-1) = \sum_{s=1}^r \hat{x}^s(k-1|k-1) \mu_{s|t}(k-1|k-1) \quad t = 1, \dots, r \quad (4.5.4-3)$$

$$\begin{aligned} P^{0t}(k-1|k-1) &= \sum_{s=1}^r \mu_{s|t}(k-1|k-1) \left\{ P^s(k-1|k-1) \right. \\ &\quad + [\hat{x}^s(k-1|k-1) - \hat{x}^{0t}(k-1|k-1)] \\ &\quad \cdot [\hat{x}^s(k-1|k-1) - \hat{x}^{0t}(k-1|k-1)]' \left. \right\} \quad t = 1, \dots, r \end{aligned} \quad (4.5.4-4)$$

Step 2: Predicted states and predicted measurements.

With each mixed estimate, one computes the predicted states:

$$\hat{x}^t(k|k-1) = F^t \hat{x}^{0t}(k-1|k-1) \quad t = 1, \dots, r \quad (4.5.4-5)$$

and their covariances

$$P^t(k|k-1) = F^t P^{0t}(k-1|k-1) (F^t)' + Q^t \quad t = 1, \dots, r \quad (4.5.4-6)$$

where F^t and Q^t are the state transition matrix and process noise covariance matrix, respectively, corresponding to mode M_t .

4.5.4 Integration of the IMM and the Multisensor PDAF

The mode-conditioned predicted measurements for sensor 1 are

$$\hat{z}^{t,1}(k) \triangleq h^1[\hat{x}^t(k|k-1)] \quad t = 1, \dots, r \quad (4.5.4-7)$$

and their covariances

$$S^{t,1}(k) = H^1(k)P^t(k|k-1)H^1(k)' + R^1 \quad t = 1, \dots, r \quad (4.5.4-8)$$

where superscript t stands for mode M_t , h^1 is as in (4.5.2-7), R^1 is given in (4.5.2-9) and H^1 in (4.5.3-1).

Step 3: Measurement validation for sensor 1.

The validation region *has to be taken the same for all models*, e.g., as the largest of them. Then the volume of the γ validation region is (for the 2-dimensional measurement)

$$V^1(k) \triangleq V^{t_1,1}(k) \triangleq \max_t [\gamma\pi|S^{t,1}(k)|^{1/2}] \quad (4.5.4-9)$$

where t_1 is the mode which has the largest validation region for sensor 1.

Assuming this region is “large enough” to include all the candidate measurements for each mode, the measurements are validated according to

$$[z^1(k) - \hat{z}^{t_1,1}(k)][S^{t_1,1}(k)]^{-1}[z^1(k) - \hat{z}^{t_1,1}(k)]' < \gamma \quad (4.5.4-10)$$

Step 4: Update with sensor 1 measurements in each filter $t = 1, \dots, r$.

The association probabilities are computed using $\hat{z}^{t,1}(k)$, $S^{t,1}(k)$, and the *common* volume (4.5.4-9).

Using $\hat{x}^t(k|k-1)$, its covariance $P^t(k|k-1)$, and the associated measurements from sensor 1, one computes the (partial) update $\hat{x}^{t,1}(k|k)$ and its covariance $P^{t,1}(k|k)$ according to the standard PDAF.

Step 5: Measurement validation for sensor 2.

The “predicted” measurements for sensor 2 are

$$\hat{z}^{t,2}(k) \triangleq h^2[\hat{x}^{t,1}(k|k)] \quad t = 1, \dots, r \quad (4.5.4-11)$$

and their covariances

$$S^{t,2}(k) = H^2(k)P^{t,1}(k|k)H^2(k)' + R^2 \quad t = 1, \dots, r \quad (4.5.4-12)$$

where H^2 is given in (4.5.3-4) and R^2 in (4.5.2-10).

The validation volume is

$$V^2(k) \triangleq V^{t_2,2}(k) \triangleq \max_t [\gamma\pi|S^{t,2}(k)|^{1/2}] \quad (4.5.4-13)$$

4.5.4 Integration of the IMM and the Multisensor PDAF

As before, the validation region (same for all models) is

$$[z^2(k) - \hat{z}^{t_2,2}(k)][S^{t_2,2}(k)]^{-1}[z^2(k) - \hat{z}^{t_2,2}(k)]' < \gamma \quad (4.5.4-14)$$

Step 6: Update with sensor 2 measurements in each filter $t = 1, \dots, r$.

The association probabilities are computed using $\hat{z}^{t,2}(k)$, $S^{t,2}(k)$, and the volume (4.5.4-13).

Starting from $\hat{x}^{t,1}(k|k)$ and $P^{t,1}(k|k)$, one computes the (final) update $\hat{x}^t(k|k)$ and $P^t(k|k)$ with a standard PDAF.

Step 7: Updating of the mode probabilities.

The mode probabilities are updated according to

$$\mu_t(k) = \frac{1}{c} \Lambda_t(k) \sum_{s=1}^r p_{st} \mu_s(k-1) \quad t = 1, \dots, r \quad (4.5.4-15)$$

The set of validated measurements for sensor i , $i = 1, 2$, is denoted as $Z^i(k)$ and their number is $m^i(k)$. With two sensors in the presence of clutter, the likelihood function for the sequential filter is the joint probability density function of the validated innovations from both sensors:

$$\begin{aligned} \Lambda_t(k) &\triangleq p[Z^1(k), Z^2(k)|M_t(k), m^1(k), m^2(k), Z^{1,k-1}, Z^{2,k-1}] \\ &= p[Z^2(k)|M_t(k), m^2(k), Z^{1,k}, Z^{2,k-1}] p[Z^1(k)|M_t(k), m^1(k), Z^{1,k-1}, Z^{2,k-1}] \\ &\triangleq \Lambda_t^2(k) \Lambda_t^1(k) \end{aligned} \quad (4.5.4-16)$$

where, with Z^{k-1} denoting the entire past data,

$$\begin{aligned} \Lambda_t^1(k) &\triangleq p[Z^1(k)|M_t(k), Z^{k-1}] = p[\nu_1^{t,1}(k), \dots, \nu_{m^1(k)}^{t,1}(k)|M_t(k), m^1(k), Z^{k-1}] \\ &= V^1(k)^{-m^1(k)} \gamma_0[m^1(k)] + V^1(k)^{-m^1(k)+1} \sum_{j=1}^{m^1(k)} P_G^{-1} \mathcal{N}[\nu_j^{t,1}(k); 0, S^{t,1}(k)] \gamma_j[m^1(k)] \end{aligned} \quad (4.5.4-17)$$

$$\begin{aligned} \Lambda_t^2(k) &\triangleq p[Z^2(k)|M_t(k), Z^{1,k}, Z^{2,k-1}] \\ &= p[\nu_1^{t,2}(k), \dots, \nu_{m^2(k)}^{t,2}(k)|M_t(k), m^2(k), Z^{1,k}, Z^{2,k-1}] \\ &= V^2(k)^{-m^2(k)} \gamma_0[m^2(k)] + V^2(k)^{-m^2(k)+1} \sum_{j=1}^{m^2(k)} P_G^{-1} \mathcal{N}[\nu_j^{t,2}(k); 0, S^{t,2}(k)] \gamma_j[m^2(k)] \end{aligned} \quad (4.5.4-18)$$

4.5.4 Integration of the IMM and the Multisensor PDAF

The terms γ_j are given, for the nonparametric PDAF, by

$$\gamma_j[m(k)] = \begin{cases} \frac{1}{m(k)} P_D P_G & j = 1, \dots, m(k) \\ 1 - P_D P_G & j = 0 \end{cases} \quad (4.5.4-19)$$

Step 8: Combination of the mode-conditioned estimates.

The combination of the estimates and covariances is done as follows:

$$\hat{x}(k|k) = \sum_{t=1}^r \hat{x}^t(k|k) \mu_t(k) \quad (4.5.4-20)$$

$$P(k|k) = \sum_{t=1}^r \mu_t(k) \{ P^t(k|k) + [\hat{x}^t(k|k) - \hat{x}(k|k)][\hat{x}^t(k|k) - \hat{x}(k|k)]' \} \quad (4.5.4-21)$$

4.5.5 An Example with Sea Skimmers

The following scenarios were considered:

Case 1

The target starts at location [21689 10840 40] in Cartesian coordinates in meters. The initial velocity (in m/s) is $[-8.3 \ -399.9 \ 0]$ and the target stays at constant altitude with a constant speed of $400m/s$. Its trajectory is:

- constant velocity between 0 and 20s
- a coordinated turn with acceleration of $60m/s^2$ between 20 and 35s
- constant velocity between 35 and 55s
- a coordinated turn with acceleration of $40m/s^2$ between 55 and 70s
- constant velocity between 70 and 90s

The end of the trajectory of the target, shown in Figure 4.5.5-1, is right above the origin of coordinates, where the sensors are located, at time 90s.

Case 2

In this case the target starts at [11097.6 3425 40] in Cartesian coordinates. The initial velocity is $[-6.2 \ 299.9 \ 0]$ and the target keeps a constant speed of $300m/s$. Its trajectory is:

- constant velocity between 0 and 10s
- a coordinated turn with acceleration of $50m/s^2$ between 10 and 25s
- constant velocity between 25 and 35s
- a coordinated turn with acceleration of $35m/s^2$ between 35 and 50s
- constant velocity between 50 and 60s.

The end of the trajectory of the target, shown in Figure 4.5.5-2, is right above the origin of coordinates, at time 60s.

These trajectories can be considered generic for sea skimmers or aircraft operating in an evasive mode. A zero-mean Gaussian noise of 2.5 meters standard deviation, which simulates turbulence, is added to each position coordinate of the target in both cases.

4.5.5 An Example with Sea Skimmers

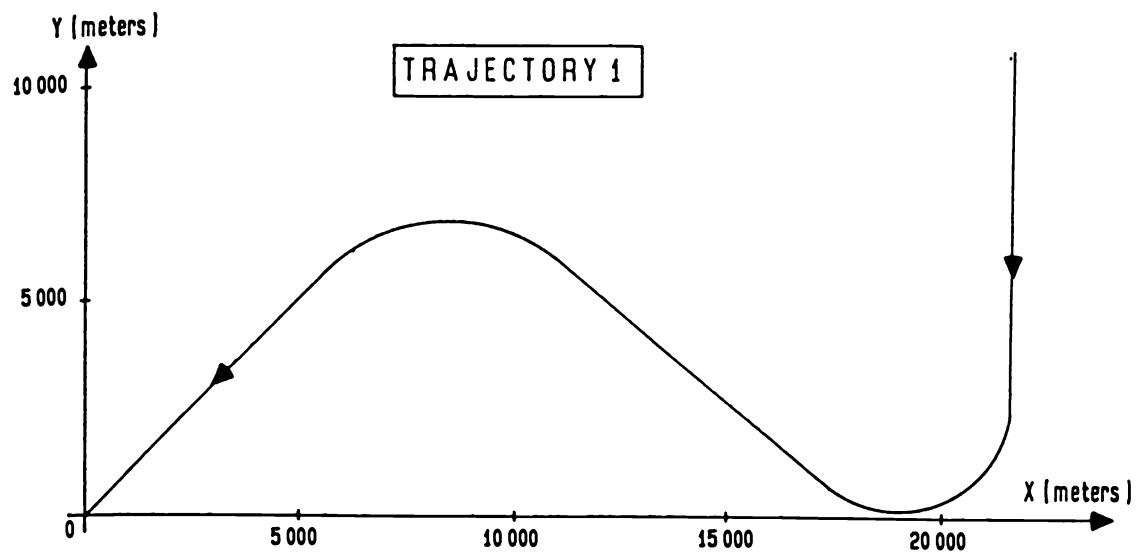


Figure 4.5.5-1: Trajectory for Case 1.

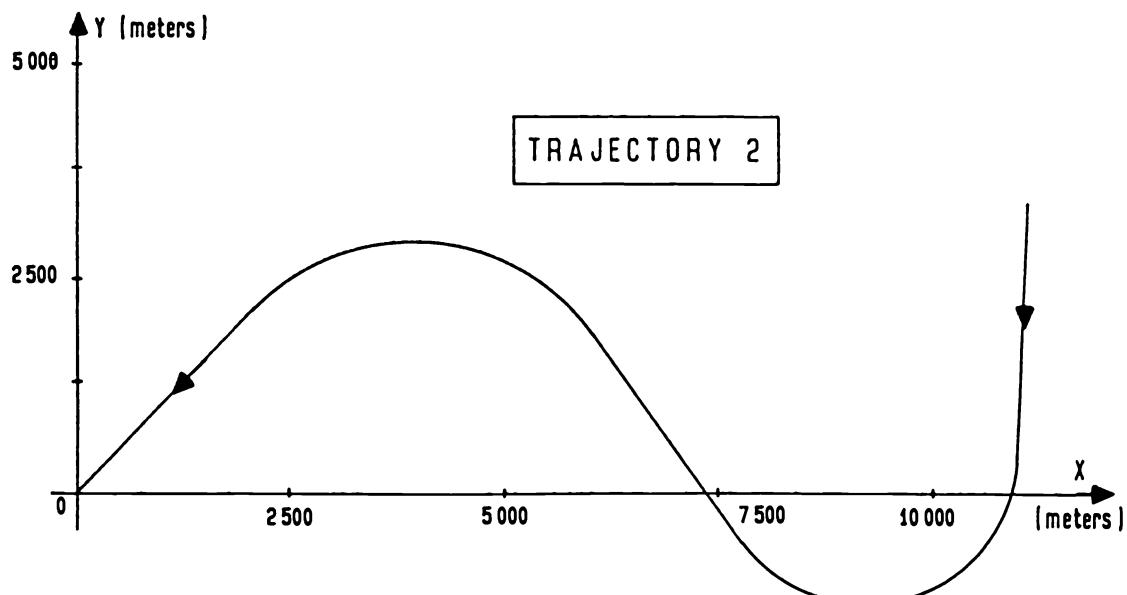


Figure 4.5.5-2: Trajectory for Case 2.

The Sensors

The measurement noises are zero-mean Gaussian with standard deviations

- radar (sensor 1): $20m$ for range and 7 mrad for azimuth;
- infrared sensor (sensor 2): 2 mrad for azimuth and elevation.

Both sensors are assumed located at the coordinate system origin.

The sampling time was $T = 1\text{ s}$ and it was assumed that $P_D = 1$ and $P_{FA} > 0$.

The IMM/MSPDAF design parameters

- process noise intensities
- mode switching probabilities

will be discussed in detail for the following two versions

- with two models, designated as IMM2
- with three models, designated as IMM3.

The IMM2 Configuration

The first model, M_1 , is a second order kinematic model with white noise acceleration (Model 1 from Subsection 4.5.2) with a standard deviation of $5m/s^2$.

The second model, M_2 , is a third order kinematic model with white noise acceleration increments (called Model 2 in Subsection 4.5.2) with a standard deviation of $20m/s^2$.

The assumed mode transition probabilities are given in Table 4.5.5-1.

	M_1	M_2
M_1	0.8	0.2
M_2	0.1	0.9

Table 4.5.5-1: Mode switching probabilities for IMM2.

The IMM3 Configuration

The first model, M_1 , is the same as in IMM2.

The second model, M_2 , and the third model, M_3 , are third order kinematic models with white noise acceleration increments (Model 2 from Subsection 4.5.2) with different variances. Model M_2 has process noise with a standard deviation of $7.5m/s^2$, and Model M_3 has process noise with a standard deviation of $40m/s^2$.

The assumed mode transition probabilities (Markov chain transition probabilities) are given in Table 4.5.5-2.

	M_1	M_2	M_3
M_1	0.8	0.0	0.2
M_2	0.0	0.8	0.2
M_3	0.3	0.3	0.4

Table 4.5.5-2: Model switching probabilities for IMM3.

Rationale for the Choice of the Models

Models M_1 in both configurations are Model 1 (from Subsection 4.5.2), suitable for the constant velocity portions of the targets, while accounting for the turbulence.

Models M_2 and M_3 in IMM3 are Model 2 (from Subsection 4.5.2) with low and high process noise, to be designated as 2L and 2H, respectively.

Model 2L is suitable for the constant acceleration portions of the trajectory, while 2H is for the onset and termination of the turns.

Model M_2 in IMM2 is Model 2 with a “compromise” intermediate noise, and will be designated as 2I.

The simulations were done for IMM/MSPDAFs (IMM2 and IMM3) as well as single model based filters using Models 1, 2L, 2I, and 2H.

The Clutter

The clutter was Poisson distributed with expected number of:

$$\lambda_1 = 13 \cdot 10^{-6} / m \cdot mrad \text{ for sensor 1 and } \lambda_2 = 7 \cdot 10^{-4} / mrad^2 \text{ for sensor 2 in Case 1,}$$

$$\lambda_1 = 6 \cdot 10^{-6} / m \cdot mrad \text{ for sensor 1 and } \lambda_2 = 3.5 \cdot 10^{-4} / mrad^2 \text{ for sensor 2 in Case 2.}$$

The Results

The results are from 20 Monte Carlo runs, for each of the two cases, for the IMM2, IMM3, as well as Model 2L, Model 2I and Model 2H alone.

Figures 4.5.5-3 and 4.5.5-9 give the RMS position error (in 3 dimensions) for the two IMM configurations for Cases 1 and 2, respectively, while Figures 4.5.5-4 and 4.5.5-10 give the corresponding RMS velocity errors. The main feature of the IMM is that it yields small errors over the trajectory portions with no acceleration. The errors shoot up (unavoidably) at the beginning and end of a maneuver, but they settle quite rapidly.

Figures 4.5.5-5 and 4.5.5-11 give the average probability of M_1 for IMM2 in Cases 1 and 2, respectively. Figures 4.5.5-6 and 4.5.5-12 give the average mode probabilities for IMM3 in the two cases. These figures show that the non-maneuvering model dominates over the periods with no acceleration both in IMM2 and IMM3.

Figures 4.5.5-7 and 4.5.5-13 give the RMS position error for the three third order kinematic models, called Model 2L (process noise with standard deviation $7.5m/s^2$), Model 2I (process noise with standard deviation $20m/s^2$), Model 2H (process noise with standard deviation $40m/s^2$) for the two cases. Figures 4.5.5-8 and 4.5.5-14 give the corresponding velocity errors. Among the single-model based filters, the one using Model 2I performed best, but still not as well as IMM3.

The overall performance is summarized in Table 4.5.5-3.

4.5.5 An Example with Sea Skimmers

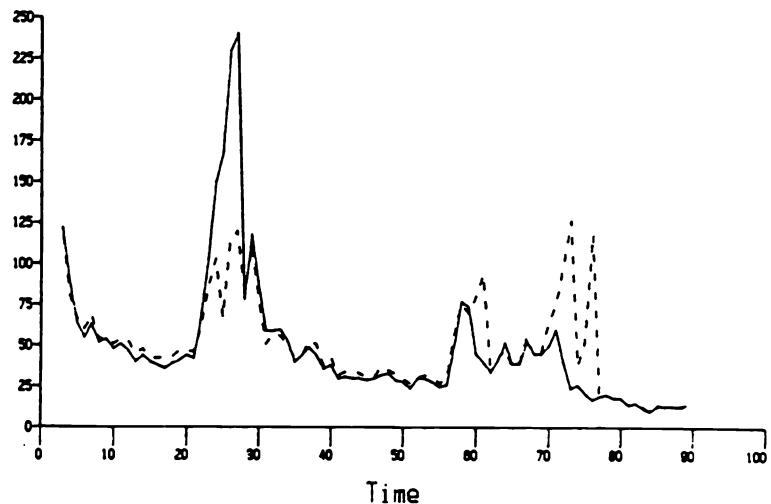


Figure 4.5.5-3: RMS position error for IMM in Case 1: — IMM2 – IMM3.

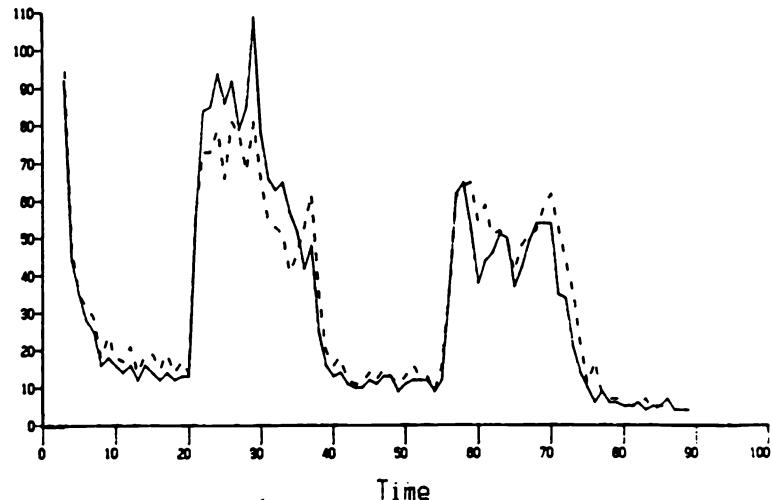


Figure 4.5.5-4: RMS velocity error for IMM in Case 1: — IMM2; – IMM3.

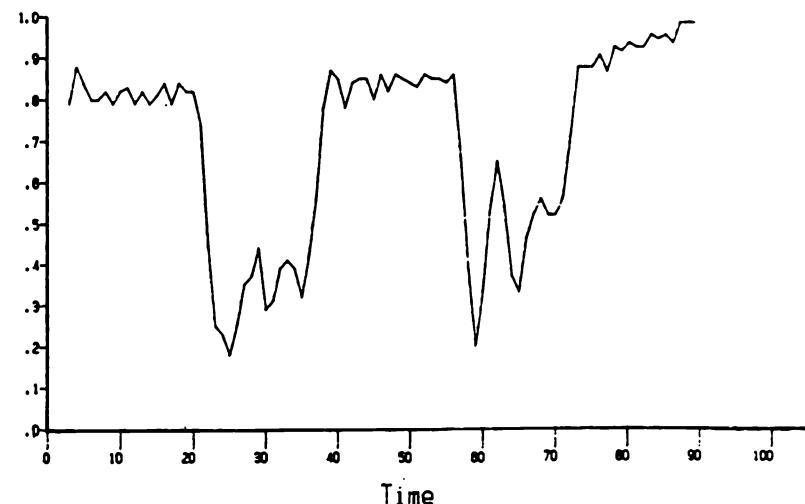


Figure 4.5.5-5: Model 1 probability in IMM2 for Case 1.

4.5.5 An Example with Sea Skimmers

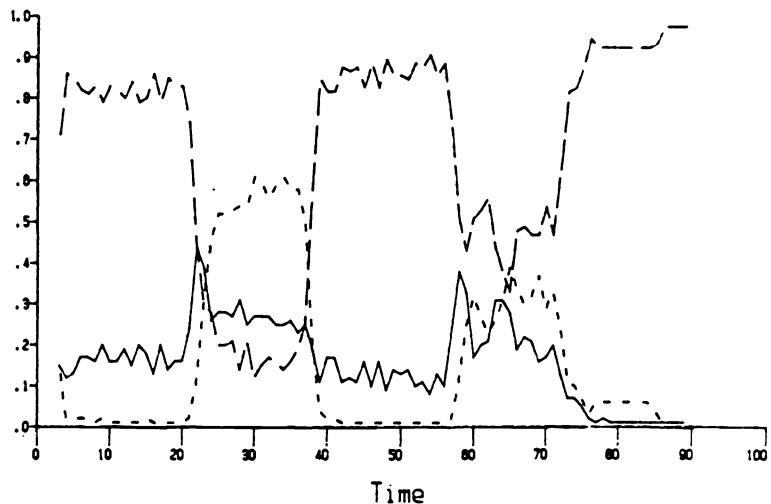


Figure 4.5.5-6: Model probabilities in IMM3 for Case 1: $-- \mu_1$; $- - \mu_2$; $- \mu_3$.

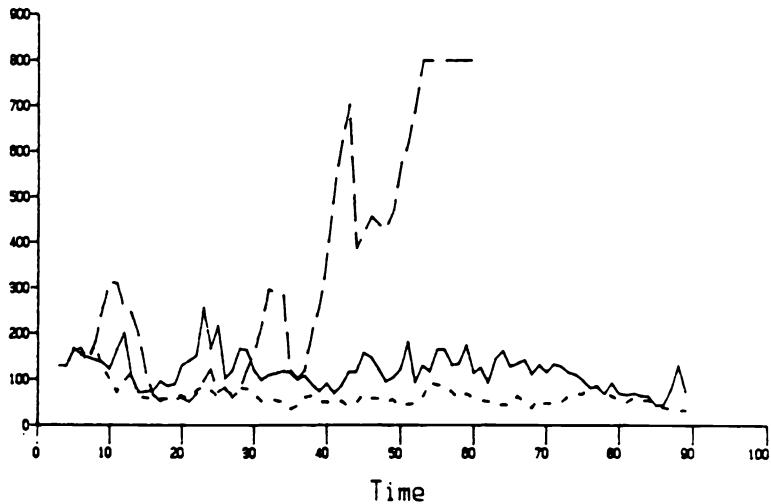


Figure 4.5.5-7: RMS position error for Model 2 in Case 1: $-- 2L$; $- - 2I$; $- 2H$.

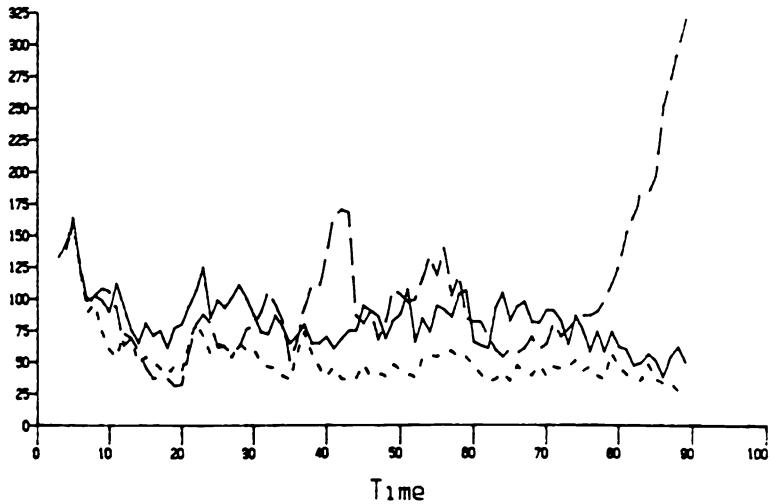


Figure 4.5.5-8: RMS velocity error for Model 2 in Case 1: $-- 2L$; $- - 2I$; $- 2H$.

4.5.5 An Example with Sea Skimmers

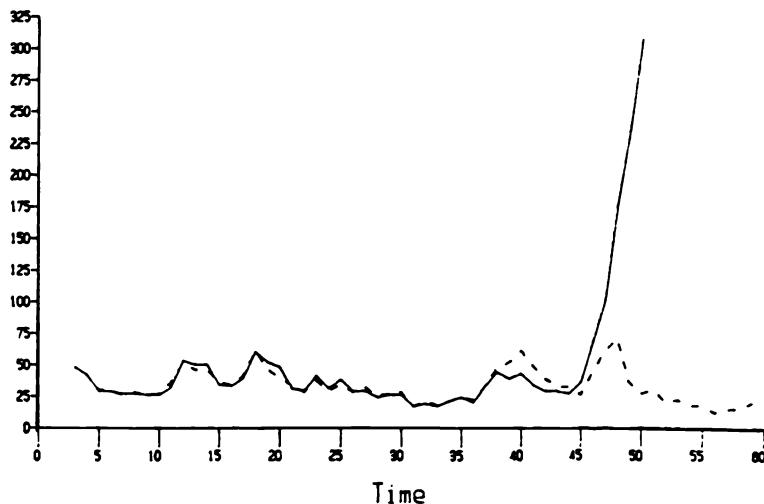


Figure 4.5.5-9: RMS position error for IMM in Case 2; — IMM2; - - IMM3.

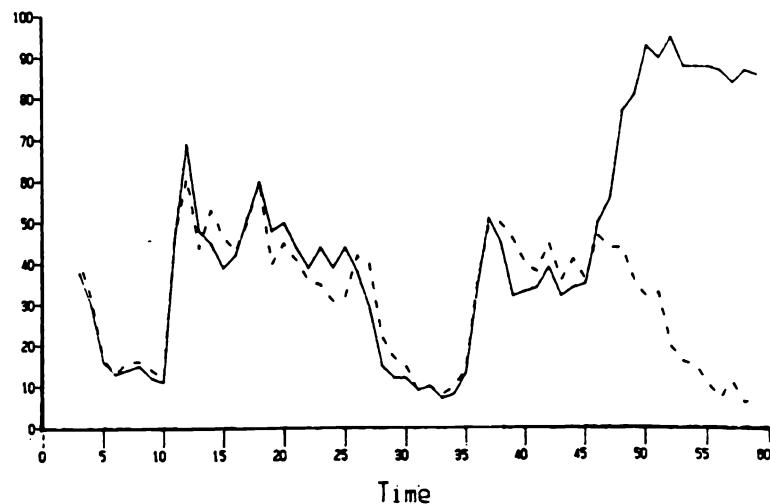


Figure 4.5.5-10: RMS velocity error for IMM in Case 2; — IMM2; - - IMM3.

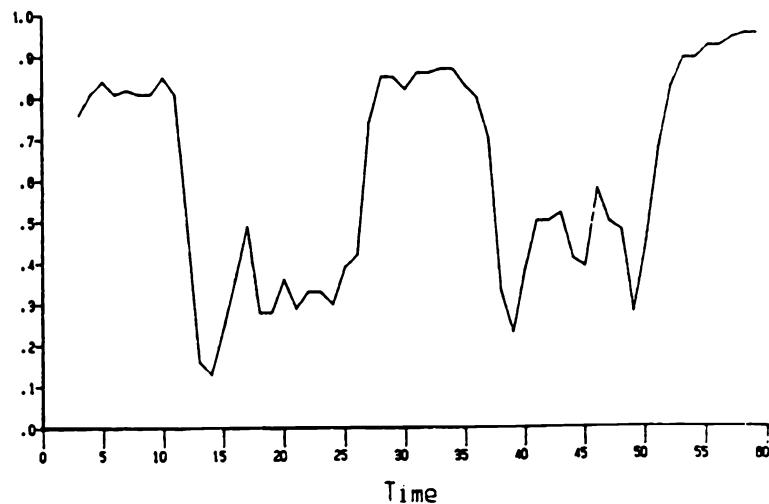


Figure 4.5.5-11: Model 1 probability in IMM2 for Case 2.

4.5.5 An Example with Sea Skimmers

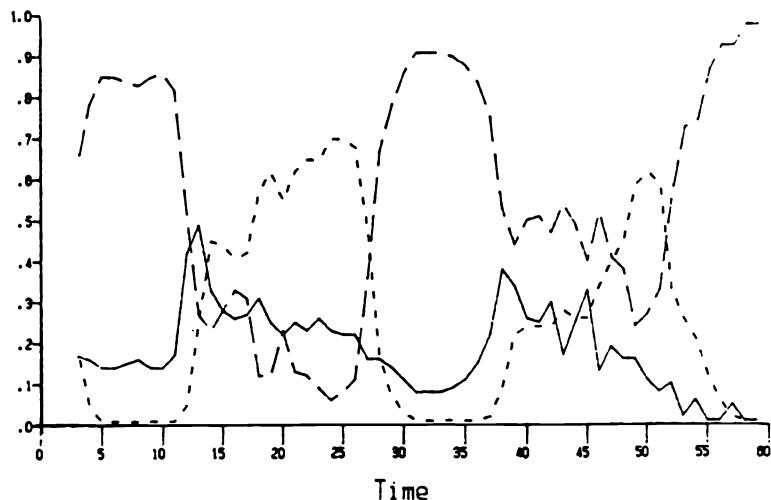


Figure 4.5.5-12: Model probabilities in IMM3 for Case 2: $-- \mu_1$; $- - \mu_2$; $- \mu_3$.

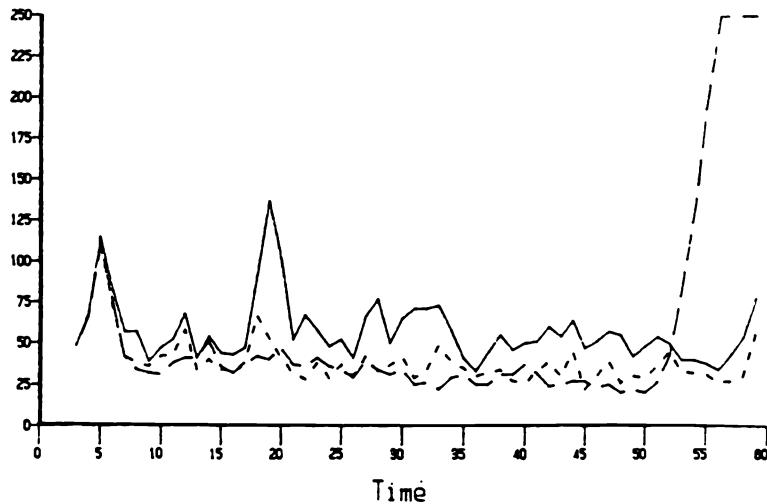


Figure 4.5.5-13: RMS position error for Model 2 in Case 2: $-- 2L$; $- - 2I$; $- 2H$.

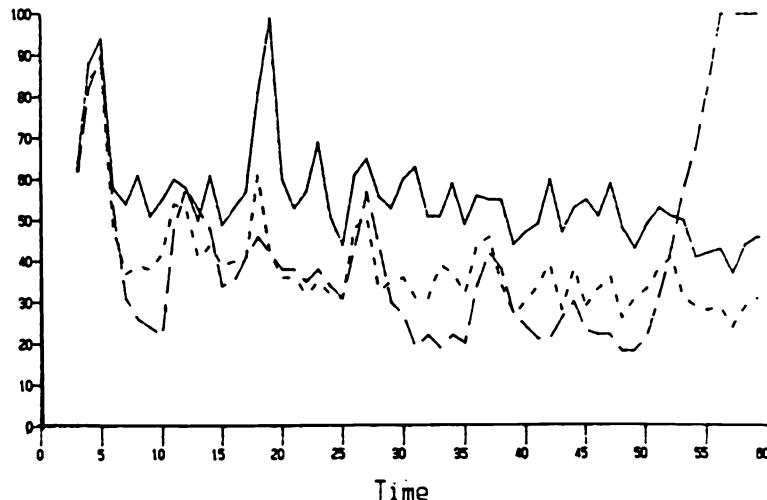


Figure 4.5.5-14: RMS velocity error for Model 2 in Case 2: $-- 2L$; $- - 2I$; $- 2H$.

Overall Performance

Estimator	Percent loss	RMS position error (m)	RMS velocity error (m/s)
Model 1	100		
Model 2L	20		
Model 2I	0	100/50	60/50
Model 2H	0	200/100	100/75
IMM2	5	75/20	60/5
IMM3	0	125/20	60/5

Table 4.5.5-3: Track loss percentage and estimation accuracy (peak error in second turn / average error during the last leg of the trajectory in Case 1).

The Effective Clutter Density

The average number of false alarms in the validation gate $\gamma = 16$ are given in Table 4.5.5-4. The clutter, which is particularly heavy in the IR sensor, precludes the use of a “nearest neighbor filter” (Section 3.2).

	Case 1		Case 2	
	Sensor 1	Sensor 2	Sensor 1	Sensor 2
Model 2L	0.37	1.33	0.15	1.02
Model 2I	0.55	2.57	0.29	2.18
Model 2H	1.18	4.61	0.59	3.70
IMM2	0.3	1.14	0.20	1.25
IMM3	0.5	2.28	0.27	2.07

Table 4.5.5-4: Average number of false alarms in the validation region.

Among the single-model based filters, the one using Model 2I was the best compromise between keeping the target in track and yielding good estimation errors; however, IMM3 yielded about the same peak errors but significantly lower errors during the straight leg, especially velocity errors.

4.5.6 The IMM/MSPDAF — Summary

4.5.6 The IMM/MSPDAF — Summary

The IMM/MSPDAF algorithm can track reliably a highly maneuvering target using measurements from diverse sensors in the presence of clutter.

This algorithm is based on

- the PDAF for the measurement association and state estimation
- sequential updating for the multisensor integration
- the IMM for efficiently handling the multiple models needed to describe the behavior of a highly maneuvering target.

The models used in the IMM3 version were:

- nearly constant velocity motion
- nearly constant acceleration motion
- motion with large acceleration increments (for the onset and termination of maneuvers).

The algorithm design parameters are:

- the Markov chain transition matrix for switching between the models
- the process noise levels for the various models.

The target detection probability was assumed known but no knowledge of the environmental parameter that describes the false alarm rate is needed.

The gain in performance vs. the best single-model based filter is significant especially in the evolution of the velocity error.

The very modest computational requirements of the IMM/PDAF combination make it an excellent candidate for real-time implementation.

4.6 INTERACTING MULTIPLE MODEL TRACKING WITH TARGET AMPLITUDE FEATURE

4.6.1 Introduction

Assumptions

- In addition to measurements of the target kinematic variables, the amplitudes of the signals (following their processing at the receiver) that exceed some threshold are available for the tracker — i.e., the detection process is *more than binary*
- Probabilistic models of this amplitude for target and noise (clutter) origins are available.

A recursive algorithm is presented which utilizes the **measured amplitude** of the “returns” — the **amplitude information (AI)** to carry out

- **track formation** and
- **track maintenance** through target maneuvers

in a cluttered environment, i.e., especially for **low observable** (low SNR) targets.

The technique, called **IMMPDAFAI**, combines

- the IMM with
- a generalized PDA which utilizes the **amplitude information** as the measured **target feature**, with suitable probabilistic models for the target and clutter return amplitudes.

The IMMPDAFAI is illustrated in the context of an active sensor (an active sonar) — the results are directly applicable to radar.

This technique can be applied also to a passive sensor — the key is to have the appropriate *probabilistic models for the feature*.

The IMMPDAFAI utilizes:

- the *consistent strength* (regularity in a probabilistic sense) of the target returns provided by the single observation detector, and
- *consistency in target motion* (again in a probabilistic sense) to detect a target track with a small number of measurements and discriminate against false tracks.

and provides a systematic way to obtain the *probability that there is a target in track*.

The amplitude feature is a very useful measurement that can greatly increase the reliability of track formation and maintenance.

4.6.1 Introduction

The IMMPDAFAI for Track Formation

For this function, the IMMPDAFAI uses two target modes (“observable” and “no target”) and attempts to satisfy the following conflicting criteria:

- a) maximize the probability of detecting a target (target track acceptance), P_{DT}
- b) minimize the probability of false track acceptance, P_{FT} , i.e., declaring a target track present based upon noise-only observations or clutter
- c) minimize the average number of measurements required to declare a target is present, i.e., *rapid target detection*
- d) minimize the duration and number of false tracks which must be maintained; this is directly related to the computational load.

The IMMPDAFAI for Track Maintenance

A means of maintaining track for a detected target through maneuvers is obtained by augmenting the mode set with suitable target maneuver models.

4.6.2 Use of Amplitude Information in the PDAF

4.6.2 Use of Amplitude Information in the PDAF

The measurement vector consists of kinematic components augmented with the amplitude

$$z^a(k) = \begin{bmatrix} z(k) \\ a(k) \end{bmatrix} \quad (4.6.2-1)$$

The set of validated measurements at time k is

$$Z(k) = \{z_i^a(k)\}_{i=1}^{m(k)} \quad (4.6.2-2)$$

The measurement validation consists of the following:

- (i) the kinematic part has to lie within the standard elliptical gate with threshold γ , and
- (ii) the amplitude has to satisfy the requirement

$$a_i(k) > \tau \quad (4.6.2-3)$$

where τ is the detection threshold, to be discussed in the sequel.

In the standard PDAF the probability of each measurement being correct utilizes only its location within the gate — it is based only on what now is called the kinematic component of the augmented measurement vector. The amplitude is used in a binary manner: only if it exceeds a threshold, then the (kinematic) measurement is retained.

The *quantitative use of the amplitude* in the evaluation of the association probabilities will be described next.

Modeling of the Amplitude Feature

The following probabilistic models are assumed to be available:

$p_0(a)$ = pdf of the amplitude if it is due to noise only

$p_1(a)$ = pdf of the amplitude if it originated from a target,

with the latter parameterized by the **expected SNR** of the target returns, d .

The measurement amplitude will be modelled as Rayleigh distributed:

$$p_0(a) = ae^{-\frac{a^2}{2}} \quad a \geq 0 \quad (4.6.2-4)$$

$$p_1(a) = \frac{a}{1+d} e^{-\frac{a^2}{2(1+d)}} \quad a \geq 0 \quad (4.6.2-5)$$

for noise only (normalized) and target, respectively.

4.6.2 Use of Amplitude Information in the PDAF

When the full signal-processed data set is thresholded to provide a set of measurements for tracking, the **target detection probability** is

$$P_D = \int_{\tau}^{\infty} p_1(a) da \quad (4.6.2-6)$$

and the **false alarm probability** is

$$P_{FA} = \int_{\tau}^{\infty} p_0(a) da \quad (4.6.2-7)$$

The corresponding pdfs of the amplitude of a measurement, which by the validation requirement must have exceeded the threshold, are

$$p_0^{\tau}(a) = \frac{1}{P_{FA}} p_0(a) \quad a \geq \tau \quad (4.6.2-8)$$

$$p_1^{\tau}(a) = \frac{1}{P_D} p_1(a) \quad a \geq \tau \quad (4.6.2-9)$$

The **target to clutter amplitude likelihood ratio**

$$\boxed{L_{T/C}[a_i(k)] \triangleq \frac{p_1^{\tau}[a_i(k)]}{p_0^{\tau}[a_i(k)]}} \quad (4.6.2-10)$$

modifies the standard PDA (cf. Subsection 3.4.7).

The association probabilities are the same as in the standard PDA

$$\beta_i(k) = \begin{cases} \frac{e_i}{b + \sum_{j=1}^{m(k)} e_j} & i = 1, \dots, m(k) \\ \frac{b}{b + \sum_{j=1}^{m(k)} e_j} & i = 0 \end{cases} \quad (4.6.2-11)$$

with the modified term that incorporates the amplitude information

$$\boxed{e_i \triangleq e^{-\frac{1}{2} \nu_i(k)' S(k)^{-1} \nu_i(k)} L_{T/C}[a_i(k)]} \quad (4.6.2-12)$$

and, for the nonparametric PDA

$$b \triangleq \left(\frac{2\pi}{\gamma} \right)^{\frac{n_x}{2}} m(k) c_{n_x}^{-1} \frac{1 - P_D P_G}{P_D} \quad (4.6.2-13)$$

4.6.3 The IMMPDAFAI

4.6.3 The IMMPDAFAI

The **IMMPDAFAI** estimator consists of the IMM configuration with

- PDAFAI estimation modules matched to each target mode;
- probability evaluator for each mode, which uses the mode likelihood function that incorporates the amplitude information, detailed below, and the mode transition probabilities;
- estimate mixer at the filter inputs;
- estimate combiner at the output of the filters.

The IMM steps are as in Subsection 1.5.4 and the PDAF steps are described in Subsection 3.4.2; the association probabilities are to be computed according to (4.6.2-11) with the amplitude information.

The Markov chain that models the transitions among the target modes (model switchings) has transition probabilities

$$p_{st} = P\{M_t(k)|M_s(k-1)\} \quad (4.6.3-1)$$

which are estimator design parameters.

The Mode Likelihood Functions with Amplitude Information

The likelihood function of mode t at time k is the joint probability density of the validated innovations from time k and can be shown to be given by

$$\Lambda_t(k) = \left(b + \sum_{j=1}^{m(k)} e_j \right) \frac{P_D V^{-m(k)}}{m(k)} \left(\frac{\gamma}{2\pi} \right)^{\frac{n_z}{2}} c_{n_z} \prod_{j=1}^{m(k)} p_0(a_j) \quad (4.6.3-2)$$

where b and e_j are given in (4.6.2-12) and (4.6.2-13), respectively.

For $P_D = 0$, i.e., for the “no target” mode, the above becomes

$$\Lambda_t(k) = V^{-m(k)} \prod_{j=1}^{m(k)} p_0(a_j) \quad (4.6.3-3)$$

4.6.4 Target Motion Models

4.6.4 Target Motion Models

The use of multiple models to perform both the track formation and the track maintenance functions is accomplished by using two models in the track formation stage, which are then augmented with additional models to provide track maintenance through target maneuvers after a target has been detected.

The two model approach for track formation assumes that target detection will be performed for non-maneuvering targets. Thus the two models used are:

- (i) a nearly constant velocity “observable” target model,
- (ii) a “no target model” with similar motion.

There are five additional candidate models to handle target maneuvers which can be combined with the two models from track formation.

The “no target” model remains to provide a means of pruning lost tracks.

The target motion models in two Cartesian coordinates with sampling period T are

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (4.6.4-1)$$

where

$$v(k) = \begin{bmatrix} v_\xi(k) \\ v_\eta(k) \end{bmatrix} \quad (4.6.4-2)$$

are zero-mean white noise sequences in the ξ and η coordinates used to model uncertain accelerations.

MODELS 1, 2: White Noise Acceleration Models

M^1 is a second order kinematic model (with position and velocity components) in each of the two Cartesian coordinates with state transition matrix

$$F^1 = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6.4-3)$$

The variations in velocity are modelled as piecewise constant zero-mean white noise accelerations. Thus the noise gain is

$$\Gamma^1 = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \quad (4.6.4-4)$$

The noise variances are

$$\sigma_{v_\xi}^2 = \sigma_{v_\eta}^2 = q_1 \quad (4.6.4-5)$$

A “small” q_1 yields a “nearly constant velocity” motion.

Model M^1 is applicable for tracking targets moving at very close to constant course and speed and, when used in conjunction with M^0 , will provide a means of performing **target track detection or track formation**.

M^2 has the same state equation as M^1 with higher level of process noise

$$\sigma_{v_\xi}^2 = \sigma_{v_\eta}^2 = q_2 \quad (4.6.4-6)$$

Model M^2 , using a higher level of white noise acceleration than M^1 , allows for target acceleration and is applicable (with a limited degree of success) to tracking maneuvering targets.

MODEL 0: The “No Target” Model

M^0 has the same state equations and noise assumptions as M^1 , but with $P_D = 0$.

4.6.4 Target Motion Models

MODELS 3, 4: Wiener Process Acceleration Models

M^3 is a third order model (with position, velocity and acceleration components) in each of the two Cartesian coordinates with state transition matrix

$$F^3 = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6.4-7)$$

The acceleration increments are a discrete-time zero-mean white noise. Thus

$$\Gamma^3 = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 1 & 0 \\ 0 & T^2/2 \\ 0 & T \\ 0 & 1 \end{bmatrix} \quad (4.6.4-8)$$

With σ_{v_ξ} , σ_{v_η} denoting the white process noise standard deviations in each component — RMS *acceleration increments over a sampling period*, the design parameter for this model is

$$\sigma_{v_\xi}^2 = \sigma_{v_\eta}^2 = q_3 \quad (4.6.4-9)$$

A “small” q_3 yields a “nearly constant acceleration” motion.

M^4 has the same state equation as M^3 with higher level of process noise

$$\sigma_{v_\xi}^2 = \sigma_{v_\eta}^2 = q_4 \quad (4.6.4-10)$$

Models M^3 and M^4 include acceleration estimation with both coordinates having equivalent levels of process noise.

A maneuver onset is a rapid jump to a non-zero target acceleration from zero and then a jump back to zero acceleration at termination -- this is matched to M_4 .

The lower level noise model M_3 provides more accurate estimation during a maneuver (nearly constant acceleration).

4.6.4 Target Motion Models

MODELS 5, 6: Coordinated Turn Models

If the target is moving with **constant speed** (the magnitude of the velocity vector) and turning with a **constant angular rate**, i.e., executing a **coordinated turn**, the equations of motion in the plane (ξ, η) are

$$\ddot{\xi} = -\Omega\dot{\eta} \quad \ddot{\eta} = \Omega\dot{\xi} \quad (4.6.4-11)$$

where Ω is the constant angular rate.

M^5 will have $\Omega > 0$ for a counterclockwise (left) turn and M^6 will have $\Omega < 0$ for a clockwise (right) turn.

The state space representation of the above with the state vector

$$x \triangleq [\xi \ \dot{\xi} \ \eta \ \dot{\eta} \ \Omega]' \quad (4.6.4-12)$$

has, for sampling period T , the following transition matrix

$$F^5 = F^6 = \begin{bmatrix} 1 & \frac{\sin \Omega T}{\Omega} & 0 & -\frac{1-\cos \Omega T}{\Omega} & 0 \\ 0 & \cos \Omega T & 0 & -\sin \Omega T & 0 \\ 0 & \frac{1-\cos \Omega T}{\Omega} & 1 & \frac{\sin \Omega T}{\Omega} & 0 \\ 0 & \sin \Omega T & 0 & \cos \Omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6.4-13)$$

and

$$\Gamma^5 = \Gamma^6 = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \\ 0 & 0 \end{bmatrix} \quad (4.6.4-14)$$

with noise variances

$$\sigma_{v_\xi}^2 = \sigma_{v_\eta}^2 = q_5 = q_6 \quad (4.6.4-15)$$

The angular rate will be estimated as part of the state vector.

In all the above models $\sigma_{v_\xi v_\eta} = 0$.

4.6.5 The Mixing with Models of Different Dimension

In the IMM the mixing step that generates the initial conditions of each mode-matched filter at the beginning of each cycle is straightforward if the models have the same state vector.

The mixing of the state estimates of M^1 (which have dimension 4) with those of M^3 and M^4 (which have dimension 6 — they also include acceleration components) is done by setting the acceleration estimates and their variances in M^1 to zero. This ensures the compatibility of the these models.

When M^5 and M^6 are used (together with M^1), they need initialization for the turn rate. This is obtained by using the *latest* estimates from M^1 in conjunction with a discretization of the coordinated turn motion equations (4.6.4-11) to obtain an estimate of the turn rate as follows:

$$\hat{\Omega}_{\xi}^1(k|k) = -\frac{\hat{\xi}^1(k|k) - \hat{\xi}^1(k-1|k-1)}{\hat{\eta}^1(k|k)T} \quad (4.6.5-1)$$

$$\hat{\Omega}_{\eta}^1(k|k) = \frac{\hat{\eta}^1(k|k) - \hat{\eta}^1(k-1|k-1)}{\hat{\xi}^1(k|k)T} \quad (4.6.5-2)$$

Since the above estimates are quite noisy, they will be only used to “detect” a sudden change of Ω from zero. This is accomplished by using

$$\hat{\Omega}^1(k) \triangleq \max\{|\hat{\Omega}_{\xi}^1(k|k)|, |\hat{\Omega}_{\eta}^1(k|k)|\} \quad (4.6.5-3)$$

to “rapidly” initialize the filters matched to the coordinated turns via a special mixing equation.

The mixed initial estimate for the turn rate in the filters matched to M^5 (which has a positive Ω) and M^6 (which has a negative Ω) are, for cycle k , given by the *special mixing equation*

$$\begin{aligned} \hat{\Omega}^{0,5}(k-1|k-1) &= \mu_{1|5}(k-1|k-1)\hat{\Omega}^1(k-1) + \mu_{5|5}(k-1|k-1)\hat{\Omega}^5(k-1|k-1) \\ &\quad - \mu_{6|5}(k-1|k-1)\hat{\Omega}^6(k-1|k-1) \end{aligned} \quad (4.6.5-4)$$

and, with change of sign

$$\begin{aligned} \hat{\Omega}^{0,6}(k-1|k-1) &= -\mu_{1|6}(k-1|k-1)\hat{\Omega}^1(k-1) - \mu_{5|6}(k-1|k-1)\hat{\Omega}^5(k-1|k-1) \\ &\quad + \mu_{6|6}(k-1|k-1)\hat{\Omega}^6(k-1|k-1) \end{aligned} \quad (4.6.5-5)$$

respectively, where $\mu_{s|t}(k-1|k-1)$ are the mixing probabilities detailed in (1.5.4-10).

4.6.6 The Simulations

4.6.6 The Simulations

Two sets of simulations were made to evaluate

- *track formation* performance for nearly constant course and speed targets,
- the estimation accuracy and ability to *Maintain target track during a maneuver*.

The Measurements

An active sonar was assumed to provide, with period $T = 60\text{s}$, the measurements

$$z^a(k) = \begin{bmatrix} \sqrt{\xi(k)^2 + \eta(k)^2} \\ \tan^{-1}\left(\frac{\eta(k)}{\xi(k)}\right) \\ a(k) \end{bmatrix} + \begin{bmatrix} w_r(k) \\ w_\theta(k) \\ 0 \end{bmatrix} \quad (4.6.6-1)$$

where $\xi(k)$ and $\eta(k)$ are the two Cartesian position components of the state vector.

The range and bearing measurement noises w_r and w_θ are both zero-mean Gaussian random sequences with standard deviations of $\sigma_r = 20\text{m}$, and $\sigma_\theta = 0.5^\circ$, respectively.

The surveillance region is a 15° bearing sector from a range of $65,000\text{m}$ to $85,000\text{m}$ ($10,000\text{m}$ range and bearing cells of resolution 1.5° and 20m).

Noise-only (clutter) measurements were generated uniformly over the entire surveillance region at each time with Rayleigh amplitudes according to (4.6.2-8).

The target amplitude values are Rayleigh random variables parameterized by the signal to noise ratio d as in (4.6.2-5). This accounts for the possibility of a missed target measurement if the amplitude does not exceed the threshold τ , so the amplitudes of the target-originated measurements obey (4.6.2-9).

In addition to the simulations, track formation was carried out on a data set consisting of real clutter on top of which target measurements were overlayed. These results are presented in the next subsection.

4.6.7 Results for Track Formation

The target initial state is $x(0) = [50000m \ 5m/s \ 50000m \ 5m/s]'$.

The target state (position and velocity in two Cartesian coordinates) evolves (with the sampling period $T = 60s$) according to

$$x(k+1) = F^1 x(k) + \Gamma^1 v(k) \quad (4.6.7-1)$$

The process noises $v_x(k)$ and $v_y(k)$ are i.i.d. zero-mean white Gaussian with standard deviation $\sqrt{q_1} = .005m/s^2$ (RMS velocity change 6% per sampling period).

Two modes are used in track formation:

- “observable target” (true target) mode M^1 , and
- “unobservable target” (no target) mode M^0 .

In both modes the measurements can originate from the target with its P_D (determined by the SNR and the threshold τ , which is a design parameter), or from clutter. In the “unobservable target” mode $P_D = 0$.

The algorithm yields mode probabilities and this provides the **True Target Probability (TTP)** for each track under consideration, defined as

$$\text{TTP}(k) = \mu^1(k) \quad (4.6.7-2)$$

corresponding to M^1 .

The performance was evaluated in a **sliding window** operation mode:

All the measurements left unvalidated at time k are used to initiate new tracks. For these tracks, one-stage track splitting is used at time $k+1$ to form a track for each measurement located in a validation region centered on the initial measurement.

The area of the initial validation region is determined by the assumed maximum target velocity and the interval T as in (4.4.2-5).

For a new track starting at k , the IMMPDAFAI is employed from time $k+2$. Each new track starts with initial TTP = 0.5. With one or two measurements, the TTP is updated with likelihood functions based on the observed amplitudes only, i.e., with

$$\Lambda^1(k) = p_1^\tau[a(k)] \quad \Lambda^0(k) = p_0^\tau[a(k)] \quad (4.6.7-3)$$

The mode transition probabilities were taken as

$$[p_{st}] = \begin{bmatrix} .98 & .02 \\ .02 & .98 \end{bmatrix} \quad (4.6.7-4)$$

4.6.7 Results for Track Formation

The validation gate size (for the kinematic part of the measurement) is the 99.97% region ($\gamma = 16$).

Simulations were run for three SNR scenarios: 10dB, 13dB and 20dB. For each of these cases three levels of clutter (determined by the detection threshold setting) were examined representing light, medium, and heavy densities: $P_{FA} = 0.001, 0.003, 0.01$. The corresponding values of P_D and the expected number of false measurements per standard filter steady-state ($k > 10$) gate, $E(m)$, are shown in Table 4.6.7-1.

SNR	20dB		13dB		10dB		
	P_{FA}	P_D	$E(m)$	P_D	$E(m)$	P_D	$E(m)$
.001		.93	.04	.72	.05	.54	.10
.003		.94	.12	.76	.16	.59	.25
.010		.96	.50	.80	.60	.66	.90

Table 4.6.7-1: Target P_D and expected number of false measurements.

A track table of all tracks is maintained and sorted based upon the TTP for each track at each time k . The sorting ensures that *the highest probability track will have “first claim” to measurements*. Measurements associated to a track are unavailable for other tracks.

The track table size is limited by discarding low probability tracks — those with $TTP < 0.1$, at each time.

The performance measures of a track formation algorithm are:

- P_{FT} — **false track acceptance probability**
- P_{DT} — **true track acceptance probability**, i.e., the **target track detection probability**

A *decision rule for track acceptance* is needed.

The joint exceedance probability distribution of

- the TTP and
- the age of all tracks

$$P\{\text{TTP} \geq x, \text{AGE} \geq y\} \quad \text{for } x = 0.1, \dots, 0.9; y = 1, 2, \dots \quad (4.6.7-5)$$

will be used as a criterion for setting a *two dimensional threshold* for **track acceptance**.

The joint exceedance distributions are shown for low, medium, and high density clutter in Figure 4.6.7-1 for the 10dB SNR case.

4.6.7 Results for Track Formation

The probability was estimated by measuring the frequency of occurrence using multiple runs to obtain 10,000 tracks at scan 60.

As expected, the largest fraction of tracks are those with low probability of originating from a target (i.e., low TTP) and exist only for a short period of time.

Increasing the clutter density (lowering the detection threshold) causes false tracks to exist longer with a higher TTP.

The decision rule for track acceptance is based on:

- the **calculated TTP** and
- the **age of the track**.

A compromise level should be chosen that is acceptable for false tracks as well as target detection.

The boundary lines in Figure 4.6.7-1 illustrate the decision boundaries for track acceptance which result in

- $P_{FT} < 0.001$ (solid line) and
- $P_{FT} < 0.01$ (dashed line).

In these cases 0.1% (1%) of the false tracks are accepted as target tracks.

The mapping of P_{FT} to the **expected number of accepted false tracks** is discussed later.

Target track detection performance as a function of the number of scans and the SNR, is evaluated for the P_{FT} values listed above.

The probability of detecting (accepting) a target track P_{DT} vs. the number of scans is shown in Figures 4.6.7-2 through 4.6.7-5 for low, medium and high clutter, respectively. The quantity P_{DT} was estimated from 1000 independent runs where the target is present from scan 1.

Figure 4.6.7-3 shows the performance of the classical sequential detector for the low clutter case: at low SNR the IMMPDAFAI (Figure 4.6.7-2) is clearly superior.

4.6.7 Results for Track Formation

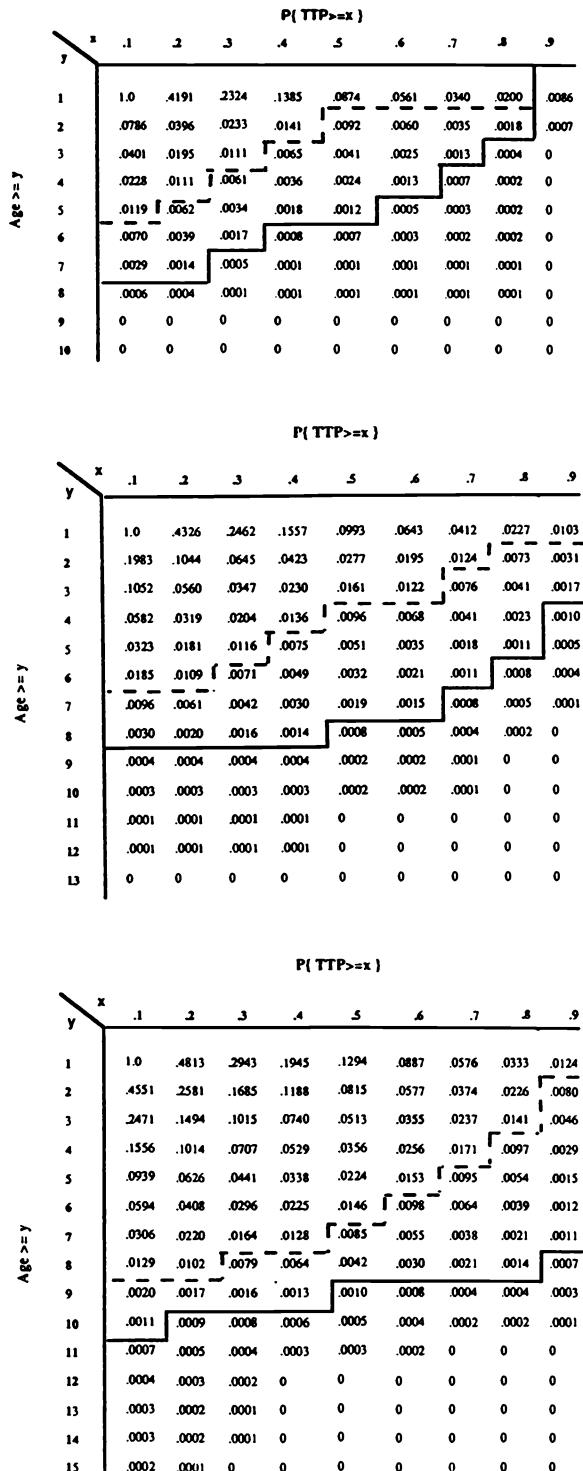


Figure 4.6.7-1: Joint exceedance distribution of TTP and track age (no target present) (SNR = 10dB, top: $P_{FA} = 0.001$, middle: $P_{FA} = 0.003$, bottom: $P_{FA} = 0.01$).

4.6.7 Results for Track Formation

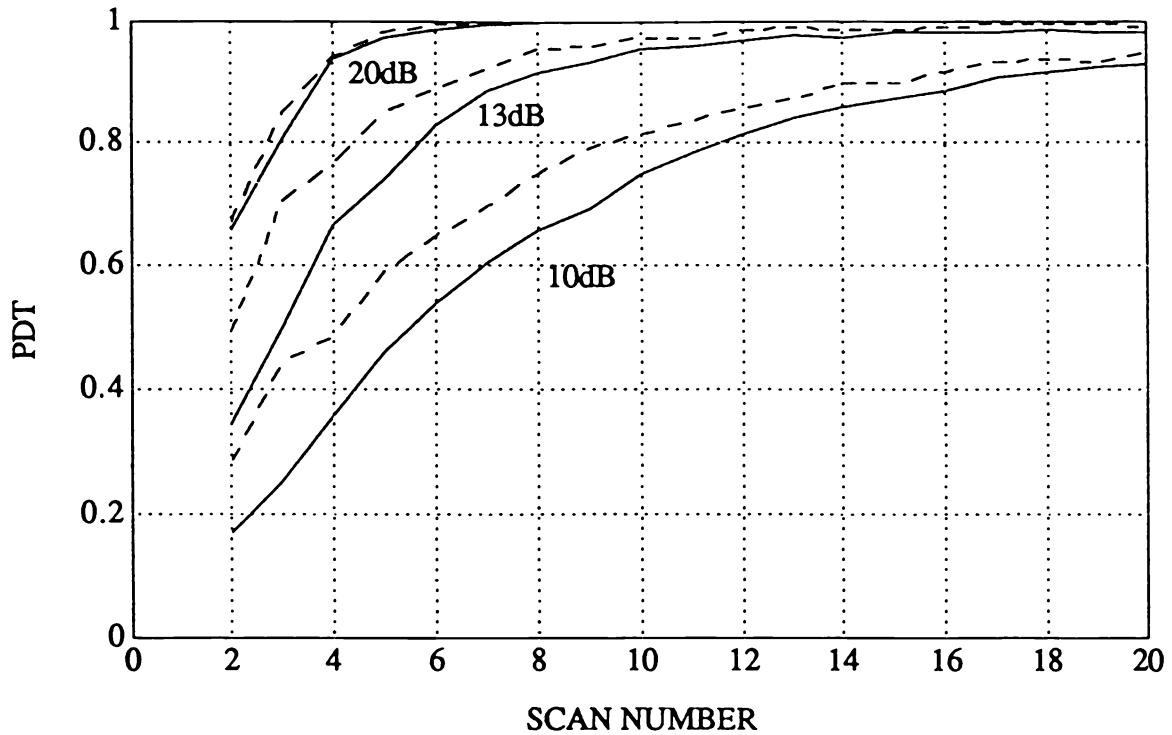


Figure 4.6.7-2: IMMPDAFAI: The probability of detecting a target track P_{DT} for $P_{FA} = 0.001$ (— $P_{FT} = 0.001$; - - $P_{FT} = 0.01$).

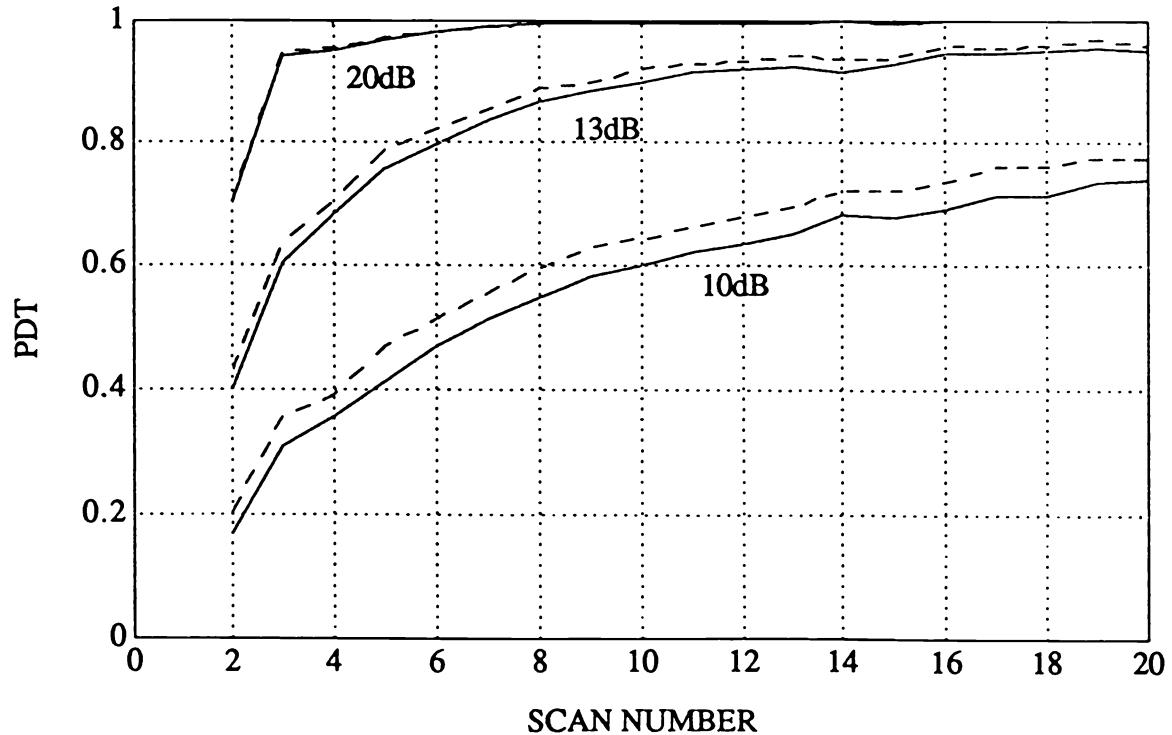


Figure 4.6.7-3: Sequential Detector: The probability of detecting a target track P_{DT} for $P_{FA} = 0.001$ (— $P_{FT} = 0.001$; - - $P_{FT} = 0.01$).

4.6.7 Results for Track Formation

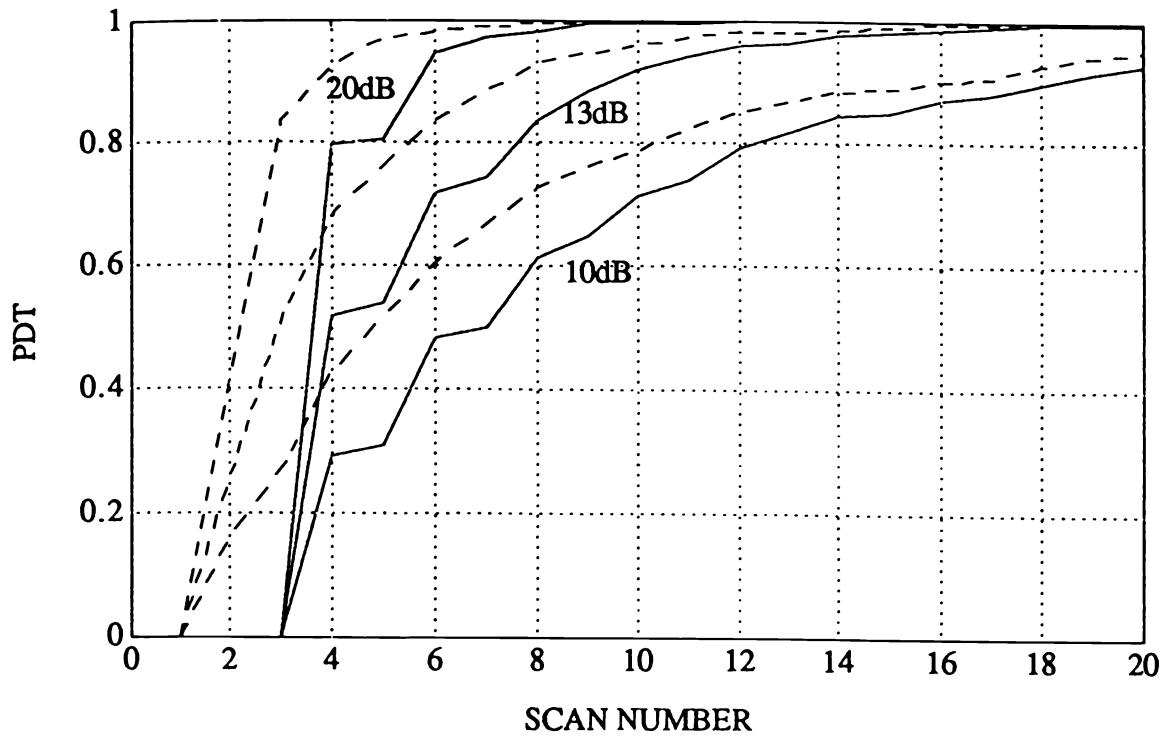


Figure 4.6.7-4: IMMPDAFAI: The probability of detecting a target track P_{DT} for $P_{FA} = 0.003$ (— $P_{FT} = 0.001$; - - $P_{FT} = 0.01$).

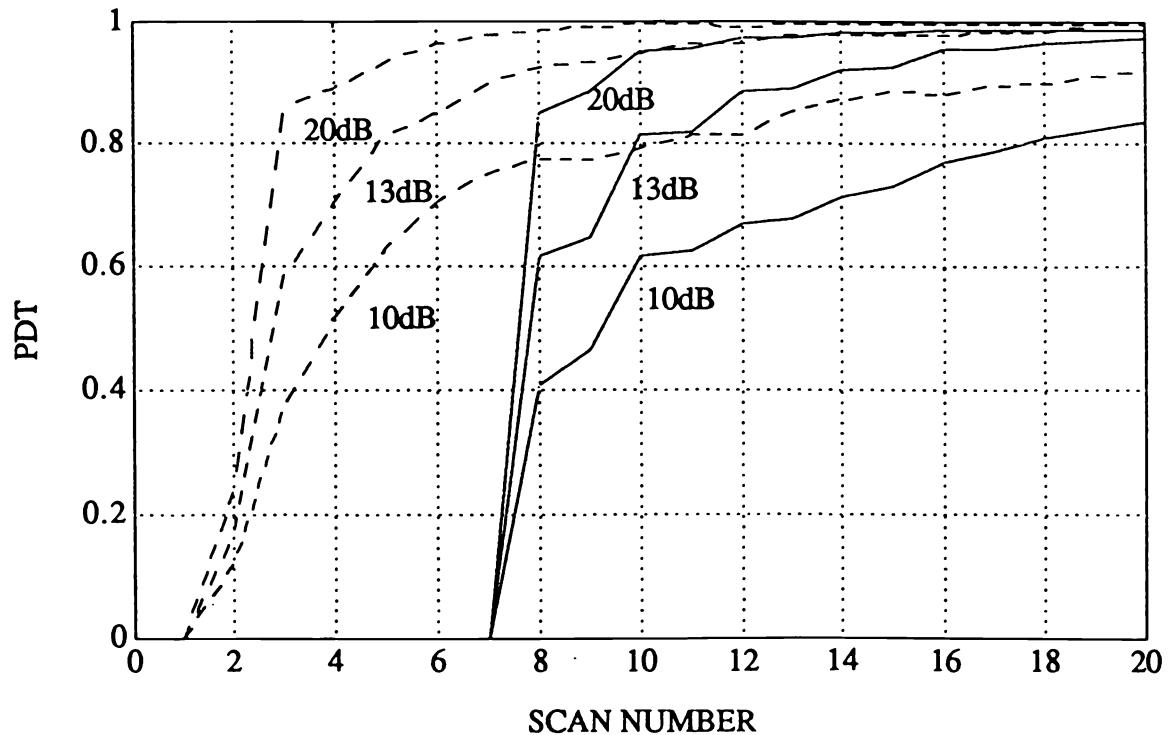


Figure 4.6.7-5: IMMPDAFAI: The probability of detecting a target track P_{DT} for $P_{FA} = 0.01$ (— $P_{FT} = 0.001$; - - $P_{FT} = 0.01$).

4.6.7 Results for Track Formation

Effect of Detection Threshold

The effect of increasing the clutter density is a degradation of target detection performance due to more stringent requirements for track acceptance. The best performance obtained among the three cases considered was for the highest detection threshold, even though this results in a lower probability of detection at each scan.

The effect of the track acceptance threshold on P_{DT} is small for $P_{FA} = 0.001$, but becomes more pronounced as P_{FA} increases.

Achieving better performance with a higher detection threshold is a desirable feature, since the computational complexity of a track formation algorithm will increase proportionally to the clutter density.

System Level Performance Measures

The measure of false track discrimination requires mapping P_{FT} into the **expected number of false tracks** that exist at each scan, $E(N_{FT}^e)$.

A **system level performance measure** is the **expected number of accepted false tracks** per scan, given by

$$E(N_{FT}) = E(N_{FT}^e)P_{FT} \quad (4.6.7-6)$$

These are shown in Table 4.6.7-2 together with:

$E(m)$ — the expected number of measurements (detections) provided to the algorithm per scan.

$E(I_{FT})$ — the expected interval of time between false track acceptances,

$E(L_{FT})$ — the expected life of a false track once accepted.

P_{FA}	$E(m)$	$E(N_{FT}^e)$	$E(N_{FT})$	$E(I_{FT})$	$E(L_{FT})$
per cell per scan	per surveillance area per scan			scans	
.001	10	5.9	.0059	169.5	1.0
.003	30	14.1	.0141	70.9	1.2
.010	100	39.3	.0393	25.4	1.7

Table 4.6.7-2: False track discrimination for $P_{FT} = 0.001$ (10,000 cell surveillance region).

The IMMPDAFAI achieves a reduction factor of $E(m)/E(N_{FT}) \approx 2000:1$.

4.6.7 Results for Track Formation

Real Data Results

A set of real data without a target was overlayed with a target with SNR=13dB and 10dB. Figure 4.6.7-6 shows the measurements in 6 beams of a sonar for 6 scans with the intensities indicated graphically.

In both cases considered the algorithm found the only target and calculated its TTP, which turned out to be 0.98 for the 13dB target and 0.67 for the 10dB target. Notice that in the 13dB case the target seems to “stand out” due to its amplitude; however, the measurements from the 10dB target cannot be identified visually — it is the (relative) consistency of the amplitude and the motion that allows the algorithm to “pull it out” from the noise. No other track with TTP exceeding 0.1 has been found by the algorithm.

4.6.7 Results for Track Formation

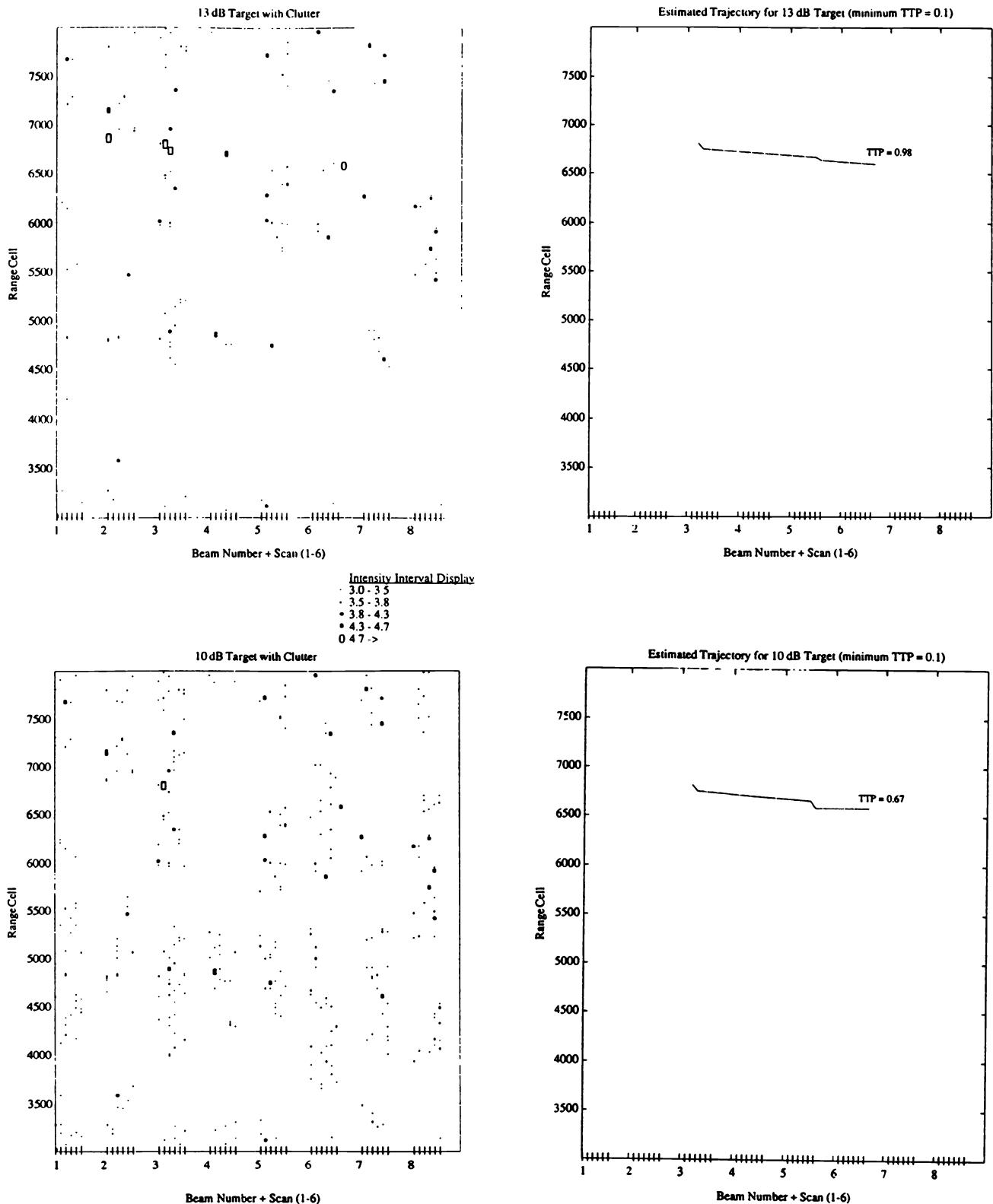


Figure 4.6.7-6: Track formation in real clutter.

4.6.8 Results for Track Maintenance

4.6.8 Results for Track Maintenance

The maneuver considered, shown in Figure 4.6.8-1, is a coordinated 180° turn that takes place over the duration of 10 sampling periods (turn rate of $18^\circ/\text{min}$).

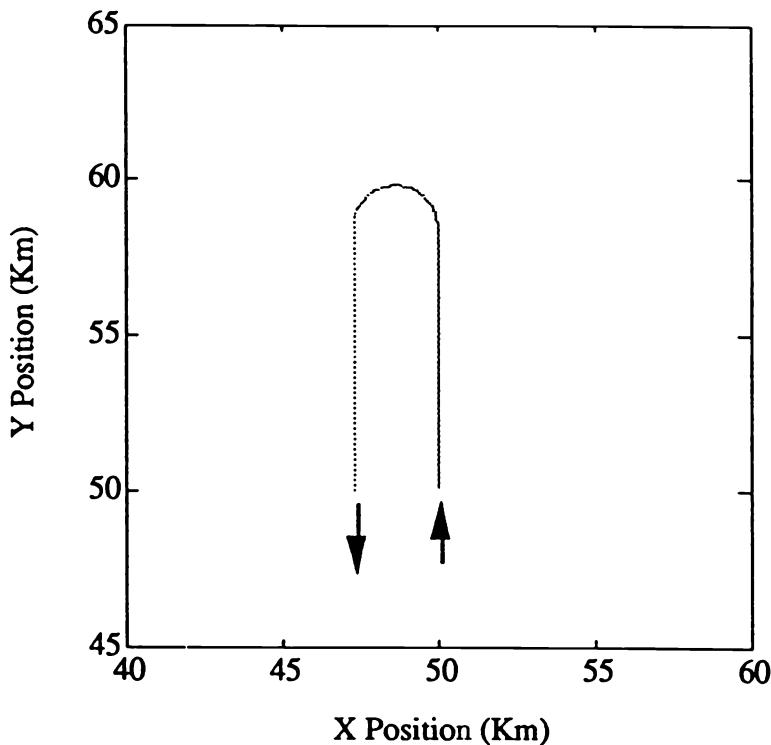


Figure 4.6.8-1: The target trajectory (U-turn of a U-boat).

The turn rate of 18° per sampling period is typical of many sensors which operate at much higher sampling rates, e.g., in civilian ATC a $3^\circ/\text{s}$ turn rate with $T = 5\text{s}$, or with military jets $20^\circ/\text{s}$ and $T = 1\text{s}$.

Figures 4.6.8-2 and 4.6.8-3 present the target velocity and acceleration.

4.6.8 Results for Track Maintenance

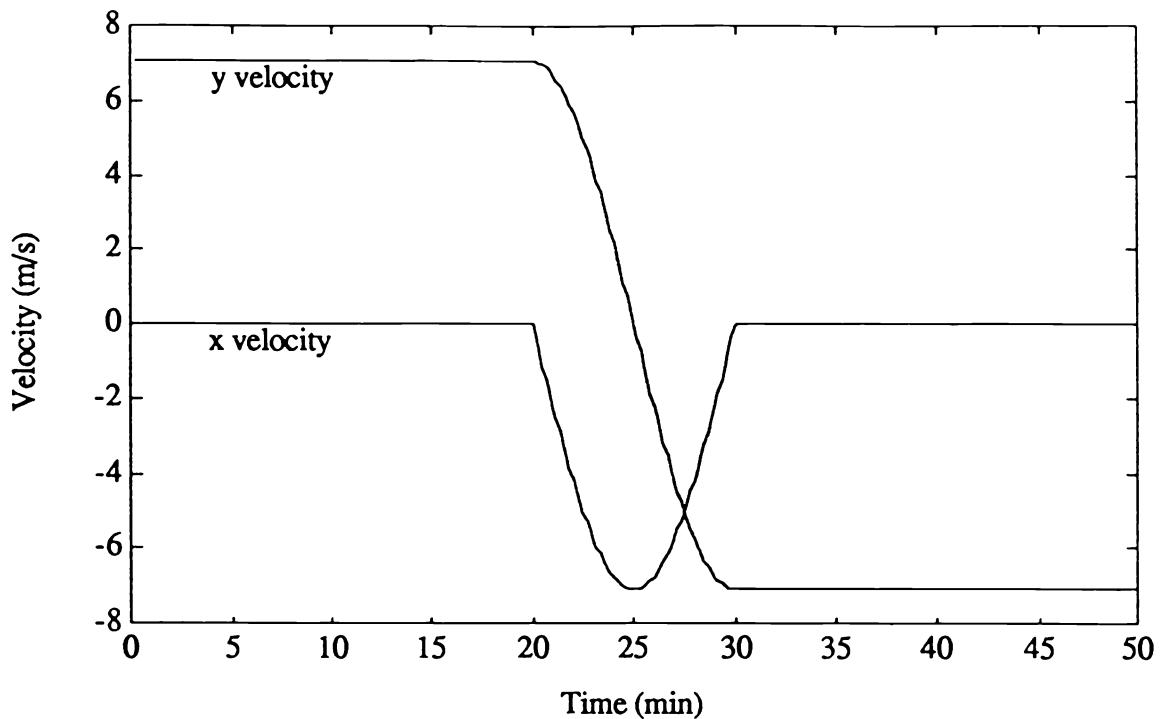


Figure 4.6.8-2: Target velocity.

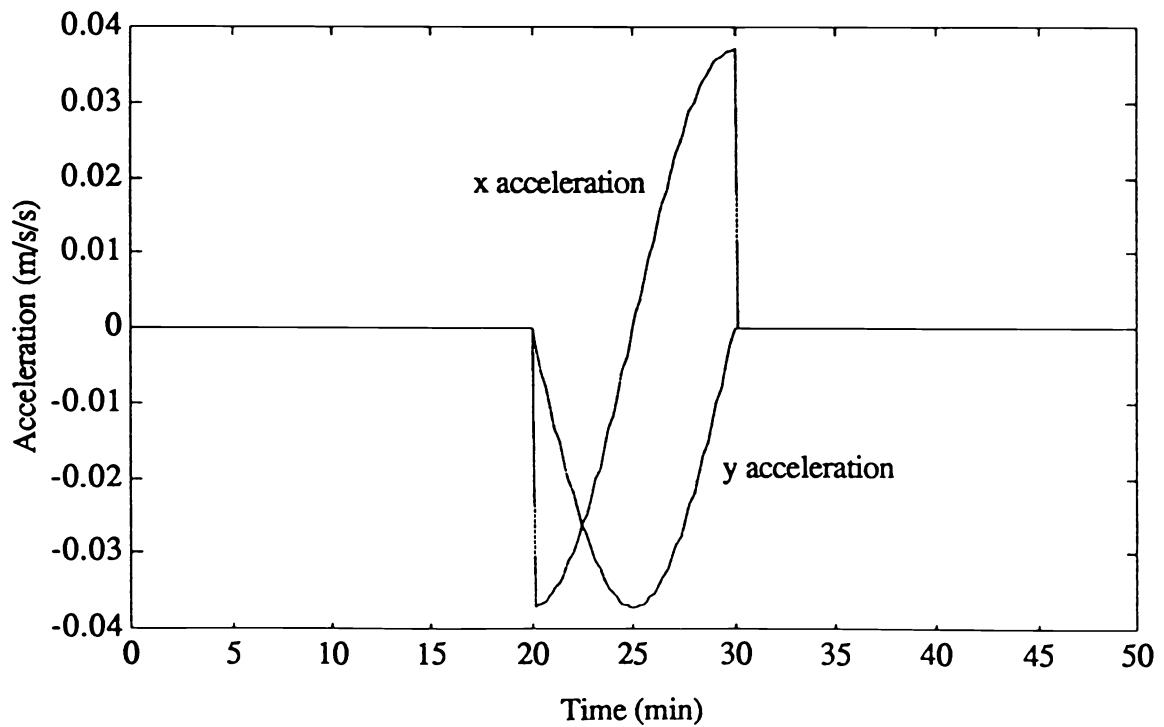


Figure 4.6.8-3: Target acceleration.

The IMMPDAFAI Configurations

In all configurations the model set $\{M^1, M^0\}$ is used for 10 measurements to perform track formation and a track acceptance decision is made at scan 10.

At scan 11, for accepted tracks, the mode set is augmented with maneuvering modes keeping model M^1 for the constant course and speed portions of the trajectory and model M^0 to dismiss the track if the target is lost.

Configuration 1. The mode set consists of three second order kinematic models with the following design parameters

M^0 : “no target” model with the same process noise as M^1 .

M^1 : white noise acceleration with variance $q_1 = 0.005^2$.

M^2 : white noise acceleration with variance $q_2 = 0.03^2$.

The assumed mode transition probability matrix is

$$\begin{bmatrix} .98 & .01 & .01 \\ .01 & .90 & .09 \\ .01 & .19 & .80 \end{bmatrix}$$

Configuration 2. The mode set consists of M^0 and M^1 and the two third order kinematic models

M^3 : white noise acceleration increment with variance $q_3 = 0.01^2$

M^4 : white noise acceleration increment with variance $q_4 = 0.03^2$.

The assumed mode transition probability matrix is

$$\begin{bmatrix} .97 & .01 & .01 & .01 \\ .02 & .80 & .09 & .09 \\ .02 & .09 & .80 & .09 \\ .02 & .09 & .09 & .80 \end{bmatrix}$$

Configuration 3. The mode set consists of models M^0 and M^1 and

M^5 : coordinated turn model for a left turn

M^6 : coordinated turn model for a right turn.

The latter models include a white noise acceleration with variance $q_5 = q_6 = 0.01^2$ in each coordinate.

4.6.8 Results for Track Maintenance

The assumed mode transition probability matrix is

$$\begin{bmatrix} .97 & .01 & .01 & .01 \\ .02 & .90 & .04 & .04 \\ .02 & .11 & .85 & .02 \\ .02 & .11 & .02 & .85 \end{bmatrix}$$

The target SNR was 20dB and the probability of false alarm 0.001.

Simulations for each IMMPDAFAI configuration with the mode sets as indicated above were performed using 200 independent *post-detection runs* i.e., where the target has passed the track acceptance decision at scan 10.

A **lost track** is declared if the target measurement falls outside the largest track gate for all models being considered, or the probability of the “no target” mode exceeds 0.95 (i.e., TTP < 0.05).

The Results

The average mode probabilities for each configuration over time are shown in Figures 4.6.8-4 through 4.6.8-6.

The position, speed and course errors are shown in Figures 4.6.8-7 through 4.6.8-9.

The estimated turn rate for Configuration 3 — this is the only one that estimates it — is shown in Figure 4.6.8-10. The true value of the turn rate was, for the maneuver considered, $\pi/600 \approx 5 \text{ mrad/s}$.

Figures 4.6.8-11 and 4.6.8-12 show the normalized estimation error squared (NEES) for position and velocity, respectively, using the filter calculated covariance.

Before the maneuver, the filters are somewhat pessimistic (the NEES is below its expected value 2), while for a short time at the end of the maneuver the filters are somewhat optimistic (NEES greater than 2).

The latter situation occurs because it is the inconsistency (errors larger than expected) that drives the adaptation. Once the filter adapts itself by changing the mode probabilities the errors return to be consistent with the calculated covariances.

Configuration 1 suffers the most from inconsistency. Configurations 2 and 3 recover in one to two samples (scans) — as fast as one can expect.

4.6.8 Results for Track Maintenance

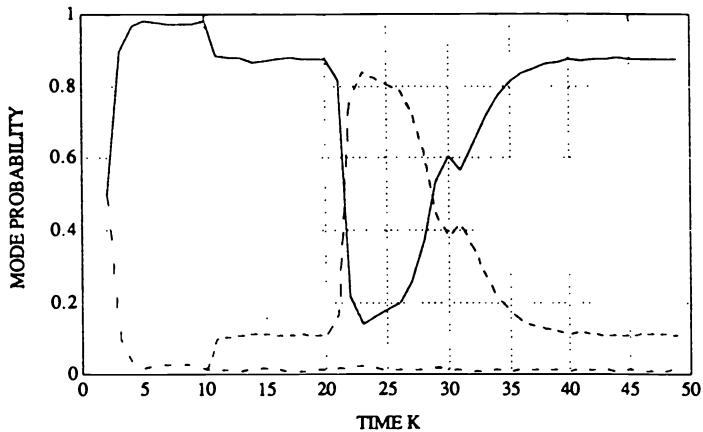


Figure 4.6.8-4: Config. 1 average mode probabilities ($\cdots M^0$; $- M^1$; $-- M^2$).

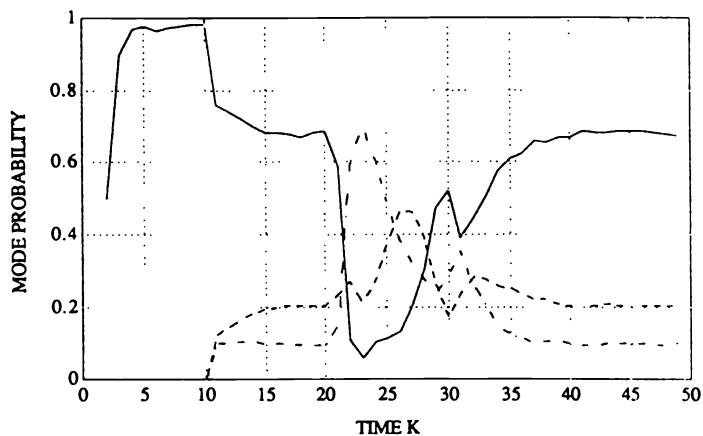


Figure 4.6.8-5: Config. 2 average mode probabilities ($\cdots M^0$; $- M^1$; $-- M^3$; $- - M^4$).

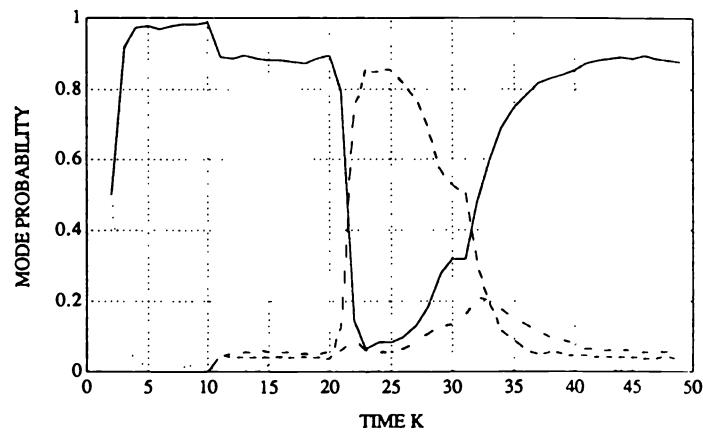


Figure 4.6.8-6: Config. 3 average mode probabilities ($\cdots M^0$; $- M^1$; $-- M^5$; $- - M^6$).

4.6.8 Results for Track Maintenance

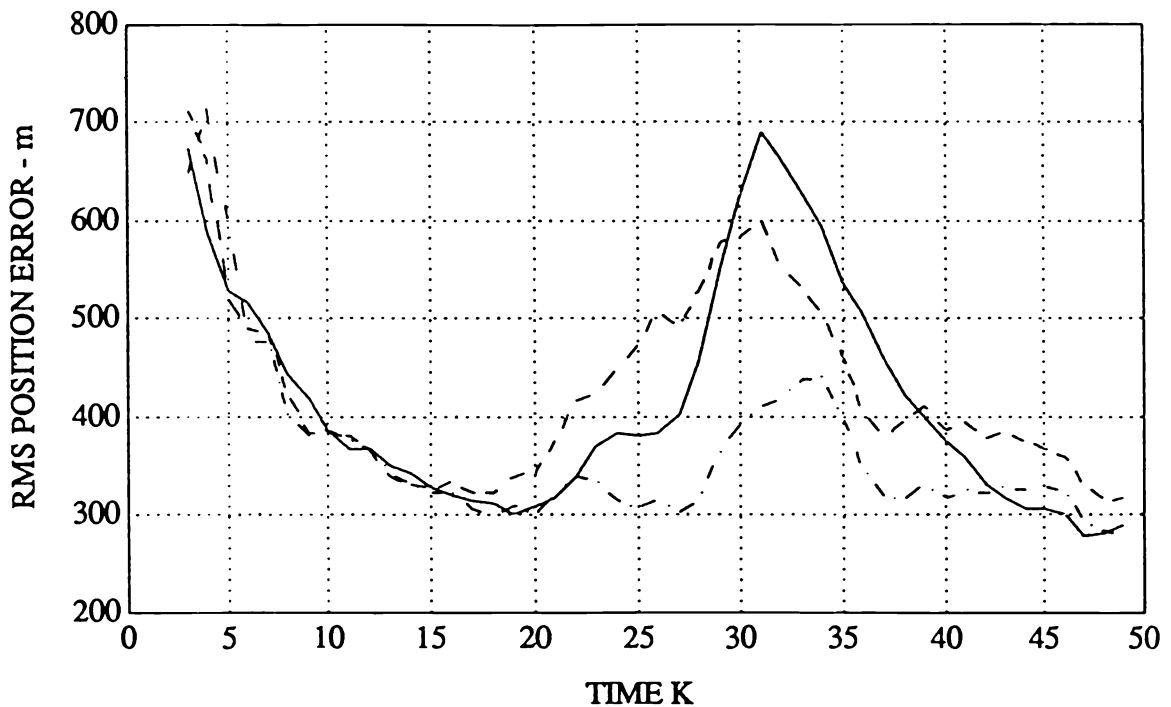


Figure 4.6.8-7: RMS position error (— Config. 1; - - Config. 2; - · - Config. 3).

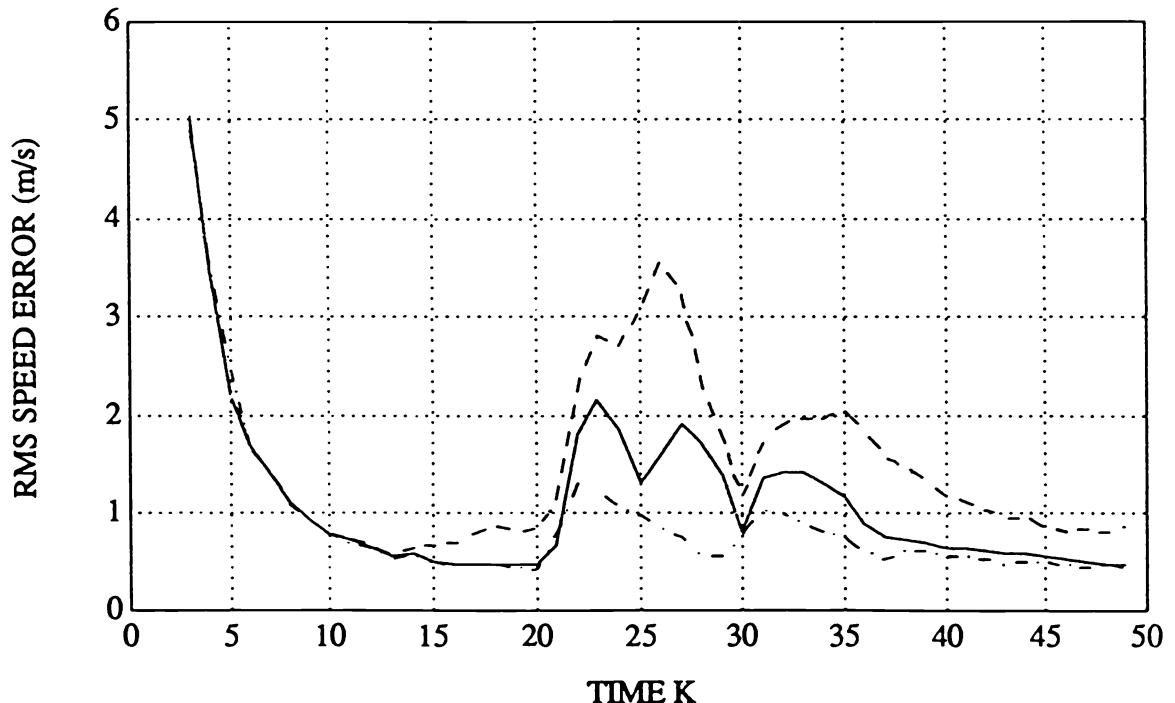


Figure 4.6.8-8: RMS speed error (— Config. 1; - - Config. 2; - · - Config. 3).

4.6.8 Results for Track Maintenance

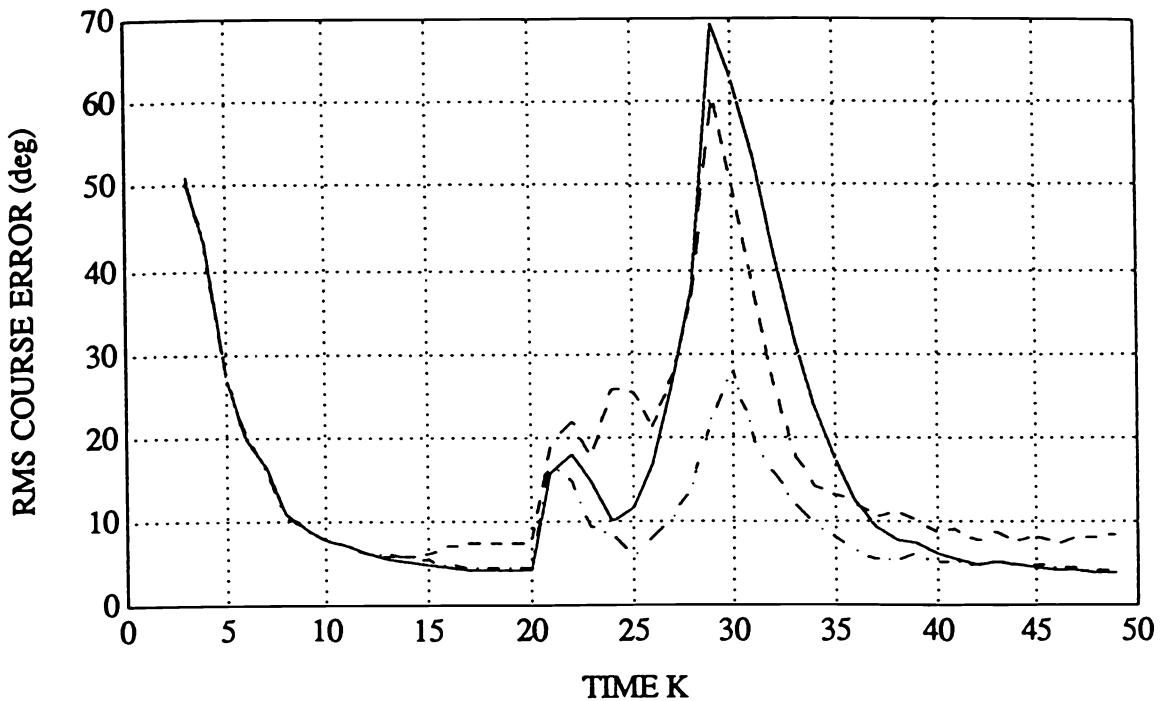


Figure 4.6.8-9: RMS course error (— Config. 1; - - Config. 2; - · - Config. 3).

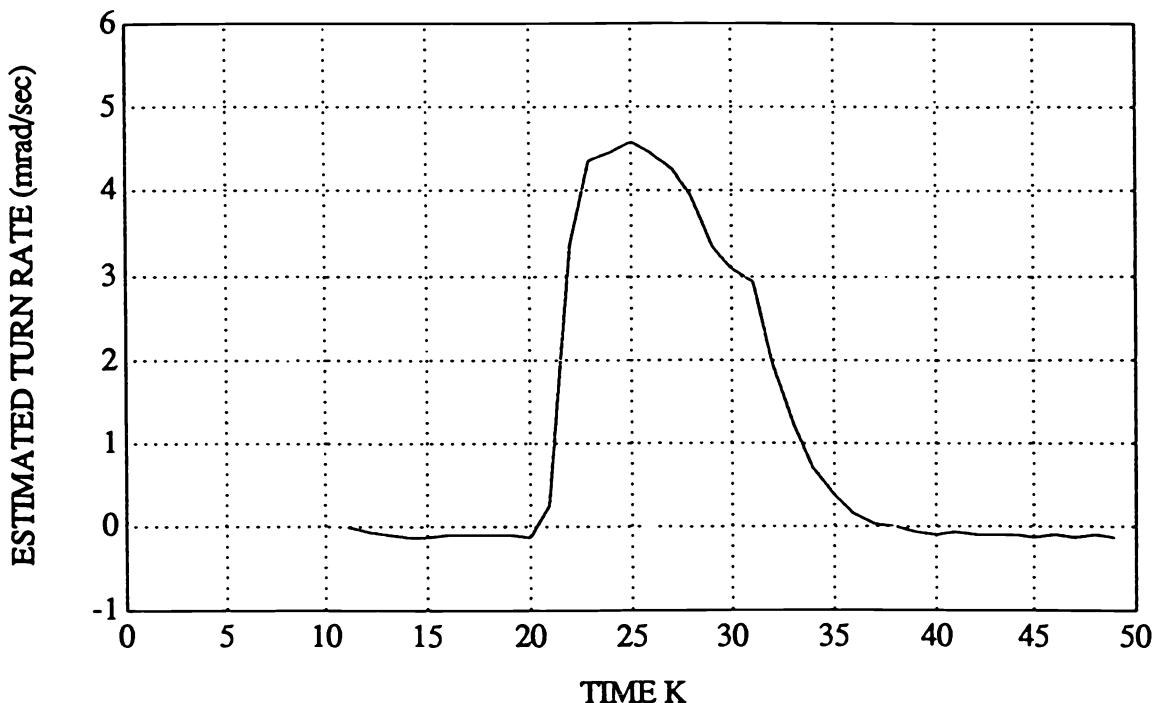


Figure 4.6.8-10: Average of estimated turn rate (Config. 3).

4.6.8 Results for Track Maintenance

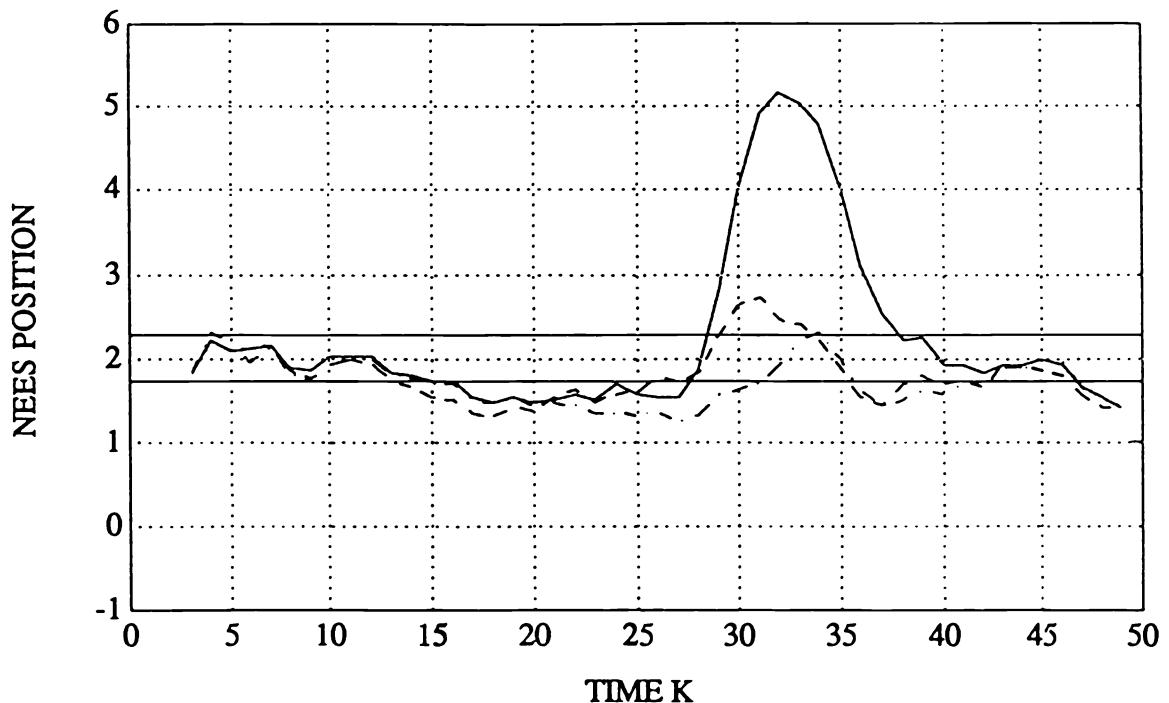


Figure 4.6.8-11: Position NEES with its 95% probability region (— Config. 1; - - Config. 2; - - - Config. 3).

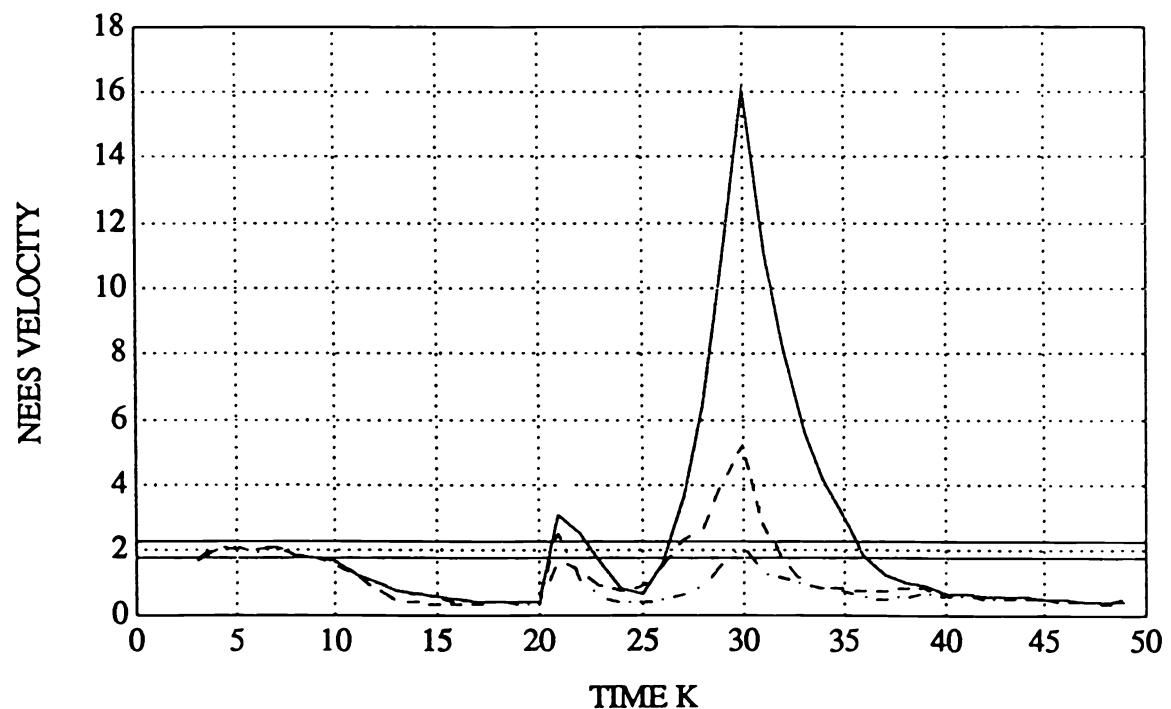


Figure 4.6.8-12: Velocity NEES with its 95% probability region (— Config. 1; - - Config. 2; - - - Config. 3).

4.6.8 Simulations Results for Track Maintenance

Track Loss

The number of lost tracks in 1000 runs for each configuration as well as for the standard Kalman filter (which was using the *strongest measurement in the gate*) is shown in Table 4.6.8-1. For all SNR the percentage of lost tracks is less than 9% for all configurations of the IMMPDAFAI.

	20dB	13dB	10dB
Configuration 1	1.5	4.4	8.6
Configuration 2	1.7	3.2	2.2
Configuration 3	1.7	3.0	4.4
Standard Kalman	25.7	81.5	97.3

Table 4.6.8-1: Percentage of lost tracks ($P_{FA} = 0.001$).

The best choice is configuration 3.

4.6.9 The IMMPDAFAI — Summary

An approach has been presented that utilizes the strength of target returns for improved

- track formation in a cluttered environment, as well as
- track maintenance.

The algorithm has several notable features:

- it discards false tracks and lost tracks quickly and
- has the ability to make target track detections with high probability using a small number of observations.

Once target track detections are made, a means of performing reliable track continuation through potential target maneuvers has been provided.

The track formation procedure does not seriously degrade in heavy clutter and performs well for weak target measurements — low observable targets.

In track formation the IMMPDAFAI outperforms the Sequential Detector, in particular for low SNR (for high SNR any scheme works).

In track maintenance the IMMPDAFAI is the only reliable algorithm.

The best configuration for the IMMPDAFAI is the one using coordinated turn models.

Note on the Feature Model

The Rayleigh distributed amplitude feature in Subsection 4.6.2 is known in the literature as the **Swerling I fluctuation model**.

4.7 NOTES AND PROBLEMS

4.7.1 Bibliographical Notes

The algorithm that incorporates the measurement rank as a feature component of the measurement vector to increase the power of the data association in the presence of maneuvers (Section 4.2) is from [NSC84].

The Multiple Model PDAF from Section 4.3 is based on [Gau84].

A combination of the N -scan-back approach of Subsection 3.5.4 with the Markov switching models approach of Subsection 11.6.3 of [BL93] simplified to limit its memory requirements was presented in [Ken81]. The models were assumed to have different input levels as maneuvers.

The combination of the IMM method of Subsection 11.6.6 of [BL93] with the PDAF for an air traffic control application has been reported in [Blo84].

The IMMPDAF as an Automatic Track Formation algorithm presented in Section 4.4 is based on Ch. 2 of [Bar90] and has been implemented in the interactive software MULTIDATTM.

The multisensor IMMPDAF from Section 4.5 is based on [HB89].

The IMMPDAFAI described in Section 4.6 is from [LB93a]. Extensive discussion of detection models for radar (and sonar) can be found in [DR80].

4.7.2 Problems

4-1 *Spread of the association probabilities.*

- (i) Find the maximum and minimum values of (4.2.2-1) in terms of $m(k)$.
- (ii) Indicate to what uncertainties these extrema correspond.

Chapter 5

TRACKING PERFORMANCE PREDICTION AND DETECTION THRESHOLD OPTIMIZATION

5.1 INTRODUCTION

5.1.1 Outline

The performance of the standard Kalman filter, measured by the state estimation covariance, follows from the matrix Riccati difference equation and it depends on the process and measurement noise covariances.

When there is also **origin uncertainty in the measurements**, the tracking performance depends critically on two additional parameters: the probabilities of target detection P_D and false alarm P_{FA} .

These two probabilities follow from the *selection of the detection threshold in the signal processing subsystem* of the tracking system, illustrated in Figure 5.1.1-1. For a given signal-to-noise ratio (SNR) the detection threshold determines the operating point on the Receiver Operating Characteristic (ROC) of the detector. Raising the threshold lowers both P_D and P_{FA} and vice-versa.

The common procedure is to set the threshold, which is a system design parameter, based on the designer's perception of trade-offs between false alarms and missed detections. A systematic quantitative consideration of the downstream effects of the **detection threshold selection** on the **tracking performance** is developed in this chapter.

5.1.1 Outline

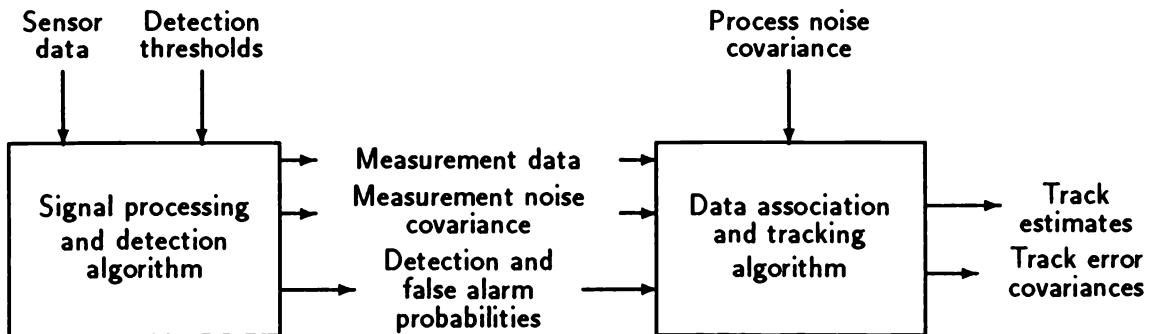


Figure 5.1.1-1: Tracking system block diagram.

The discussion is based on the PDAF, whose estimation accuracy is described by the *stochastic matrix Riccati equation* (Subsection 3.4.2). Section 5.2 presents the covariance of the combined innovation in the PDAF needed for **nonsimulation performance prediction**. Section 5.3 summarizes the stochastic Riccati equation and presents an approximate solution to it via the **average Riccati equation**.

Section 5.4 develops the **Tracker Operating Characteristic (TOC)** contours and shows how one can optimize the tracking accuracy w.r.t. the detection threshold for a given SNR.

Section 5.5 discusses, using a similar methodology, a more accurate **hybrid approximation of the Riccati equation** to yield also the transient behavior — **track life** — without the need for expensive Monte Carlo simulations.

5.1.2 Summary of Objectives

5.1.2 Summary of Objectives

Relate, in the context of the PDAF, tracking performance to the detection threshold from the signal processing subsystem.

Discuss the covariance of the combined innovation.

Present the stochastic Riccati equation that yields the tracking accuracy.

Approximate it by a deterministic “average” for *nonsimulation performance prediction*.

Introduce the “information reduction factor” due to the false measurements and $P_D < 1$.

Present the Tracker Operating Characteristic and relate it to the detector’s ROC.

Present a hybrid approximation of the stochastic Riccati equation that *also captures the transient*.

5.2 THE COVARIANCE OF THE COMBINED INNOVATION IN THE PDAF

5.2.1 The Combined Innovation

The **combined innovation** of the PDAF is

$$\nu(k) = \sum_{i=1}^{m(k)} \beta_i(k) \nu_i(k) \quad (5.2.1-1)$$

where the association probabilities have the expressions

$$\beta_i(k) = \frac{e_i}{b + \sum_{j=1}^{m(k)} e_j} \quad i = 1, \dots, m(k) \quad (5.2.1-2)$$

$$\beta_0(k) = \frac{b}{b + \sum_{j=1}^{m(k)} e_j} \quad (5.2.1-3)$$

$$e_i = e^{-\frac{1}{2} \nu_i(k)' S(k)^{-1} \nu_i(k)} \quad (5.2.1-4)$$

and b is the constant given by (3.4.3-14).

The covariance of the combined innovation (5.2.1-1) to be evaluated is

$$S^*(k) \triangleq E[\nu(k)\nu(k)'|Z^{k-1}] = E\left[\sum_{i=1}^{m(k)} \beta_i(k) \nu_i(k) \sum_{j=1}^{m(k)} \beta_j(k) \nu_j(k)' | Z^{k-1} \right] \quad (5.2.1-5)$$

Note the fact that (5.2.1-5) is a highly nonlinear function of the innovations because of the exponentials (5.2.1-4) entering into the probabilities $\beta_i(k)$.

The expectation (5.2.1-5) will be carried out using the smoothing property of the expectations (1.4.12-3) of [BL93]

$$S^*(k) = E \left\{ E \left[\sum_{i=1}^{m(k)} \beta_i(k) \nu_i(k) \sum_{j=1}^{m(k)} \beta_j(k) \nu_j(k)' | m(k), Z^{k-1} \right] | Z^{k-1} \right\} \quad (5.2.1-6)$$

The inner expectation is conditioned on $m(k)$, the number of validated measurements, which is then averaged out in the outside expectation. In order to average out $m(k)$, the Poisson clutter model (3.4.3-6) is used — the diffuse prior is not suitable in this case because it has no moments.

5.2.1 The Combined Innovation

The joint pdf of the innovations in the validation region is the mixture

$$\begin{aligned} p[\nu_1(k), \dots, \nu_{m(k)}(k) | m(k), Z^{k-1}] &= V(k)^{-m(k)} \gamma_0[m(k)] \\ &\quad + V(k)^{-m(k)+1} \sum_{j=1}^{m(k)} P_G^{-1} \mathcal{N}[\nu_j(k); 0, S(k)] \gamma_j(m(k)) \end{aligned} \quad (5.2.1-7)$$

where $V(k)$ is the validation region volume and γ_j is the prior probability for the corresponding measurement being correct.

The innovations $\nu_j(k)$ are *identically distributed* but *not independent* because, if one is correct (normally distributed) then the others are false (uniformly distributed).

Note that the innovations are symmetrically distributed within the validation region. Both the uniform and the normal pdf in the mixture (5.2.1-7) are symmetric and thus also zero-mean. Consequently, all the cross terms in the multiplication of the two sums in (5.2.1-6) will be zero. Therefore

$$\begin{aligned} S^*(k) &= E \left\{ E \left[\sum_{i=1}^{m(k)} \beta_i(k)^2 \nu_i(k) \nu_i(k)' | m(k), Z^{k-1} \right] | Z^{k-1} \right\} \\ &= E \{ E[m(k) \beta_i(k)^2 \nu_i(k) \nu_i(k)' | m(k), Z^{k-1}] | Z^{k-1} \} \end{aligned} \quad (5.2.1-8)$$

where the last equality above follows from the fact that each summand above has the same expected value.

A lengthy derivation (see [FBG85, JB90]) yields for the inner expectation in (5.2.1-8)

$$\begin{aligned} U_2[m(k)] &\triangleq E[m(k) \beta_i(k)^2 \nu_i(k) \nu_i(k)' | m(k), Z^{k-1}] \\ &= u_2[m(k)] S(k) \end{aligned} \quad (5.2.1-9)$$

where $u_2(m)$ is a scalar function of the number of validated measurements.

To carry out the outside expectation in (5.2.1-8) requires averaging over the possible values of $m(k)$ according to its pmf with the Poisson prior (3.4.3-6) with expected number of false measurements $\lambda V(k)$ in the validation region.

5.2.2 The Decrease Factor

5.2.2 The Decrease Factor

The **covariance of the combined innovations**, $S^*(k)$, is proportional to the covariance of the true innovation, $S(k)$, with a *factor of proportionality less than unity* (due to the larger weighting given to the smaller innovations)

$$S^*(k) = q_2[P_D, \lambda V(k)]S(k) \quad (5.2.2-1)$$

The factor q_2 decreases as

- the target P_D decreases
- the density (expected number) of false measurements increases.

The evaluation of the coefficient q_2 , which is the average of u_2 from (5.2.1-9), using numerical integration, yields the results plotted in Figure 5.2.2-1 in terms of the expected number of false measurements $\lambda V(k)$, for various values of the target detection probability P_D .

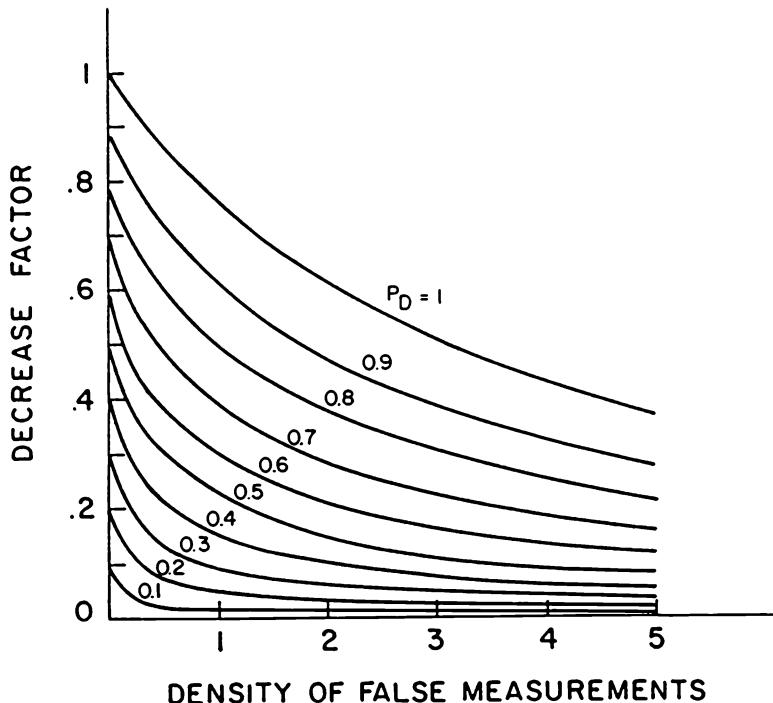


Figure 5.2.2-1: The covariance decrease factor for 2-dimensional combined innovations with a $g = 4$ validation region.

5.3 TRACKING ACCURACY IN CLUTTER

5.3.1 The Stochastic Riccati Equation

The covariance propagation equations for the PDAF are, from Section 3.4,

$$P(k|k-1) = F(k-1)P(k-1|k-1)F(k-1)' + Q(k-1) \quad (5.3.1-1)$$

$$P(k|k) = \beta_0(k)P(k|k-1) + [1 - \beta_0(k)]P^c(k|k) + \tilde{P}(k) \quad (5.3.1-2)$$

for the prediction and update, respectively, where

$$\tilde{P}(k) = W(k) \left[\sum_{i=1}^{m(k)} \beta_i(k) \nu_i(k) \nu_i(k)' - \nu(k) \nu(k)' \right] W(k)' \quad (5.3.1-3)$$

$$W(k) = P(k|k-1)H(k)'S(k)^{-1} \quad (5.3.1-4)$$

$$S(k) = H(k)P(k|k-1)H(k)' + R(k) \quad (5.3.1-5)$$

$R(k)$ is the measurement noise covariance and

$$P^c(k|k) = P(k|k-1) - W(k)S(k)W(k)' \quad (5.3.1-6)$$

is the updated covariance with the correct measurement, i.e., in the absence of measurement origin uncertainty.

Note the stochastic terms $\beta_0(k)$ and $\tilde{P}(k)$ in the covariance update equation (5.3.1-2) of the PDAF. If they are zero, the covariance update equation yields the standard Riccati equation of the Kalman filter.

When these two terms are not zero, then, since they are random, one has a **stochastic Riccati equation** that characterizes the **tracking accuracy** of the PDAF: the state estimation covariance is a random process.

We are interested in characterizing this random process for

- tracking performance prediction for given system parameters
- performance optimization with respect to some design parameters.

to *avoid the need for Monte Carlo simulations*.

5.3.2 Approximation of the Stochastic Riccati Equation

A *global averaging* approximation of the *stochastic covariance update equation* (5.3.1-2) is obtained by replacing the stochastic terms in it with their expected values

$$\bar{P}(k) = E[\tilde{P}(k)|Z^{k-1}] \quad (5.3.2-1)$$

$$\begin{aligned} \bar{\beta}_0(k) &= E[\beta_0(k)|Z^{k-1}] \\ &= E[\beta_0(k)] \\ &= 1 - P_D P_G \end{aligned} \quad (5.3.2-2)$$

These substitutions will make (5.3.1-2) into a *deterministic* equation that can be iterated forward in time in conjunction with the *covariance prediction equation* (5.3.1-1), which is deterministic.

In view of the fact that (5.3.1-2) is nonlinear in $P(k|k-1)$ — due to the inversion in (5.3.1-4) — this approach does not yield $E[P(k|k)]$; nevertheless, it will give approximate values of the future *average state error covariance in the presence of uncertain detections and false alarms*.

The resulting uncertainties will depend on the environmental parameters P_D and P_{FA} , as well as on the noise covariances Q and R .

Eq. (5.3.2-1) can be rewritten using (1.4.12-3) of [BL93] as

$$\begin{aligned} \bar{P}(k) &= E[\tilde{P}(k)|Z^{k-1}] \\ &= E\{E[\tilde{P}(k)|m, Z^{k-1}]|Z^{k-1}\} \\ &= \sum_{m=0}^{\infty} E[\tilde{P}(k)|m, Z^{k-1}] P\{m|Z^{k-1}\} \end{aligned} \quad (5.3.2-3)$$

where m is the number of validated measurements at time k (its time argument omitted for simplicity).

In the above, the pmf (probability mass function) of m is

$$P\{m|Z^{k-1}\} = P_D P_G \mu_F(m-1) + (1 - P_D P_G) \mu_F(m) \quad (5.3.2-4)$$

where $\mu_F(\cdot)$ is the pmf of the number of false measurements. In view of the summation in (5.3.2-3) it is clear that a proper pmf is needed.

5.3.2 Approximation of the Stochastic Riccati Equation

Using (5.3.1-3) in (5.3.2-1) yields

$$\begin{aligned}\bar{P}(k) &= E \left\{ E \left\{ W(k) \left[\sum_{i=1}^m \beta_i(k) \nu_i(k) \nu_i(k)' - \nu(k) \nu(k)' \right] W(k)' | m, Z^{k-1} \right\} | Z^{k-1} \right\} \\ &= E \{ W(k) [U_1(m) - U_2(m)] W(k)' | Z^{k-1} \}\end{aligned}\quad (5.3.2-5)$$

where

$$U_1(m) \triangleq E \left[\sum_{i=1}^m \beta_i(k) \nu_i(k) \nu_i(k)' | m, Z^{k-1} \right] \quad (5.3.2-6)$$

and, as in (5.2.1-9),

$$\begin{aligned}U_2(m) &\triangleq E[\nu(k) \nu(k)' | m, Z^{k-1}] \\ &= E \left[\sum_{i=1}^m \beta_i(k) \nu_i(k) \sum_{j=1}^m \beta_j(k) \nu_j(k)' | m, Z^{k-1} \right]\end{aligned}\quad (5.3.2-7)$$

Lengthy derivations (see [FBG85, JB90, LB91b]) yield

$$U_1(m) = u_1(m) S(k) \quad (5.3.2-8)$$

$$U_2(m) = u_2(m) S(k) \quad (5.3.2-9)$$

where u_1 and u_2 are scalar functions of m and $S(k)$ is the covariance of the correct innovation.

Using (5.3.2-8) and (5.3.2-9) and carrying out the outside averaging in (5.3.2-5) yields

$$\bar{P}(k) = (q_1 - q_2) W(k) S(k) W(k)' \quad (5.3.2-10)$$

where q_1 and q_2 are the averages of u_1 and u_2 , respectively.

The functions q_1 and q_2 depend on the target detection probability, the parameter(s) of the false measurement pmf, and gate size. Using a large gate with $\gamma = 16$ (for which $P_G = 1$, practically) and a Poisson model for the false measurements, one obtains

$$q_1 = P_D \quad (5.3.2-11)$$

$$q_2 = q_2[P_D, \lambda V(k)] \quad (5.3.2-12)$$

where the latter is the same as in (5.2.2-1) and is plotted in Figure 5.2.2-1.

5.3.2 Approximation of the Stochastic Riccati Equation

Using (5.3.1-6) in (5.3.1-2), one has

$$\begin{aligned} P(k|k) &= \beta_0(k)P(k|k-1) - [1 - \beta_0(k)][P(k|k-1) - W(k)S(k)W(k)'] + \tilde{P}(k) \\ &= P(k|k-1) - [1 - \beta_0(k)]W(k)S(k)W(k)' + \tilde{P}(k) \end{aligned} \quad (5.3.2-13)$$

Substituting (5.3.2-1), (5.3.2-2) and (5.3.2-10) in (5.3.2-13) yields the *deterministic equation*

$$P(k|k) = P(k|k-1) - [P_D P_G - q_1 + q_2]W(k)S(k)W(k)' \quad (5.3.2-14)$$

The Average Covariance Update Equation

With (5.3.2-11) and $P_G = 1$, Eq. (5.3.2-14) becomes the **average covariance update equation**

$$P(k|k) = P(k|k-1) - q_2[S(k), P_D, P_F]W(k)S(k)W(k)' \quad (5.3.2-15)$$

Note the similarity between the standard covariance update equation (5.3.1-6) and the above: the only difference is the factor q_2 that quantifies the *reduced information obtained from the observations*.

The term q_2 , called the **information reduction factor**, depends on

- the target detection probability P_D and
- the expected number of false measurements in the gate.

The latter depends on the false alarm probability P_{FA} and the volume $V(k)$ of the validation region, determined in turn by the innovation covariance matrix $S(k)$.

The equations to be iterated forward for evaluating (approximately) the tracking performance in clutter are the *average covariance update equation* (5.3.2-15) and the prediction equation (5.3.1-1). Together they will be referred to as the **modified Riccati equation** or the **average Riccati equation**.

This yields a tool for **nonsimulation performance prediction**.

5.3.3 Tracking Accuracy in Clutter — Summary

5.3.3 Tracking Accuracy in Clutter — Summary

In the standard linear state estimator the covariance of the state is propagated via

- the prediction equation coupled with
- the update equation.

These two equations combined yield the (matrix) *Riccati equation*, which is deterministic.

In the PDAF the covariance update equation is *stochastic*.

Combining the deterministic covariance prediction equation with the stochastic update equation results in a *stochastic Riccati equation* for the state estimation covariance from the PDAF.

The *stochastic Riccati equation* is approximated by replacing the stochastic terms in it with their expected values, yielding the *modified Riccati equation*, which allows *nonsimulation performance prediction*.

In the modified Riccati equation the *covariance decrease* in the update is *reduced* by the *combined innovation's covariance decrease factor* discussed earlier.

This *information reduction factor* depends on

- the target's probability of detection
- the expected number of false measurements in the validation region.

Plots of the information reduction factor (for $n_z = 2$ and gate $g = 4$) and various values of the above parameters have been obtained. These are *universal curves* because they hold for any situation with a 2-dimensional measurement.

5.4 THE TRACKER OPERATING CHARACTERISTIC

5.4.1 Iteration of the Modified Riccati Equation

For given values of P_D and P_{FA} one can iterate forward the modified Riccati equation consisting of

$$P(k|k-1) = F(k)P(k-1|k-1)F(k)' + Q(k) \quad (5.4.1-1)$$

$$P(k|k) = P(k|k-1) - q_2[S(k), P_D, P_{FA}]W(k)S(k)W(k)' \quad (5.4.1-2)$$

where

$$S(k) \triangleq H(k)P(k|k-1)H(k)' + R(k) \quad (5.4.1-3)$$

$$W(k) \triangleq P(k|k-1)H(k)'S(k)^{-1} \quad (5.4.1-4)$$

The iteration can be done until steady-state is reached or, alternatively, the covariance might diverge.

Clutter Density vs. False Alarm Probability

The clutter density (expected number of false measurements per unit volume) relates to the false alarm probability, which is defined usually per cell of the signal processor. The relationship is

$$\lambda = \frac{P_{FA}}{V_c} \quad (5.4.1-5)$$

where V_c is the **resolution cell volume**.

The *validation region volume* corresponding to the threshold $\gamma = g^2$ ("g-sigma" gate) is

$$V(k) = V[S(k)] = c_{n_z}|\gamma S(k)|^{1/2} = c_{n_z}g^{n_z}|S(k)|^{1/2} \quad (5.4.1-6)$$

where n_z is the dimension of the measurement and c_{n_z} is the volume of the unit hypersphere of this dimension.

For the Poisson clutter model the *information reduction factor* q_2 has the functional form

$$q_2 = q_2\{P_D, \lambda V[S(k)]\} \quad (5.4.1-7)$$

where λ follows from P_{FA} according to (5.4.1-5).

The Iterations

The sequence of iterations (one cycle) is:

1. From $P(k-1|k-1)$ obtain $P(k|k-1)$ using (5.4.1-1)
2. From (5.4.1-3) obtain $S(k)$
3. From (5.4.1-6) obtain V
4. The factor q_2 follows as in (5.4.1-7) from the curves given in Figure 5.2.2-1
5. With the gain (5.4.1-4), $P(k|k)$ follows from (5.4.1-2).

The Tracker Operating Characteristic

A *norm of the steady-state covariance* — the **tracking accuracy** — can then be represented in terms of the two basic parameters P_D and P_{FA} in the form of the **Tracker Operating Characteristic (TOC)**.

An example of a set of TOC curves for a certain problem (discussed in detail in Section 6.3) is presented in Figure 5.4.1-1. The curves represent the locus of points in the P_D - P_{FA} space corresponding to various levels of position error in steady state. A **tracking instability region** exists: for low P_D and high P_{FA} the iteration diverges, as one can expect.

Detection Threshold Optimization

Figure 5.4.1-2 presents a Receiver Operating Characteristic (ROC) for a quadrature receiver for various SNRs. Superimposing, e.g., the 8dB ROC from Figure 5.4.1-2 on the TOC plot shows how one can choose the **optimal detection threshold** (ROC operating point) for **tracking error minimization**.

5.4.1 Iteration of the Modified Riccati Equation

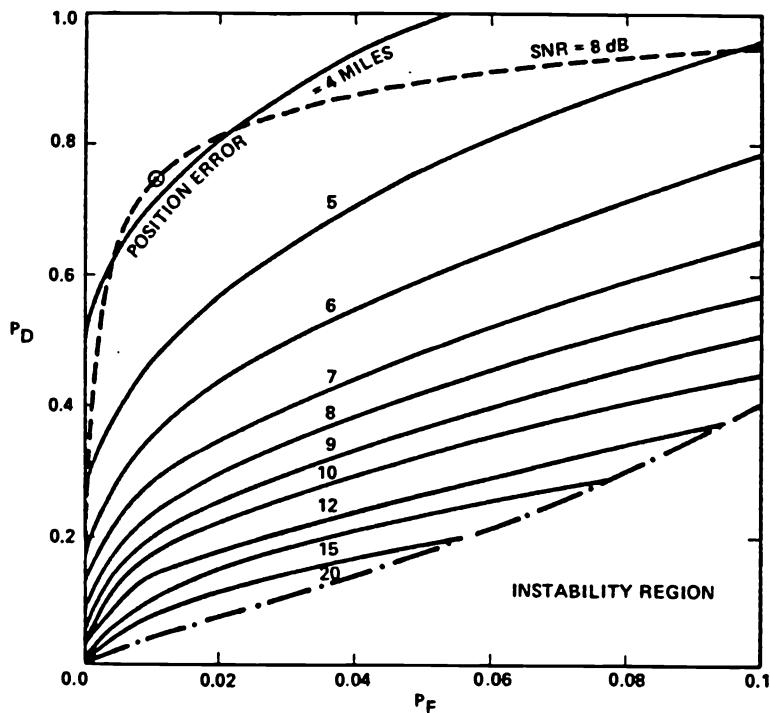


Figure 5.4.1-1: Tracker operating characteristic (TOC) curves for a problem with bearing and frequency measurements.

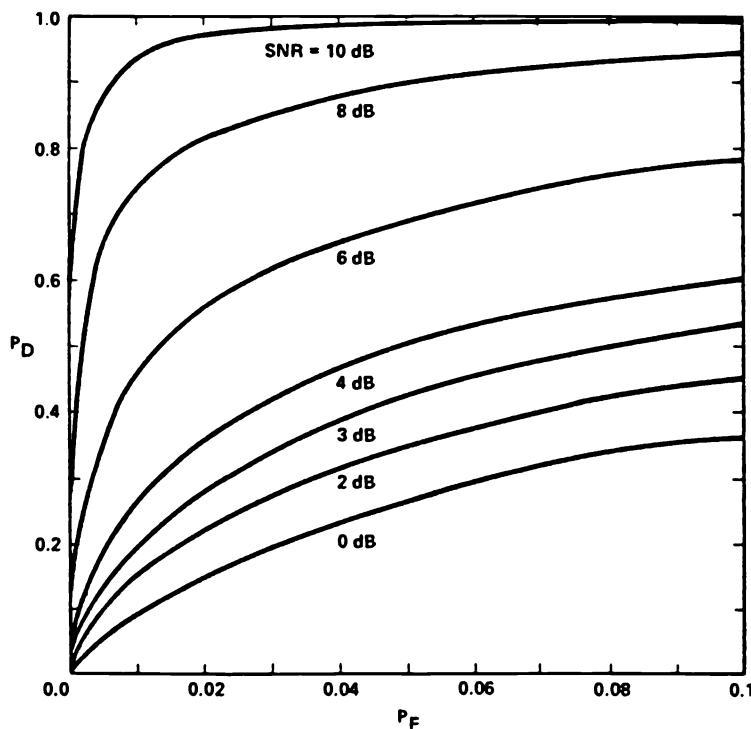


Figure 5.4.1-2: ROC curves for a quadrature receiver.

5.4.2 Tracker Operating Characteristic — Summary

5.4.2.1 Tracker Operating Characteristic — Summary

The *modified Riccati equation* can be iterated forward for given parameters:

- initial covariance
- target detection probability
- expected number of false measurements in the validation region, which follows from:
 - the volume of the validation region, as determined by the innovation covariance
 - the expected number of false measurements per unit volume, obtained from the signal processor's resolution cell size and false alarm probability.

This iteration, which amounts to *nonsimulation performance prediction*, can be done

- to steady state — if it exists
- one or several steps ahead.

Iteration to steady state has shown the existence of a region of *tracking instability* — heavy clutter and/or low detection probability will definitely lead to loss of track.

Instability is a consequence of the following:

- For a given initial state covariance the resulting measurement validation region has a certain size
- For this size validation region and the given expected number of false measurements per unit volume one has a certain *expected number* of false measurements in the validation region
- To this corresponds a certain *information reduction factor*
- If, in view of the information reduction factor, the next updated covariance is *larger* than the previous one, then there will be *more false measurements in the validation region* and therefore *less information* about the state in the measurements
- This can lead to covariance divergence.

5.4.2 Tracker Operating Characteristic — Summary

If the modified Riccati equation converges to a steady-state value, this will depend on

1. The process and measurement noise covariances
2. The target detection probability
3. The *spatial density* (expected number per unit volume) of the false measurements, which depends on the signal processor's false alarm probability.

For fixed values of the first set of parameters one can plot a *norm of the steady state covariance* as a function of P_D and P_{FA} of the signal receiver — this is the *Tracker Operating Characteristic (TOC)*.

Using the Receiver Operating Characteristic curve for a certain SNR one can find the operating point on it corresponding to the detection threshold that will yield the lowest covariance norm (steady state error) from the TOC.

5.5 A HYBRID ALGORITHM FOR PREDICTION OF TRACKING PERFORMANCE IN CLUTTER

5.5.1 The Hybrid Nature of the Performance Prediction Problem

Tracking performance depends on the following random factors:

- (a) the target motion uncertainty
- (b) the measurement inaccuracies
- (c) the target detection events
- (d) the false detection events.

While the first two factors are *continuous-valued*, the latter are *discrete-valued* — this is a **hybrid system**.

In the previous sections the tracking accuracy has been evaluated for a PDAF by carrying out a **global averaging** on the stochastic Riccati equation to yield an approximate estimation error covariance — a norm of which is the tracking accuracy — leading to the modified Riccati equation.

A more sophisticated **hybrid conditional averaging (HYCA) performance prediction algorithm** — with explicit probabilistic accounting of the *continuous and the discrete uncertainties* — is presented next for a detailed evaluation of

- the **track loss probability**
- the estimation error covariance — in a more accurate manner than the global averaging algorithm.

Track loss is a “runaway” phenomenon — it requires detailed transient evaluation capability, which can be provided only by a hybrid approach.

5.5.2 The Hybrid Approach

The major difference between the global and the hybrid approaches is

- In the global approach *all uncertainties* are averaged out, whereas
- In the hybrid approach *only the continuous-valued uncertainties* are averaged out.

The latter is accomplished by a conditional expectation *conditioned on the discrete-valued uncertainty*, which in the present case is $m(k)$, the number of measurements in the validation region.

The **hybrid performance prediction algorithm** for the PDAF consists of the following steps in each cycle:

1. Starting with the expected values of the updated state covariance *conditioned* on each realization of the number of measurements $m(k)$ in the validation region, $\bar{P}[k-1|k-1, m(k-1)]$, one computes the prediction covariance $\bar{P}[k|k-1, m(k-1)]$
2. Calculation of the expected value of the updated state covariance at k conditioned on $m(k-1)$ and $m(k)$, $\bar{P}[k|k, m(k-1), m(k)]$, using the appropriate information reduction factor
3. Calculation of the transition probabilities of a Markov chain for the transitions from the various values of $m(k-1)$ to the values of $m(k)$
4. A “mixing” of the results from step 2 with the probabilities calculated in step 3, to yield $\bar{P}[k|k, m(k)]$.

Figure 5.5.2-1 illustrates the above sequence of calculations with the circles symbolizing the “magnitude” of the covariance — if there is no measurement in the gate the covariance increases, and it decreases most if the number of measurements is one.

The number of possible validated measurements in the gate (whose size varies according to the innovation covariance, also evaluated within the iterations) is limited to a certain maximum N beyond which the track will be practically lost.

Figure 5.5.2-2 shows the corresponding algorithmic flowchart. Details can be found in [LB91b].

5.5.2 The Hybrid Approach

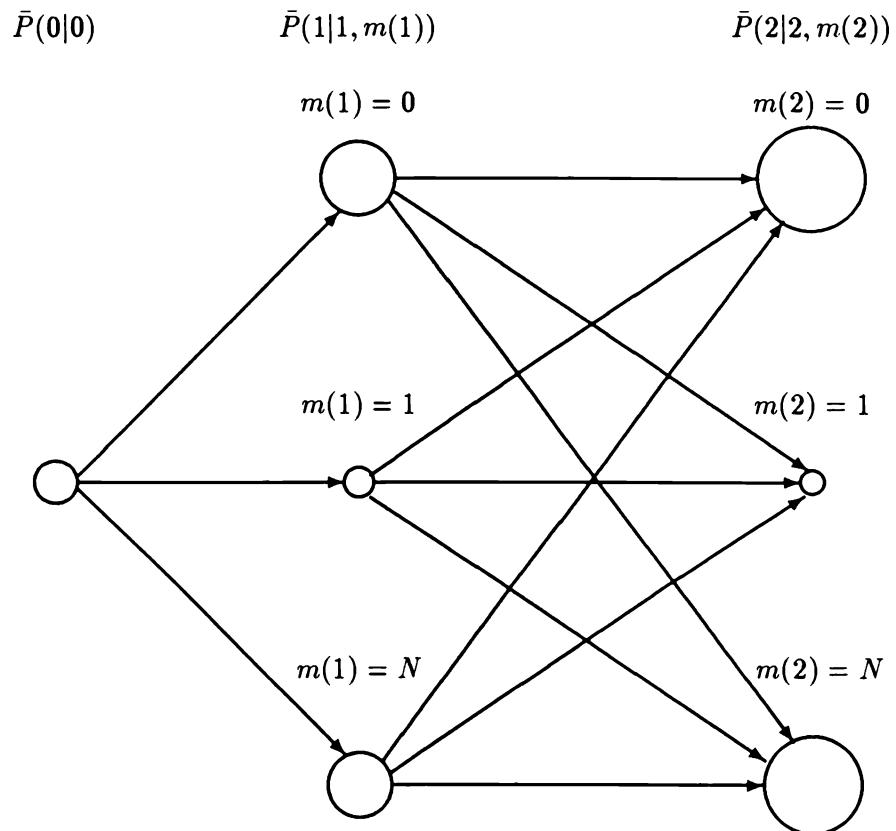


Figure 5.5.2-1: Illustration of the covariance calculation.

5.5.2 The Hybrid Approach

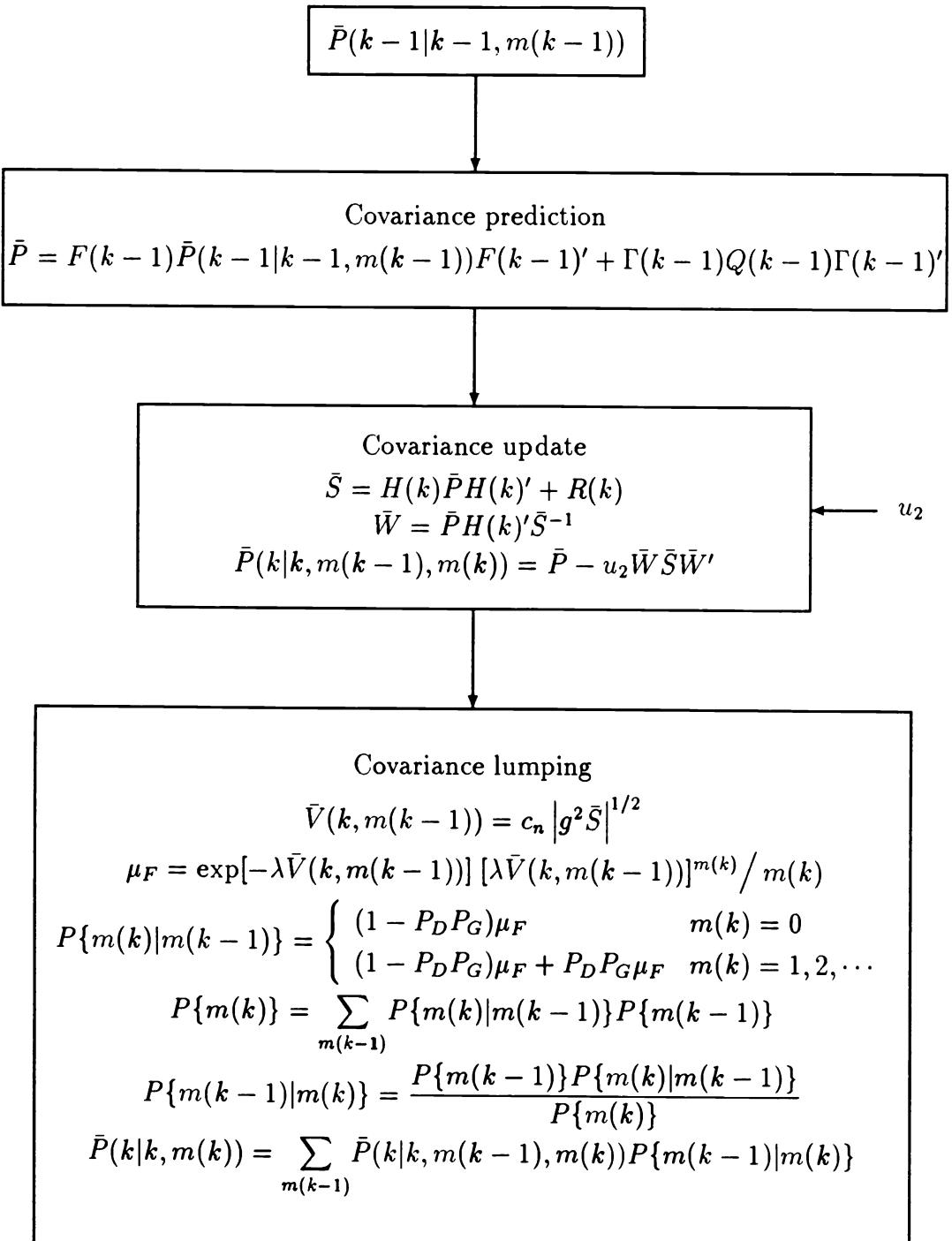


Figure 5.5.2-2: Recursion of $\bar{P}(k|k, m(k))$.

5.5.3 Track Loss

5.5.3.3 Track Loss

The phenomenon of **track loss**, being a gradual process, has no unique definition.

Criterion for Declaring a Track Lost

$$\{\text{number of measurements in the gate} \geq M\} \implies \{\text{track loss}\}$$

While this requires the choice of the threshold M , the result is not very sensitive to it: once the number of measurements is large, it will rapidly increase and diverge.

Track Loss Prediction

1. Probability that track loss will occur *exactly at time k*:

$$P_{tl}(k) \triangleq 1 - \sum_{m(k)=0}^{M-1} P^*\{m(k)\} \quad (5.5.3-1)$$

where the probability that the number of measurements in the gate at k is $m(k)$ conditioned on the event that the track was not lost before k (the threshold M has not been reached before) is

$$P^*\{m(k)\} \triangleq \frac{\sum_{m(k-1)=0}^{M-1} P\{m(k)|m(k-1)\} P^*\{m(k-1)\}}{\sum_{m(k-1)=0}^{M-1} P^*\{m(k-1)\}} \quad (5.5.3-2)$$

2. **Cumulative probability of track loss** in N time steps:

$$\begin{aligned} P_{TL}(N) &\triangleq 1 - \prod_{k=1}^N \left(\sum_{m(k)=0}^{M-1} P^*\{m(k)\} \right) \\ &= P_{TL}(N-1) + [1 - P_{TL}(N-1)] P_{tl}(N) \end{aligned} \quad (5.5.3-3)$$

3. **Average track life** (in sampling periods):

$$\bar{\tau} = \sum_k k[P_{TL}(k+1) - P_{TL}(k)] \quad (5.5.3-4)$$

5.5.4 Example of Track Loss Prediction

5.5.4 Example of Track Loss Prediction

The Scenario

- A target with a certain SNR is tracked by a sensor with measurement noise of variance r normalized to unity
- The sensor's two dimensional resolution cell has area $C = (3\sqrt{r})^2 \approx 10$
- The maneuvering index λ_m of the target is 0.2, i.e., moderate maneuvering capability
- The return signal is processed by a quadrature receiver as in Subsection 5.4.1 with a CFAR setting for $P_{FA} = 0.1$ per cell, which yields a false alarm spatial density of $\lambda_F = P_{FA}/C = 0.01$.

Results

Table 5.5.4-1 shows the values of the target P_D for SNR = 3dB (Case 1) as well as for 6dB (Case 2) and 9dB (Case 3).

	Case 1	Case 2	Case 3
SNR (dB)	3	6	9
P_D	0.54	0.78	0.97

Table 5.5.4-1: The three cases with CFAR.

Figure 5.5.4-1 shows the track loss probabilities P_{TL} (with the PDAF as the tracking filter) for these cases obtained using the hybrid performance prediction algorithm. It should be noted that the threshold settings, chosen for the indicated CFAR, are not optimal.

For the purpose of illustration of the accuracy of this hybrid performance predictor, Figure 5.5.4-1 also shows the results of 100 Monte Carlo runs for Case 1 — quite close to the prediction.

5.5.4 Example of Track Loss Prediction

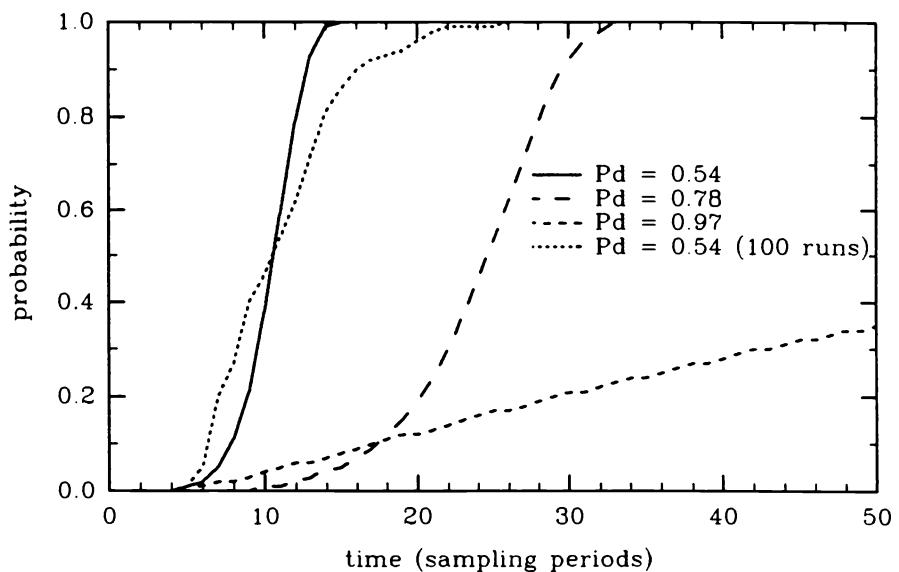


Figure 5.5.4-1: Track loss probability for $\lambda_F = 0.01$, $\lambda_m = 0.2$: — $P_D = 0.54$; - - $P_D = 0.78$; - · - $P_D = 0.97$; ··· $P_D = 0.54$ (from 100 Monte Carlo runs).

5.5.5 Detection Threshold Optimization

5.5.5.5 Detection Threshold Optimization

A Swerling I target fluctuation model was considered.

Table 5.5.5-1 presents the values of P_D for various values of the detection threshold setting which are indicated in terms of the resulting P_{FA} .

P_{FA}	0.1	0.05	0.02	0.01	0.005	0.002	0.001
SNR (dB)							
3	0.46	0.37	0.27	0.21	0.17	0.13	
6	0.63	0.55	0.46	0.40	0.35	0.29	0.25
9	0.77	0.72	0.65	0.60	0.55	0.50	0.46
12	0.87	0.84	0.79	0.76	0.73	0.69	0.66
15	0.93	0.91	0.89	0.87	0.85	0.83	0.81

Table 5.5.5-1: Swerling I target P_D for various threshold settings and SNR.

The **average track life** for the above cases, obtained using the hybrid performance predictor is presented in Table 5.5.5-2.

P_{FA}	0.1	0.05	0.02	0.01	0.005	0.002	0.001
SNR (dB)							
3	*	*	*	*	*	*	*
6	*	2.5	6.2	17	>500	>500	
9	*	3	>500	>500	>500	>500	>500
12	1.3	≈ 100	>500	>500	>500	>500	>500
15	1.4			>500	>500	>500	>500

Table 5.5.5-2: Swerling I target average track life for various threshold settings and SNR (* negligible).

5.5.5 Detection Threshold Optimization

Table 5.5.5-3 presents the *position estimation RMS errors obtained by the hybrid performance predictor* for various threshold settings. The optimal settings (within the grid of values considered) are indicated by bold values for the corresponding minimal RMS errors.

P_{FA}	0.1	0.05	0.02	0.01	0.005	0.002	0.001
SNR (dB)							
3	*	*	*	*	*	*	*
6	*	*	*	*	2.33	2.86	
9	*	*	1.54	1.50	1.56	1.67	1.80
12	*	1.56	1.26	1.25	1.25	1.28	1.31
15	1.87			1.11	1.11	1.10	1.12

Table 5.5.5-3: Swerling I target average position estimation error for various threshold settings and SNR (* unstable).

The optimal settings are not CFAR, but the CFAR setting of $P_{FA} = 0.005$ will yield near optimal results.

5.5.6 Hybrid Performance Prediction — Summary

The *hybrid performance prediction algorithm* for the PDAF, in comparison with the *averaging algorithm* is

- more versatile — it yields
 - tracking accuracy, as well as
 - track life
- more accurate.

This is due to the fact that, by its hybrid nature, it accounts explicitly for the

- continuous, as well as
- discrete

uncertainties inherent in the tracking in clutter problem.

The applications shown for the hybrid predictor are

- track life evaluation for various SNR situations
- tracking accuracy evaluation and detection threshold selection for the optimization of the tracking accuracy.

5.6 BIBLIOGRAPHICAL NOTES

Sections 5.2–5.4 are based on [FBSG81] and [FBSG85]. Further work on this method was presented in [GFB84] where the criterion is a one-step-ahead error minimization of the error in the PDAF. Extensive discussion of detection models for radar (and sonar) can be found in [DR80].

A simple one-step-ahead error minimization criterion (assuming a nearest-neighbor approach) for adaptive threshold setting was proposed in [MWI80] and can also be found in [Bla86].

Section 5.5 is based on [LB91b, LB94b]. A description of the hybrid performance prediction technique in a general setting can be found in [LB94c, LB94a].

Chapter 6

MULTIPLE TARGETS IN CLUTTER: BAYESIAN APPROACHES

6.1 INTRODUCTION

6.1.1 Outline

As discussed in Section 2.4, the association of measurements in a multitarget environment has to be done by considering *simultaneously* all the targets: when a measurement could have originated from more than one target, these targets cannot be considered separately in the association problem.

Section 6.2 presents the ***Joint Probabilistic Data Association Filter (JPDAF)***. This method extends the PDAF from Section 3.4 to a *known number of targets* whose tracks have been established and evaluates the measurement-to-target association probabilities for the *latest set of measurements* and then combines them into the state estimates: it is a *target-oriented approach*.

Extending the above mentioned observation that one has to consider simultaneously all the targets to also considering simultaneously all the scans, yields the most complex algorithm, the Multiple Hypothesis Tracker.

6.1.1 Outline

The **Multiple Hypothesis Tracker (MHT)** in Section 6.3 evaluates the probabilities that there is a target from which a *sequence of measurements* originated. This is done for *each measurement sequence* — each hypothesis is considered. The MHT does not assume a known number of targets — it is a *measurement-oriented approach* — and it also has track initiation capability.

When two targets are close, due to a sensor's inherent **resolution** limitation, there might be a single detection originating from both, yielding a merged (unresolved) measurement. Section 6.4 extends the JPDA to the **JPDAM** with a model for the **merged measurements**.

6.1.2 Summary of Objectives

6.1.2 Summary of Objectives

Present the Bayesian approaches — with prior probabilistic models about the targets

- Target-oriented approach: Joint Probabilistic Data Association Filter (JPDAF) — for a known number of targets it evaluates the measurement-to-target association probabilities (for the latest set of measurements) and combines them into the corresponding state estimates
- Measurement-oriented approach: Multiple Hypothesis Tracker (MHT) — evaluates the probabilities that there is a target from which a sequence of measurements originated.

Extend the JPDA to handle merged (unresolved) measurements.

6.2 THE JOINT PROBABILISTIC DATA ASSOCIATION FILTER

6.2.1 Introduction

The **Joint Probabilistic Data Association (JPDA)** approach is the extension of the PDA method of Section 3.4.

Assumptions

- There is a known number of established targets in clutter
- Measurements from one target can fall in the validation region of a neighboring target — this can happen over several sampling times and acts as a *persistent interference*
- The past is summarized by an approximate sufficient statistic — state estimates (approximate conditional mean) and covariances for each target
- The states are assumed Gaussian distributed with the above means and covariances
- Each target has a dynamic and a measurement model as in (3.1.3-1), (3.1.3-2). The models for the various targets do not have to be the same.

The PDAF modeled all the incorrect measurements as *random interference*, with uniform spatial distribution. The performance of the PDAF degrades significantly when a neighboring target gives rise to persistent interference.

The Approach

- The measurement-to-target association probabilities are computed *jointly across the targets*.
- The association probabilities are computed only for the latest set of measurements — this is a non-backscan approach.
- The state estimation is done
 - separately for each target as in the PDAF (decoupled) — JPDAF
 - in a coupled manner using a stacked state vector — JPDACF.

The Key Feature of the JPDA

The evaluation of the conditional probabilities of the following **joint association events** pertaining to the current time k (the time index k is omitted for simplicity where it does not cause confusion) is carried out:

$$\theta = \bigcap_{j=1}^m \theta_{jt}, \quad (6.2.1-1)$$

where

- θ_{jt} is the event that measurement j originated from target t , $j = 1, \dots, m; t = 0, 1, \dots, N_T$
- t_j is the index of the target to which measurement j is associated in the event under consideration
- N_T is the known number of targets.

Remark

For the purpose of deriving the joint probabilities, no individual validation gates will be assumed for the various targets. Instead, each measurement will be assumed validated for each target, i.e., every validation gate coincides with the entire surveillance region.

This approach is adopted in order to have the pdf of each false measurement the *same*, i.e., uniformly distributed in the *entire validation region*.

The good news is that this yields simpler expressions for the probabilities of the events (6.2.1-1); the bad news is that this leads to the extra computational burden of considering events with negligible probability. This burden can, however, be easily avoided by using a logic that considers only the joint events made up of marginal events involving validated measurements.

6.2.2 The Feasible Joint Events

6.2.2 The Feasible Joint Events

Validation gates are used for the selection of the **feasible joint events** but not in the evaluation of their probabilities. This logic avoids considering events whose probabilities are negligible and thus has a negligible effect on the other probabilities.

The Validation Matrix

Define the **validation matrix**

$$\Omega = [\omega_{jt}] \quad j = 1, \dots, m; \quad t = 0, 1, \dots, N_T \quad (6.2.2-1)$$

with binary elements that indicate if measurement j lies in the validation gate of target t . The index $t = 0$ stands for “none of the targets” and the corresponding column of Ω has all units since each measurement could have originated from clutter or false alarm.

An example of a validation matrix for two targets is

$$\Omega = \begin{matrix} & \begin{matrix} t & 0 & 1 & 2 \end{matrix} \\ \begin{matrix} j \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{matrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 \\ 3 & 1 & 0 & 1 \\ 4 & 1 & 0 & 1 \end{matrix} \end{matrix} \quad (6.2.2-2)$$

This corresponds to the situation depicted in Figure 2.4.2-1. The two targets in this example are “coupled” by measurement $j = 2$ which has been validated for both targets — it lies in the intersection of the two validation regions.

The Event Matrix

A **joint association event** θ is represented by the **event matrix**

$$\hat{\Omega}(\theta) = [\hat{\omega}_{jt}(\theta)] \quad (6.2.2-3)$$

consisting of the units in Ω corresponding to the associations in θ ,

$$\hat{\omega}_{jt}(\theta) = \begin{cases} 1 & \text{if } \theta_{jt} \in \theta \\ 0 & \text{otherwise} \end{cases} \quad (6.2.2-4)$$

6.2.2 The Feasible Joint Events

A **feasible association event** is one where

- (i) a measurement can have only one source, i.e.,

$$\sum_{t=0}^{N_T} \hat{\omega}_{jt}(\theta) = 1 \quad \forall j \quad (6.2.2-5)$$

- (ii) at most one measurement can originate from a target

$$\delta_t(\theta) \triangleq \sum_{j=1}^m \hat{\omega}_{jt}(\theta) \leq 1 \quad t = 1, \dots, N_T \quad (6.2.2-6)$$

Generation of the Feasible Joint Association Events

The generation of the event matrices $\hat{\Omega}$ corresponding to feasible events can be done by scanning Ω and picking

- (i) one unit per row, and
- (ii) one unit per column except for $t = 0$ where the number of units (which is the number of false measurements) is not restricted.

The binary variable $\delta_t(\theta)$ defined in (6.2.2-6) is called **target detection indicator** since it indicates whether a measurement is associated with target t in event θ , i.e., whether it has been detected.

It is also convenient to define another binary variable, called **measurement association indicator**

$$\tau_j(\theta) \triangleq \sum_{t=1}^{N_T} \hat{\omega}_{jt}(\theta) \quad (6.2.2-7)$$

to indicate if measurement j is associated with a target in event θ .

With this definition, the number of false (unassociated) measurements in event θ is

$$\phi(\theta) = \sum_{j=1}^m [1 - \tau_j(\theta)] \quad (6.2.2-8)$$

6.2.3 Evaluation of the Joint Probabilities

6.2.3 Evaluation of the Joint Probabilities

The **joint association event probabilities** are, with Bayes' formula,

$$\begin{aligned} P\{\theta(k)|Z^k\} &= P\{\theta(k)|Z(k), m(k), Z^{k-1}\} = \frac{1}{c} p[Z(k)|\theta(k), m(k), Z^{k-1}] P\{\theta(k)|Z^{k-1}, m(k)\} \\ &= \frac{1}{c} p[Z(k)|\theta(k), m(k), Z^{k-1}] P\{\theta(k)|m(k)\} \end{aligned} \quad (6.2.3-1)$$

where c is the normalization constant. The irrelevant conditioning term has been omitted in last line of the above equation.

Assumption

The states of the targets conditioned on the past observations are mutually *independent*.

The Likelihood Function of the Measurements

The **likelihood function of the measurements** on the r.h.s. of (6.2.3-1) is

$$p[Z(k)|\theta(k), m(k), Z^{k-1}] = \prod_{j=1}^{m(k)} p[z_j(k)|\theta_{jt_j}(k), Z^{k-1}] \quad (6.2.3-2)$$

where $m(k)$ is the number of measurements in the union of the validation regions at time k . The product form of (6.2.3-2) follows from the above assumption.

The conditional pdf of a measurement given its origin is

$$p[z_j(k)|\theta_{jt_j}(k), Z^{k-1}] = \begin{cases} f_{t_j}[z_j(k)] & \text{if } \tau_j[\theta(k)] = 1 \\ V^{-1} & \text{if } \tau_j[\theta(k)] = 0 \end{cases} \quad (6.2.3-3)$$

where

$$f_{t_j}[z_j(k)] = \mathcal{N}[z_j(k); \hat{z}^{t_j}(k|k-1), S^{t_j}(k)] \quad (6.2.3-4)$$

and $\hat{z}^{t_j}(k|k-1)$ is the predicted measurement for target t_j , with associated innovation covariance $S^{t_j}(k)$.

Measurements not associated with a target are assumed uniformly distributed in the surveillance region of volume V .

Using (6.2.3-3), the pdf (6.2.3-2) can be written as follows

$$p[Z(k)|\theta(k), m(k), Z^{k-1}] = V^{-\phi} \prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j} \quad (6.2.3-5)$$

In the above V^{-1} is raised to power $\phi(\theta)$, the total number of false measurements in event $\theta(k)$ and the indicators $\tau_j(\theta)$ select the single measurement densities according to their associations in event $\theta(k)$.

6.2.3 Evaluation of the Joint Probabilities

The Prior Probability of a Joint Association Event

The prior (to time k) probability of an event $\theta(k)$, the last term in (6.2.3-1), is obtained next. Denote by $\delta(\theta)$ the vector of target detection indicators (6.2.2-6) corresponding to event $\theta(k)$. Note that, given θ , the vector $\delta(\theta)$ is completely defined and so is the number ϕ of false measurements given by (6.2.2-8). Therefore

$$P\{\theta(k)|m(k)\} = P\{\theta(k), \delta(\theta), \phi(\theta)|m(k)\} \quad (6.2.3-6)$$

The above joint probability can be rewritten as

$$P\{\theta(k)|m(k)\} = P\{\theta(k)|\delta(\theta), \phi(\theta), m(k)\} P\{\delta(\theta), \phi(\theta)|m(k)\} \quad (6.2.3-7)$$

The first term on the r.h.s. of the above is obtained from the following reasoning based on **combinatorics**:

- (i) In event $\theta(k)$ the set of targets assumed detected consists of $m(k) - \phi$ targets.
- (ii) The number of measurement-to-target assignment events $\theta(k)$ in which the same set of targets is detected is given by the number of *permutations* of the $m(k)$ measurements taken as $m(k) - \phi$, the number of targets to which a measurement is assigned under the same detection event.

Therefore, assuming each such event a priori equally likely, one has

$$P\{\theta(k)|\delta(\theta), \phi(\theta), m(k)\} = \left(P_{m(k)-\phi}^{m(k)} \right)^{-1} = \left(\frac{m(k)!}{\phi!} \right)^{-1} \quad (6.2.3-8)$$

The last term in (6.2.3-7) is, assuming δ and ϕ independent,

$$P\{\delta(\theta), \phi(\theta)|m(k)\} = \prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t} \mu_F(\phi) \quad (6.2.3-9)$$

where P_D^t is the detection probability of target t and $\mu_F(\phi)$ is the prior pmf of the number of false measurements (the clutter model). The indicators $\delta_t(\theta)$ have been used in (6.2.3-9) to select the probabilities of detection and no detection events according to the event $\theta(k)$ under consideration.

Combining (6.2.3-8) and (6.2.3-9) into (6.2.3-7) yields the **prior probability of a joint association event** $\theta(k)$ as

$$P\{\theta(k)|m(k)\} = \frac{\phi!}{m(k)!} \mu_F(\phi) \prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t} \quad (6.2.3-10)$$

6.2.3 Evaluation of the Joint Probabilities

The Posterior Probability of a Joint Association Event

Combining (6.2.3-5) and (6.2.3-10) into (6.2.3-1) yields the ***posterior probability of a joint association event*** $\theta(k)$ as

$$P\{\theta(k)|Z^k\} = \frac{1}{c m(k)!} \frac{\phi!}{\mu_F(\phi)} V^{-\phi} \prod_j \{f_t, [z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t} \quad (6.2.3-11)$$

where ϕ , δ_t and τ_j are all functions of the event $\theta(k)$ under consideration.

The above still needs the specification of the pmf of the number of false measurements $\mu_F(\phi)$, carried out in the next subsection.

6.2.4 The Parametric and Nonparametric JPDA

6.2.4 The Parametric and Nonparametric JPDA

As in the case of the PDA, the JPDA has two versions, according to the model used for the pmf $\mu_F(\phi)$ of the number of false measurements.

The Parametric JPDA

The **parametric JPDA** uses the Poisson pmf

$$\mu_F(\phi) = e^{-\lambda V} \frac{(\lambda V)^\phi}{\phi!} \quad (6.2.4-1)$$

which requires the spatial density λ of the false measurements.

Using (6.2.4-1) in (6.2.3-11) leads to the cancellation of V^ϕ and $\phi!$. Furthermore, each term contains $e^{-\lambda V}$ and $m(k)!$, which also cancel since they appear in the denominator c of (6.2.3-11), which is the sum of all the numerators.

Thus the joint association probabilities of the parametric JPDA are

$$P\{\theta(k)|Z^k\} = \frac{\lambda^\phi}{c_1} \prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1 - P_D)^{1 - \delta_t} \quad (6.2.4-2)$$

where c_1 is the appropriate normalization constant.

Since $m(k)$ is a fixed number, define a new normalization constant

$$c_2 \triangleq c_1 \lambda^{-m(k)} \quad (6.2.4-3)$$

With this, the expression (6.2.4-2) of the joint association probabilities can be rewritten as

$$P\{\theta(k)|Z^k\} = \frac{1}{c_2} \prod_j \{\lambda^{-1} f_{t_j}[z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1 - P_D)^{1 - \delta_t} \quad (6.2.4-4)$$

6.2.4 The Parametric and Nonparametric JPDA

The Nonparametric JPDA

The **nonparametric JPDA** uses the **diffuse prior** (3.4.3-7)

$$\mu_F(\phi) = \epsilon \quad \forall \phi \quad (6.2.4-5)$$

which does not require the parameter λ .

With this, (6.2.3-11) becomes after cancelling the constant ϵ and $m(k)!$, which appear in each expression,

$$P\{\theta(k)|Z^k\} = \frac{1}{c_3} \frac{\phi!}{V^\phi} \prod_j \{f_t, [z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1 - P_D)^{1-\delta_t} \quad (6.2.4-6)$$

where c_3 is the appropriate normalization constant.

Similarly to the nonparametric case, with

$$c_4 \triangleq c_3 V^{-m(k)} \quad (6.2.4-7)$$

the expression (6.2.4-6) of the joint association probabilities can be rewritten as

$$P\{\theta(k)|Z^k\} = \frac{1}{c_4} \phi! \prod_j \{V f_t, [z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1 - P_D)^{1-\delta_t} \quad (6.2.4-8)$$

Remark

As it can be seen from (6.2.4-6), the **nonparametric JPDA** expressions contain a term that can be called **pseudo sample spatial measurement density** $\phi!/V^\phi$ in place of λ^ϕ in the **parametric JPDA**.

This is analog with the PDA where the nonparametric version contains $m(k)/V$ in place of λ from the parametric version.

6.2.5 The State Estimation

The state estimation (filtering) algorithm can be carried out in two ways.

Assumption

The states of the targets conditioned on the past observations are mutually *independent*. This is the same as assumed in Subsection 6.2.3 for the JPDA.

Then one has

- **Decoupled estimation** for the targets under consideration — the **Joint Probabilistic Data Association Filter (JPDAF)**.

In this case one needs the **marginal association probabilities**, which are obtained from the joint probabilities by summing over all the joint events in which the marginal event of interest occurs. Using definition (6.2.2-4) this summation can be written as follows

$$\begin{aligned}
 \beta_{jt} &\triangleq P\{\theta_{jt}|Z^k\} \\
 &= \sum_{\theta} P\{\theta|Z^k\} \hat{\omega}_{jt}(\theta) \\
 &= \sum_{\theta: \theta_{jt} \in \theta} P\{\theta|Z^k\}
 \end{aligned} \tag{6.2.5-1}$$

The state estimation equations are exactly the same as in the PDAF, presented in Subsection 3.4.2, and will not be repeated.

Alternative Assumption

Considering the targets' states, given the past, as *correlated* — characterized by means, covariances as well as cross-covariances — leads to

- **coupled estimation** for the targets under consideration — the **JPDA Coupled Filter (JPDACF)**.

This is discussed in Subsection 6.2.7.

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

This example considers a multitarget tracking problem in the presence of clutter with measurements obtained from several sensors. The targets under consideration were assumed to follow a nominal straight-line trajectory with random disturbances (process noise) and radiate acoustic energy in one or more characteristic narrow or wide bands. Since much of the acoustic energy is produced by the propulsion system the corresponding frequencies (narrow band emissions) are coupled with the target's speed.

The Target State Model

The relationships between the state components are

$$\dot{L} = \frac{V \cos C}{60} + v_L \quad (\text{latitude}) \quad (6.2.6-1)$$

$$\dot{M} = \frac{V \sin C}{60 \cos L} + v_m \quad (\text{longitude}) \quad (6.2.6-2)$$

$$\dot{C} = 0 + v_C \quad (\text{course}) \quad (6.2.6-3)$$

$$\dot{V} = -\frac{1}{\tau} V + \frac{1}{\tau} F_1 K + v_V \quad (\text{speed}) \quad (6.2.6-4)$$

$$\dot{K} = 0 + v_K \quad (\text{speed-frequency coupling}) \quad (6.2.6-5)$$

$$\dot{F}_1 = 0 + v_{F_1} \quad (\text{propulsion frequency}) \quad (6.2.6-6)$$

$$\dot{F}_j = 0 + v_{F_j} \quad (\text{other frequencies}) \quad (6.2.6-7)$$

More concisely, the state equation can be written as

$$\dot{x} = f[x(t)] + v(t) \quad (6.2.6-8)$$

where the process noise $v(t)$ is assumed zero-mean, white, and Gaussian.

The speed-frequency (or geographic-acoustic) coupling appears in (6.2.6-4) via the so-called **knots/Hertz ratio** K and the time constant τ .

Estimation of K can aid both tracking and classification while estimation of the source frequencies (narrow band emissions) has proved to be particularly valuable for decreasing the uncertainties in the data association.

In general, the inclusion of **target features** such as F_j , K or other variables in the state vector, can be very helpful in tracking and classification of targets.

The Measurements

The set of passive sensors — **acoustic arrays** — considered provide

- (i) single sensor measurements consisting of bearing and frequency lines from the **narrow band target emission** with **Doppler shift**
- (ii) multisensor measurements — **wide band correlation** — consisting of **time difference of arrival (TDOA)** and **Doppler differences (FDOA — frequency difference of arrival)** between pairs of arrays.

The single sensor measurements (i) at sensor n have the form

$$z_n(k) = \begin{bmatrix} \beta_n[L(k), M(k)] \\ hF_j(k) \left\{ 1 + \frac{V(k)}{c_s} \cos \alpha_n[L(k), M(k), C(k)] \right\} \end{bmatrix} + w_n(k) \quad (6.2.6-9)$$

where the target's bearing $\beta_n(\cdot)$ and aspect angle $\alpha_n(\cdot)$ w.r.t. sensor n are standard trigonometric functions, F_j is one of the source frequencies, h is a harmonic number and c_s is the speed of the sound (assumed known and constant).

The measurement noise $w_n(k)$ is assumed zero-mean white and Gaussian.

Other signal attributes (“target features”) such as signal-to-noise ratio and bandwidth are usually measured but are not included in this model.

Using **cross-correlation** of the data streams from two sensors (n and m) with variable shifts in time and frequency one can determine the time delay (TDOA) and Doppler shift (FDOA) between the signals received.

The multisensor measurements (ii) from a pair of sensors n and m are

$$z_{nm}(k) = \begin{bmatrix} \frac{1}{c_s} \{ r_n[L(k), M(k)] - r_m[L(k), M(k)] \} \\ hF_j(k) \frac{V(k)}{c_s} \{ \cos \alpha_n[L(k), M(k), C(k)] - \cos \alpha_m[L(k), M(k), C(k)] \} \end{bmatrix} + w_{nm}(k) \quad (6.2.6-10)$$

where $r_n(\cdot)$ is the target's range to sensor n .

The plant equations with six states (6.2.6-1)–(6.2.6-6) were discretized and linearized and an EKF was run on the resulting equations.

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

The target-sensor geometry is shown in Figure 6.2.6-1. The targets travel at 6 knots on courses of 100° and 80° , respectively, and cross midway through the 6 hour period shown. The crossing of their trajectories and their common source frequency (12 Hz) result in a severe interference.

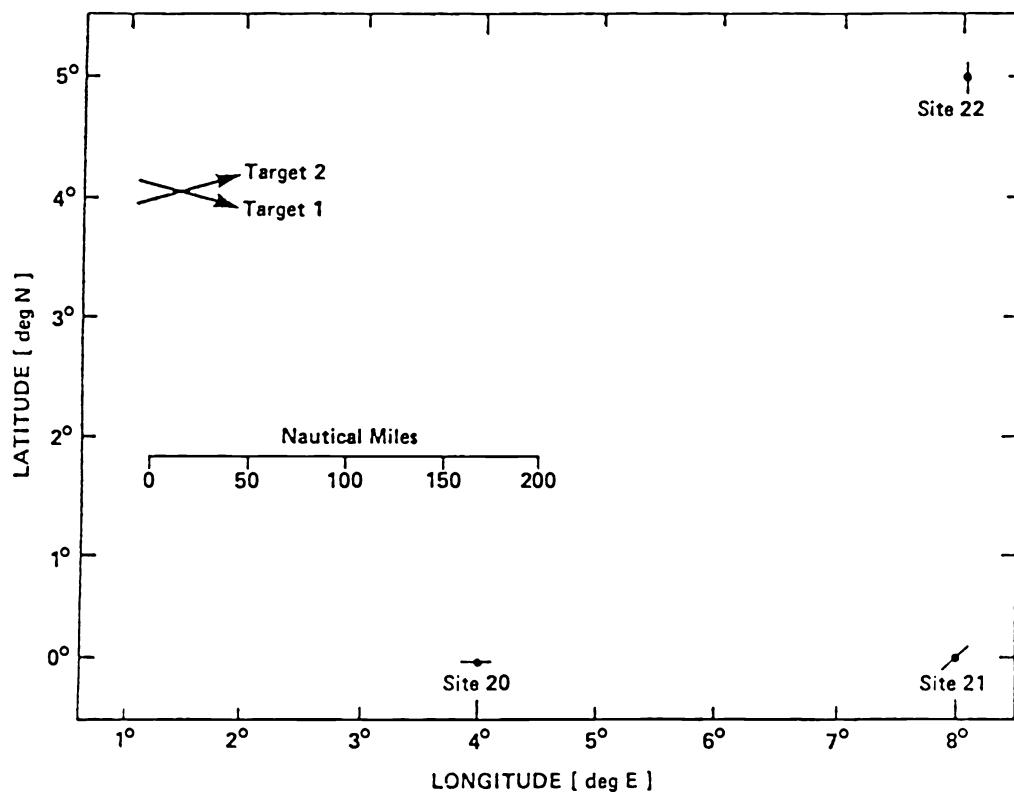


Figure 6.2.6-1: Target/sensor geometry.

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

The measurement data were generated by dead reckoning target motion (no process noise) and adding noise to the computed true measurements, and then adding clutter measurements (false alarms). The number of sampling times was 71 over the 6 hour tracking period.

The measurement noise standard deviations were 5° for bearing, 80 mHz for frequency, 3.6s for TDOA and 4 mHz for FDOA.

The true measurements were detected with probability $P_D = 0.7$, independently over time.

The number of clutter measurements was Poisson distributed and their location uniformly distributed in the measurement space over a very broad region about the actual track. The clutter densities were

- (i) $\lambda_1 = 0.25/(\text{degree}\cdot\text{Hz})$ in the bearing-frequency space
- (ii) $\lambda_2 = 0.25/(s\cdot\text{Hz})$ in the TDOA-FDOA space.

With varying validation gate sizes this yielded an expected number of 0.2 to 2 false detections per gate $\gamma = 16$.

The initial position uncertainty for both targets was taken with standard deviations of 2.5 nmi in each of the two coordinates and uncorrelated. The course and speed standard deviations were taken as 10° and 1.5 kt, respectively. In practice one would use somewhat looser limits to initialize uncertain targets, but the object here was to simulate a situation in which the tracks are already well established before they intersect.

Although the true tracks had no process noise, the filter was given process noise standard deviations of 0.2 kt in speed, 0.2° in course and 0.01 Hz in source frequency to avoid the state estimation covariance to converge to zero.

The measurements from the various sensors were incorporated into the tracking filter sequentially, as discussed in Subsection 2.2.2.

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

Results

Figure 6.2.6-2 shows the bearing measurements from one of the targets at the three sensors with the corresponding validation regions. The thickness of the lines representing the measurements indicates (roughly) the magnitude of the association probabilities.

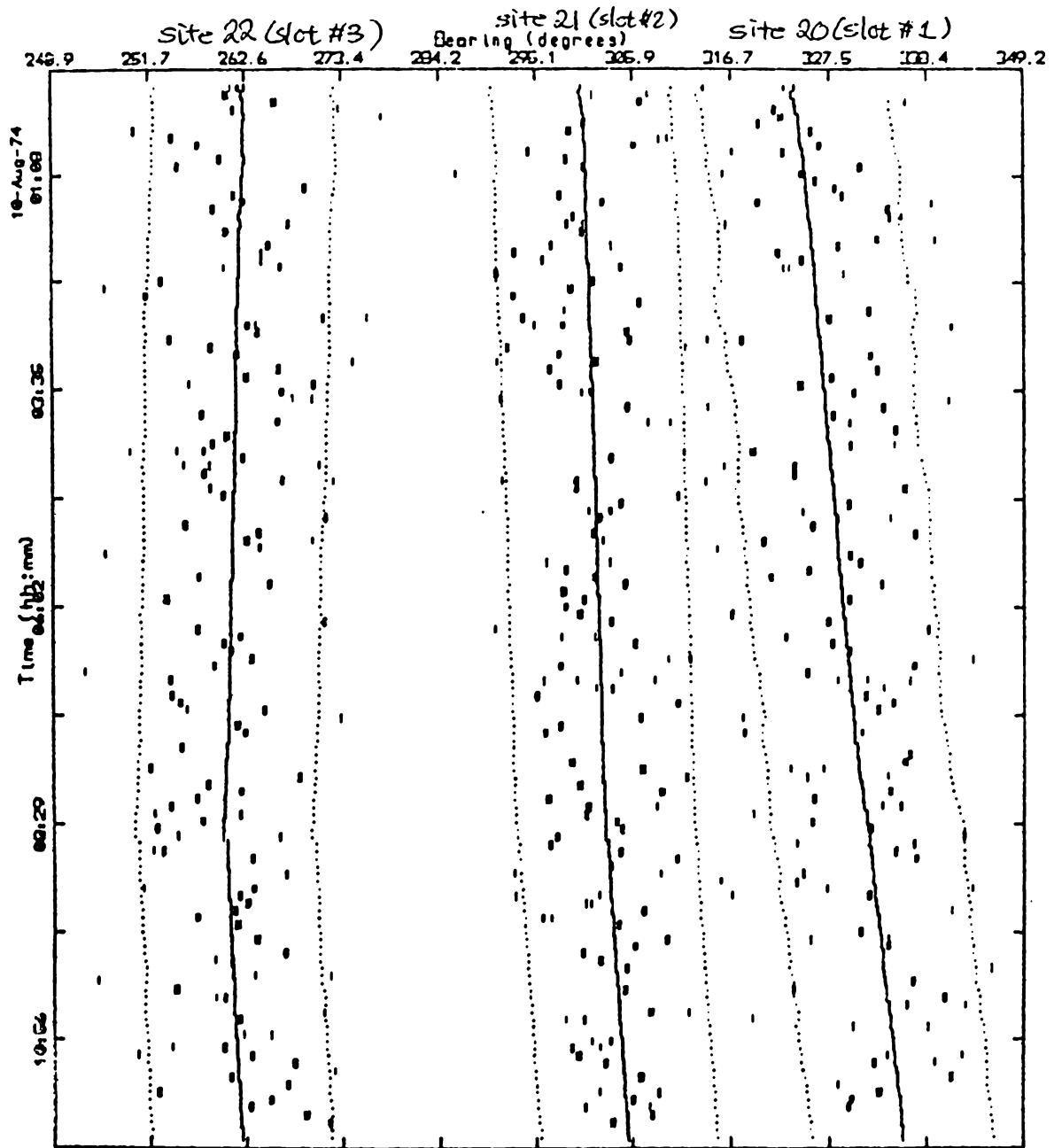


Figure 6.2.6-2: The bearing measurements from the three sensors.

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

Figure 6.2.6-3 shows the track estimates obtained using the nearest-neighbor standard filter (NNSF), described in Section 3.2. Two such filters operated on the corresponding tracks without accounting neither for the clutter (the *random interference*) nor for the other target (the *persistent interference*) — the candidate measurement closest to the center of each validation gate was accepted as correct. The covariance equation did not account for the possibility of association errors. The ellipses represent the filter-calculated “ 2σ ” ($\gamma = 4$) confidence regions for the target position.

The interference resulted in the two NNSF tracks getting merged and the position errors are unacceptably large — both targets’ true positions are far from the ellipses by the end of the 6 hour tracking period.

Figure 6.2.6-4 shows how the PDA filter described in Section 3.4 performed in this case. Each target was tracked separately by a PDAF which modeled the extraneous measurements as clutter, i.e., random interference. Since a significant part of the extraneous measurements were due to the neighboring target, which acted as a persistent interference, the performance is poor. The two tracks became overlapped into a “compromise.”

This shows that the computation of the association probabilities separately for each target is not effective in the presence of a neighboring target.

The JPDA computes the association probabilities jointly across the targets and, as shown in Figure 6.2.6-5, this improves the situation dramatically. The 2-sigma confidence ellipses contain the true positions in all cases.

Comparing to Figure 6.2.6-4, it can be seen that the ellipses are larger, particularly near the point of intersection, where the presence of a persistent interference from the neighboring target increases the uncertainty.

The association probabilities of a measurement belonging to the correct target vs. the other one, as computed by the JPDA, typically have ratios of 0.8/0.2, or greater, early in the run. As the targets draw nearer each other, this ratio becomes about 0.5/0.5, which reflects more uncertainty, and the uncertainty ellipses increase accordingly; then they switch back again when the targets separate.

Finally, tracks made using perfect data association were obtained to provide an (unachievable) lower bound on the estimation performance. As can be seen from Figure 6.2.6-6, the uncertainty ellipses are smaller than with the JPDA, which did not have access to perfect origin information, but the difference is not large.

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

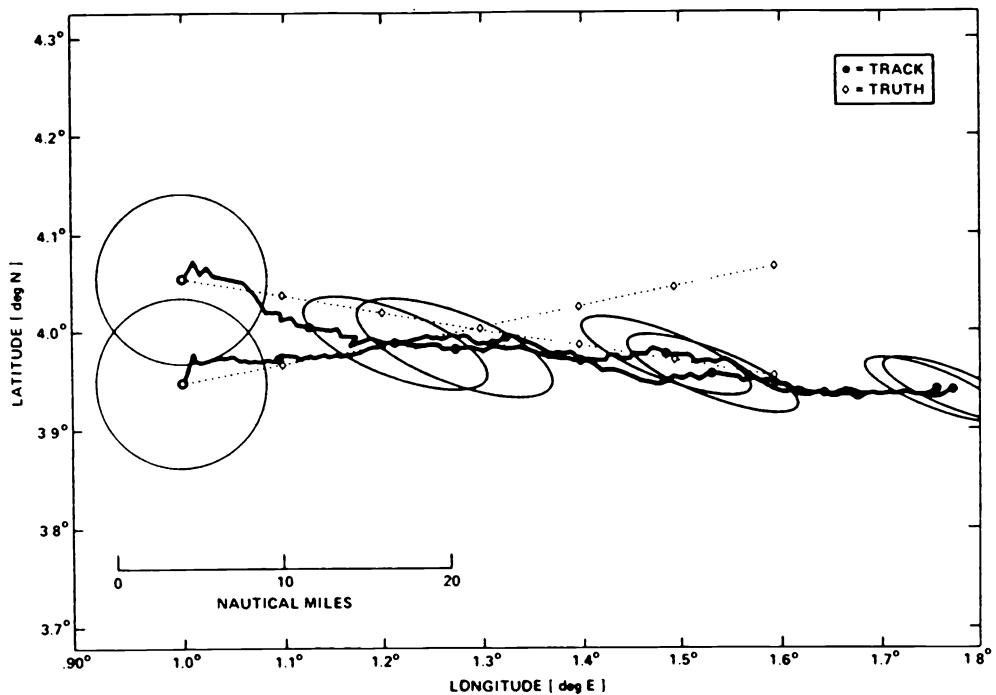


Figure 6.2.6-3: Tracking crossing targets with the NNSF.

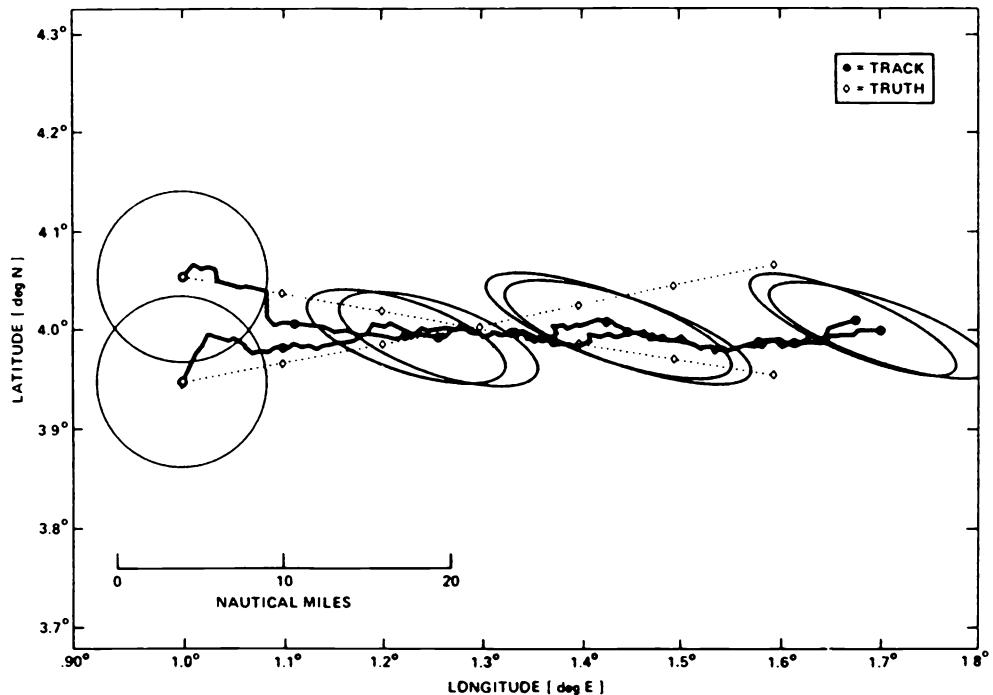


Figure 6.2.6-4: Tracking crossing targets with separate PDA filters.

6.2.6 Passive Sonar Tracking of Two Crossing Targets in Clutter

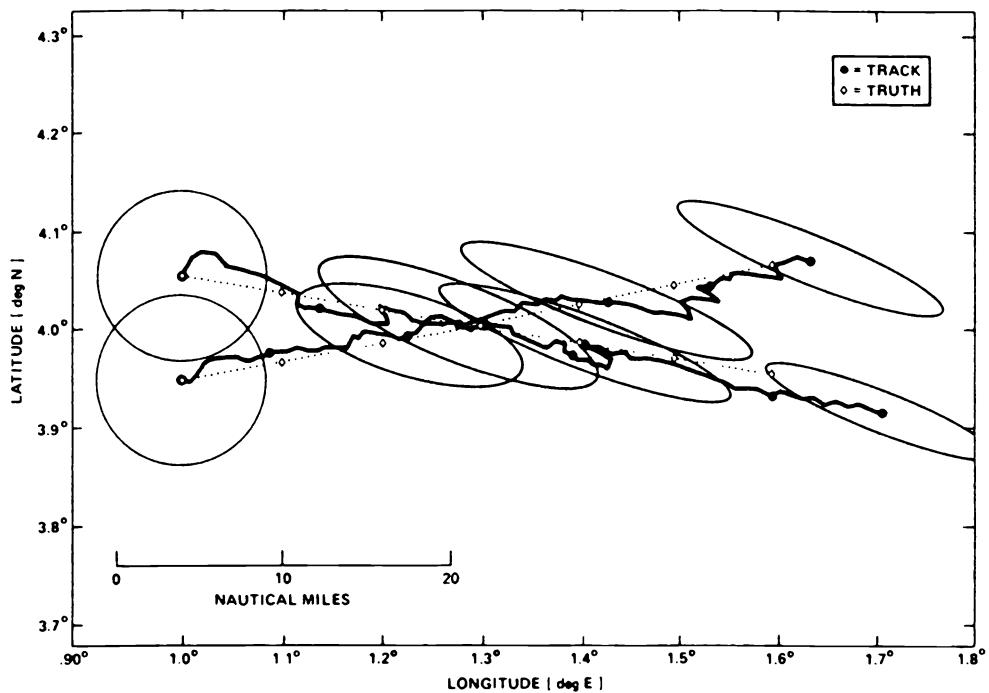


Figure 6.2.6-5: Tracking crossing targets with the JPDAF.

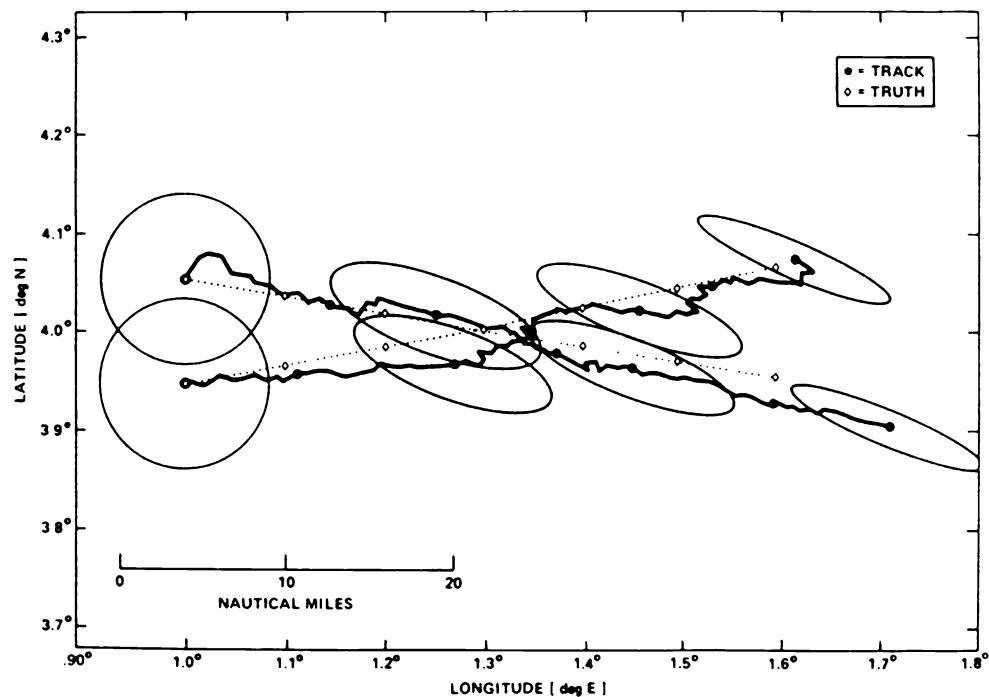


Figure 6.2.6-6: Tracking crossing targets with perfect data association.

6.2.7 A Modification of the JPDAF: Coupled Filtering

6.2.7 A Modification of the JPDAF: Coupled Filtering

In Subsections 6.2.3 and 6.2.4 the JPDA was developed assuming that, conditioned on the past, the target states (and, thus, the target-originated measurements) are *independently distributed*. Consequently, the joint association was followed by *decoupled* filtering of the targets' states — this is an approximation that simplifies the resulting algorithm.

Approach to JPDA with Coupling

For targets that “share” measurements (in the JPDA sense) for several sampling times, a dependence of their estimation errors ensues and this can be taken into account by calculating the resulting error correlations.

The resulting algorithm, called **JPDA Coupled Filter (JPDACF)** does the filtering in a *coupled* manner for the targets with “common” measurements, yielding a covariance matrix with off-diagonal blocks — *cross-covariances* — that reflect the correlation between the targets' state estimation errors.

JPDA with Coupling

The conditional probability for a joint association event (6.2.3-11) becomes

$$P\{\theta(k)|Z^k\} = \frac{1}{c} \frac{\phi!}{m(k)!} \mu_F(\phi) V^{-\phi} f_{t_{j_1}, t_{j_2}, \dots} [z_j(k), j : \tau_j = 1] \prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t} \quad (6.2.7-1)$$

where $f_{t_{j_1}, t_{j_2}, \dots}$ is the joint pdf of the measurements of the targets under consideration; t_{j_1} is the target to which $z_{j_1}(k)$ is associated in event θ .

The joint probabilities are not “boiled down” to the marginal association probabilities as in (6.2.5-1) for use in decoupled PDA filters. Instead, these joint probabilities are used directly in a *coupled filter*.

6.2.7 A Modification of the JPDAF: Coupled Filtering

The JPDAF

Denote the stacked vector of the predicted states of the targets under consideration (assumed here to be 2) and the associated covariance matrix

$$\hat{x}^T(k|k-1) = \begin{bmatrix} \hat{x}^1(k|k-1) \\ \hat{x}^2(k|k-1) \end{bmatrix} \quad (6.2.7-2)$$

$$P^T(k|k-1) = \begin{bmatrix} P^{11}(k|k-1) & P^{12}(k|k-1) \\ P^{21}(k|k-1) & P^{22}(k|k-1) \end{bmatrix} \quad (6.2.7-3)$$

where \hat{x}^t and P^{tt} correspond to target t ; $P^{t_1 t_2}$ is the **cross-covariance** between targets t_1 and t_2 (it will be zero before these targets become “coupled”).

The coupled filtering is done as follows

$$\hat{x}^T(k|k) = \hat{x}^T(k|k-1) + W^T(k) \sum_{\theta} P\{\theta(k)|Z^k\} [z^T(k, \theta) - \hat{z}^T(k|k-1)] \quad (6.2.7-4)$$

where

$$z^T(k, \theta) = \begin{bmatrix} z_{j_1(\theta)}(k) \\ z_{j_2(\theta)}(k) \end{bmatrix} \quad (6.2.7-5)$$

and $j_t(\theta)$ is the index of the measurement associated with target t in event θ at time k .

The filter gain in (6.2.7-4) is

$$W^T(k) = P^T(k|k-1) H^T(k)' [H^T(k) P^T(k|k-1) H^T(k)' + R^T(k)]^{-1} \quad (6.2.7-6)$$

where

$$H^T(k) = \begin{bmatrix} H^1(k) & 0 \\ 0 & H^2(k) \end{bmatrix} \quad R^T(k) = \begin{bmatrix} R^1(k) & 0 \\ 0 & R^2(k) \end{bmatrix} \quad (6.2.7-7)$$

are the (block diagonal) measurement matrix and noise covariance matrix, respectively, for the two targets under consideration. The predicted stacked measurement vector is

$$\hat{z}^T(k|k-1) = H^T(k) \hat{x}^T(k|k-1) \quad (6.2.7-8)$$

The state covariance update is as in (3.4.2-10).

6.2.8 The JPDAF — Summary

Assumptions of the JPDAF:

- There are *several targets* to be tracked in the presence of *false measurements*
- The number of targets is known
- The track of each target has been initialized
- The state equations of the targets are not necessarily the same
- The validation regions of these targets can intersect and have “common” measurements
- A target can give rise to at most one measurement — no multipath
- The detection of a target occurs independently over time and from other targets according to a known probability
- A measurement could have originated from at most one target (or none) — no unresolved measurements are considered here
- The conditional pdf of each target’s state given the past measurements is assumed *Gaussian* (a quasi-sufficient statistic that summarizes the past) and *independent across targets* with
 - means and
 - covariances

available from the previous cycle of the filter.

With the past summarized by an approximate sufficient statistic, the association probabilities are computed (only for the latest measurements) *jointly across the measurement and the targets*.

The JPDAF Algorithm Steps

- A validation matrix that indicates all the possible sources of each measurement is set up
- From this validation matrix all the *feasible joint association events* are obtained according to the rules
 - one source for each measurement
 - one measurement (or none) from each target
- The probabilities of these joint events are evaluated according to the assumptions
 - Target-originated measurements are Gaussian distributed around the predicted location of the corresponding target's measurement
 - False measurements are uniformly distributed in the surveillance region
 - The number of false measurements is distributed according to
 - ★ Poisson prior — *Parametric JPDA*
 - ★ Diffuse prior — *Nonparametric JPDA*
- *Marginal* (individual measurement-to-target) association probabilities are obtained from the joint association probabilities
- The target states are estimated by *separate (uncoupled)* PDA filters using these marginal probabilities.

The JPDACF Algorithm

Same assumptions as the JPDAF except that the targets' states are not assumed independently distributed when conditioned on the past.

This results in *coupled filter* — a filter for the stacked vector of the targets' states that uses the joint association probabilities directly.

6.2.9 Joint Probabilistic Data Association Filter Overview

The Joint Probabilistic Data Association Filter (JPDAF) is described as follows.

Modeling Assumptions

- There is a known number of established targets in clutter with known dynamic and measurement models (not necessarily the same)
- Measurements from one target can fall in the validation region of a neighboring target — this can happen over several sampling times and acts as a *persistent interference*
- The past is summarized by an approximate Gaussian sufficient statistic — specified by state estimates and covariances for each target.

The PDAF modeled all the incorrect measurements as *random interference* — its performance degrades when there is a persistent interference.

The JPDA Approach

- The measurement-to-target association probabilities are computed *jointly across the targets*
- The association probabilities are computed only for the latest set of measurements
- A *validation matrix* is defined with binary elements that indicate if measurement j lies in the validation gate of target t
- The *feasible association events* are obtained assuming
 - one source (target or clutter) for a measurement, and
 - at most one measurement per target
- As in the PDA there is a parametric and a nonparametric version (with a Poisson or diffuse prior for the number of false measurements).

The State Estimation

Options for state estimation (filtering):

(i) The states of the targets conditioned on the past observations are assumed *independent*. Then one has *decoupled estimation* for the targets under consideration — the **Joint Probabilistic Data Association Filter (JPDAF)**.

In this case one needs the *marginal association probabilities*, which are obtained from the joint probabilities by summing over all the joint events in which the marginal event of interest occurs.

(ii) The states of the targets given the past are *correlated* — they are characterized by means, covariances as well as cross-covariances.

This leads to *coupled estimation* for the targets under consideration — the **JPDA Coupled Filter (JPDACF)**.

Extension to Multiple Source Measurement

One can have an *unresolved (merged) measurement* from, e.g., two targets.

The **JPDAM** — JPDA with Merged measurement includes a special model for a merged measurement.

A version of this is the **JPDAMCF** — JPDA with Merged measurement and Coupled Filter.

Extension to a Splitting Target

A possible situation of interest is a platform that launches a weapon.

The JPDA has been extended to cover such a situation by using multiple models with the IMM configuration:

- there is a single nonmaneuvering target
- there is a single maneuvering target
- the target split into two targets.

This provides a “warm start” for the new target (see Subsection 1.4.4).

6.3 THE MULTIPLE HYPOTHESIS TRACKING

6.3.1 Introduction

The **Multiple Hypothesis Tracking (MHT)** approach considers

- the association of **sequences of measurements**

and evaluates

- the probabilities of *all* the **association hypotheses**.

This leads to a *complexity that increases exponentially with time* and appropriate techniques have to be used to limit the number of hypotheses under consideration:

- clustering to reduce the combinatorial complexity
- pruning of low probability hypotheses
- merging of similar hypotheses.

For each sequence of measurements — which is a **hypothesized track** — a *standard KF* yields the corresponding state estimate and covariance.

Target-Oriented vs. Measurement-Oriented Algorithms

The previous Bayesian approaches (Sections 3.4 and 3.5 for a single target and Section 6.2 for multiple targets) are of the **target-oriented** type, i.e., the probability that a measurement belongs to an established target is evaluated.

The MHT approach from this section is **measurement-oriented** in the sense that the probability that an established target or a new target gave rise to a certain measurement sequence is obtained.

The latter feature allows *inclusion of track initiation* for new targets within the framework of the algorithm.

6.3.2 The Hypothesis Generation Technique

6.3.2.2 The Hypothesis Generation Technique

Let the set of **association hypotheses** through time k be

$$\Omega^k = \{\Omega_j^k\} \quad (6.3.2-1)$$

This set is obtained from Ω^{k-1} and the latest set of measurements

$$Z(k) = \{z_i(k)\}_{i=1}^{m(k)} \quad (6.3.2-2)$$

as follows.

The set Ω^k is generated by forming new hypotheses from associating to Ω^{k-1} first $z_1(k)$, then the resulting set is augmented by associating $z_2(k)$, etc.

Thus, denoting

$$\bar{\Omega}^{k,0} \triangleq \Omega^{k-1} \quad (6.3.2-3)$$

one obtains $\bar{\Omega}^{k,i}$ by augmenting the set $\bar{\Omega}^{k,i-1}$ with *all the feasible associations* between its members and $z_i(k)$.

The **feasible associations** for $z_i(k)$ are

1. It is the continuation of a previous **track** or **measurement history** — a sequence of measurements hypothesized as having originated from a target, or, possibly, an established track
2. It is a false alarm
3. It is a new target.

In each new hypothesis, to each track (history) one can associate at most one current measurement, which has to fall in its validation region.

Remarks

The MHT requires an **exhaustive search** — all the hypotheses are generated and evaluated in order to find the most probable hypothesis.

The representation of the hypotheses is done using tree data structures (see Ch. 3 of [Bar90]), rather than matrices as in the JPDA.

6.3.3 Evaluation of the Hypothesis Probabilities

6.3.3 Evaluation of the Hypothesis Probabilities

A **current association event** $\theta(k)$ consists of

1. τ measurements from previously hypothesized/established tracks
2. ϕ measurements deemed false alarms or clutter
3. ν measurements from new targets

The following indicator variables pertaining to event $\theta(k)$ are defined

$$\tau_i = \tau_i[\theta(k)] = \begin{cases} 1 & \text{if } z_i(k) \text{ originated from a previous track} \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \dots, m(k) \quad (6.3.3-1)$$

$$\nu_i = \nu_i[\theta(k)] = \begin{cases} 1 & \text{if } z_i(k) \text{ is a new target} \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \dots, m(k) \quad (6.3.3-2)$$

$$\delta_t = \delta_t[\theta(k)] = \begin{cases} 1 & \text{if track } t \text{ from } \Omega^{k-1} \text{ is detected at } k \\ 0 & \text{otherwise} \end{cases} \quad (6.3.3-3)$$

The **number of continuing tracks** in $\theta(k)$ is

$$\tau = \sum_{i=1}^{m(k)} \tau_i \quad (6.3.3-4)$$

The **number of new tracks** in $\theta(k)$ is

$$\nu = \sum_{i=1}^{m(k)} \nu_i \quad (6.3.3-5)$$

The **number of false measurements** in $\theta(k)$ is

$$\phi = m(k) - \tau - \nu \quad (6.3.3-6)$$

Cumulative Association Events

A joint **cumulative event** (set of **association histories**) at k

$$\Theta^{k,l} = \{\Theta^{k-1,s}, \theta(k)\} \quad (6.3.3-7)$$

is made up of a **parent event** through $k-1$ and the **offspring event** or **current association event**

$$\theta(k) = \{\theta_T(k), \theta_N(k), \theta_F(k)\} \quad (6.3.3-8)$$

Current Association Events

The **current associations with established tracks** event is

$$\theta_T(k) = \bigcap_{i:\tau_i=1} \theta_{it_i}(k) \quad (6.3.3-9)$$

with t_i the index of the track to which $z_i(k)$ is associated.

The **new targets** event at time k is

$$\theta_N(k) = \bigcap_{i:\nu_i=1} \theta_{in_i}(k) \quad (6.3.3-10)$$

where n_i is the index given to the new target corresponding to $z_i(k)$. It is implicitly assumed here that one can initialize a new track based on a single measurement, i.e., *a full state estimate can be obtained* from it and possibly some a priori information.

The **false alarms** event at time k is

$$\theta_F(k) = \bigcap_{i:\tau_i+\nu_i=0} \theta_{i0(k)} \quad (6.3.3-11)$$

where $t = 0$ is the target index corresponding to false alarms.

The Probability of a Cumulative Event

Using Bayes' formula, the conditional probability of a cumulative event is

$$\begin{aligned} P\{\Theta^{k,l}|Z^k\} &= P\{\theta(k), \Theta^{k-1,s}|Z(k), Z^{k-1}\} \\ &= \frac{1}{c} p[Z(k)|\theta(k), \Theta^{k-1,s}, Z^{k-1}] P\{\theta(k)|\Theta^{k-1,s}, Z^{k-1}\} P\{\Theta^{k-1,s}|Z^{k-1}\} \end{aligned} \quad (6.3.3-12)$$

The Likelihood Function of a Current Event

If associated with track t_i , the pdf of $z_i(k)$ is $f_{t_i}(\cdot)$ — the innovation pdf from a standard KF. For a false alarm the pdf is uniform in the surveillance volume V , i.e., V^{-1} . The pdf of the measurement from a new target is also taken as V^{-1} , even though it might be reasonable that a new target is detected near the edge of the surveillance region.

Thus the pdf in (6.3.3-12) — the **likelihood function** of $\theta(k)$ — is

$$p[Z(k)|\theta(k), \Theta^{k-1,s}, Z^{k-1}] = \prod_{i=1}^{m(k)} \{f_{t_i}[z_i(k)]\}^{\tau_i} V^{-(1-\tau_i)} = V^{-\phi-\nu} \prod_{i=1}^{m(k)} \{f_{t_i}[z_i(k)]\}^{\tau_i} \quad (6.3.3-13)$$

where the indicator (6.3.3-1) has been used for compactness of notation.

6.3.3 Evaluation of the Hypothesis Probabilities

The Prior Probability of a Current Event

The second term on the r.h.s. of (6.3.3-12) is, similarly to (6.2.3-7),

$$\begin{aligned}
 P\{\theta(k)|\Theta^{k-1,s}, Z^{k-1}\} &= P\{\theta(k), \delta(\theta), \phi(\theta), \nu(\theta)|\Theta^{k-1,s}, Z^{k-1}\} \\
 &= P\{\theta(k)|\delta(\theta), \phi(\theta), \nu(\theta), \Theta^{k-1,s}, Z^{k-1}\} \\
 &\quad \cdot P\{\delta(\theta), \phi(\theta), \nu(\theta)|\Theta^{k-1,s}, Z^{k-1}\} \\
 &= P\{\theta(k)|\delta(\theta), \phi(\theta), \nu(\theta)\} P\{\delta(\theta), \phi(\theta), \nu(\theta)\} \quad (6.3.3-14)
 \end{aligned}$$

where $\delta(\theta)$ is the vector of detection indicators; ϕ and ν are the number of false alarms and new targets, respectively. The irrelevant conditionings above have been dropped.

The Combinatorics of the Current Events

The number of events with detections from the same set of τ tracks (as given by δ) and ν new tracks (with the leftover $\phi = m(k) - \tau - \nu$ false alarms) is given by

- (a) the number of *permutations* of $m(k)$, the total number of measurements at time k , taken as $\tau = m(k) - \phi - \nu$, (permutations because each track has an identity), multiplied by
- (b) the number of ways in which one can label ν measurements out of $\phi + \nu$ as new tracks, i.e., *combinations* of $\phi + \nu$ taken as ν (combinations because a new track has no identity associated with it).

Assuming each such event a priori equally likely, one has

$$\begin{aligned}
 P\{\theta(k)|\delta(\theta), \phi(\theta), \nu(\theta)\} &= \left[P_{\tau}^{m(k)} \cdot \binom{\phi + \nu}{\nu} \right]^{-1} = \left[\frac{m(k)!}{(m(k) - \tau)!} \frac{(\phi + \nu)!}{\phi! \nu!} \right]^{-1} \\
 &= \frac{\phi! \nu!}{m(k)!} \quad (6.3.3-15)
 \end{aligned}$$

Again, similarly to (6.2.3-9), the last term from (6.3.3-13) is

$$P\{\delta(\theta), \phi(\theta), \nu(\theta)\} = \prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1 - \delta_t} \mu_F(\phi) \mu_N(\nu) \quad (6.3.3-16)$$

where P_D^t is the detection probability of track t ; $\mu_F(\cdot)$ is the **pmf of the number of false measurements** and $\mu_N(\cdot)$ is the **pmf of the number of new targets**.

Combining (6.3.3-15) and (6.3.3-16) in (6.3.3-14) yields

$$P\{\theta(k)|\Theta^{k-1,s}, Z^{k-1}\} = \frac{\phi! \nu!}{m(k)!} \mu_F(\phi) \mu_N(\nu) \prod_t (P_D^t)^{\delta_t} (1 - P_D^t)^{1 - \delta_t} \quad (6.3.3-17)$$

6.3.3 Evaluation of the Hypothesis Probabilities

The Recursion of the Probability of a Cumulative Event

Using (6.3.3-13) and (6.3.3-17) in (6.3.3-12) yields the final expression of the conditional probability of a **cumulative event** $\Theta^{k,l}$ as

$$P\{\Theta^{k,l}|Z^k\} = \frac{1}{c m(k)!} \frac{\phi! \nu!}{\mu_F(\phi) \mu_N(\nu)} V^{-\phi-\nu} \prod_{i=1}^{m(k)} \{f_{t_i}[z_i(k)]\}^{\tau_i} \prod_t [(P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t}] P\{\Theta^{k-1,s}|Z^{k-1}\} \quad (6.3.3-18)$$

in terms of its **parent event** $\Theta^{k-1,s}$.

The Most Probable Hypothesis

The **most probable hypothesis** is obtained from an **exhaustive search** over all the hypotheses. Thus organizing the hypotheses and reducing their number is a key issue.

One Cycle of the MHT Algorithm

Figure 6.3.3-1 summarizes one cycle of the MHT algorithm.

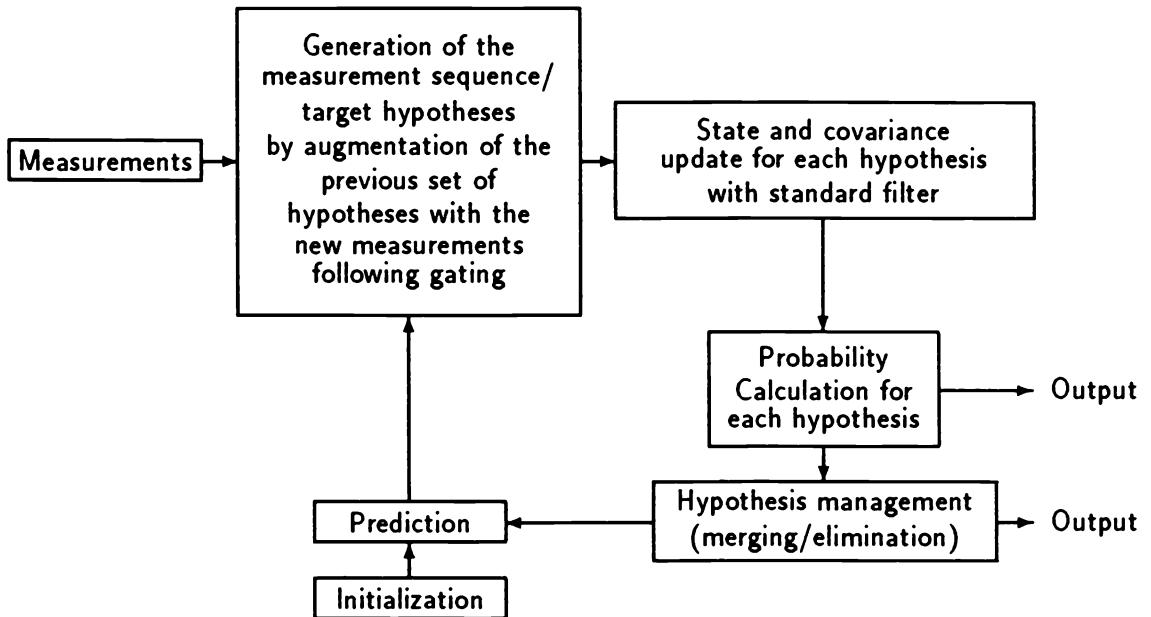


Figure 6.3.3-1: One cycle of the MHT.

6.3.4 Clustering and Hypotheses Reduction Techniques

The simultaneous consideration of hypotheses has to be done within clusters. A **cluster** is a set of tracks with *common measurements*.

Initially, one cluster is set up for each confirmed target. Each new measurement is associated with a cluster if it falls in the validation region of any track from that cluster.

A new cluster is set up for each unassociated measurement.

If a measurement is associated with more than one cluster then those clusters are combined into a **supercluster**.

If tracks within a cluster separate and have no more common measurements, then this cluster is subdivided accordingly into smaller clusters.

The growth of the number of hypotheses (exponential in the worst case) can be limited by:

- Eliminating hypotheses with negligible probabilities
- Combining the hypotheses (histories) with similar effects, e.g., the same number of targets but with “slightly” different state estimates.

Display of the Results

- Display the most probable hypothesis (which is not necessarily the true one and thus not the most informative display), or
- Use the **coordinated presentation** [BB89]: indicate the location, extent and the **expected number of targets** in each cluster.

Remark

In very high density scenarios only **cluster tracking** might be feasible. An efficient way of cluster tracking is discussed in [DBP90].

6.3.5 Inclusion of Occasional Measurements

6.3.5 Inclusion of Occasional Measurements

It is sometimes of interest to include an **occasional measurement**. Such measurements can arrive one at a time asynchronously, e.g., from visual sightings or when the target emits energy and a passive sensor detects it.

The possible origins of such a measurement, denoted as z , are

- (a) from a previously established track t (event θ_t , $t = 1, \dots, N_T$)
- (b) from a false alarm (event θ_0)
- (c) from a new target (event θ_{N_T+1})

where N_T is the number of established tracks.

The conditional probabilities of the above events can be obtained from Bayes' formula as follows

$$P\{\theta_t|z\} = \frac{1}{c} p(z|\theta_t) P\{\theta_t\} \quad (6.3.5-1)$$

where c is the normalization constant and

$$p(z|\theta_t) = \begin{cases} V^{-1} & t = 0 \text{ and } t = N_T + 1 \\ f_t(z) & t = 1, \dots, N_T \end{cases} \quad (6.3.5-2)$$

The prior probabilities of these events can be taken (somewhat subjectively) as proportional to the following expressions

$$P\{\theta_t\} \propto \begin{cases} \bar{\phi} & t = 0 \\ P_D^t & t = 1, \dots, N_T \\ \bar{\nu} & t = N_T + 1 \end{cases} \quad (6.3.5-3)$$

where $\bar{\phi}$ is the expected number of false alarms for the sensor under consideration; P_D^t is the detection probability of established target t by this sensor; $\bar{\nu}$ is the expected number of new targets to be detected by this sensor.

These expressions can be then combined into (6.3.5-1).

6.3.6 Example of Track Initiation

6.3.6 Example of Track Initiation

The problem on which this algorithm was simulated is one of a nearly constant velocity target moving in a plane. The motion was assumed to be independent for the two coordinates and obeying the same equation.

With position and velocity components, the state equation along each coordinate was taken as

$$x(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ T \end{bmatrix} v(k) \quad (6.3.6-1)$$

and the measurement (position only)

$$z(k) = [1 \ 0] x(k) + w(k) \quad (6.3.6-2)$$

with the noises zero-mean, white and mutually independent with variances

$$E[v(k)^2] = q \quad (6.3.6-3)$$

$$E[w(k)^2] = r \quad (6.3.6-4)$$

These two variances were taken as $q = r = 0.02$ and the sampling period $T = 1$.

The Scenario Parameters

The probability mass functions of the number of *false targets* and *new true targets* in (6.3.3-16) were taken as Poisson with spatial densities

$$\lambda_{FT} = 0.1 \quad (6.3.6-5)$$

$$\lambda_{NT} = 0.5 \quad (6.3.6-6)$$

respectively.

The probability of detection of the target was taken as $P_D = 0.9$.

The track initiation is done from the first measurement assuming an initial velocity variance of 1. The evolution of the hypotheses and their probabilities over five sampling times is illustrated in Figure 6.3.6-1.

After each measurement the most likely hypothesis is that there is one target. The figure shows the estimated position and velocity as well as the 1σ position error circle. After the first measurement is processed the probability that there is a target is $5/6$ because of the 5:1 relative densities of new targets vs. false measurements.

The probability that there is at least one target increases monotonically to practically 100% after 5 measurements.

6.3.6 Example of Track Initiation

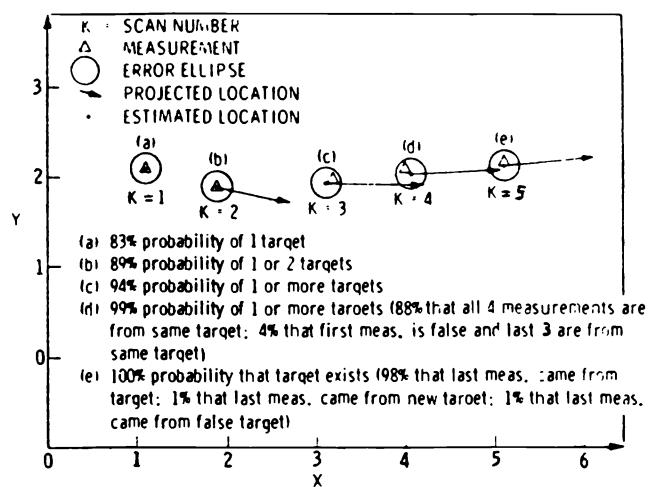


Figure 6.3.6-1: Track initiation.

6.3.7 Example of Crossing Tracks

6.3.7 Example of Crossing Tracks

In this example, based on the same target model as before, two targets have crossing trajectories. It is known that there is one target starting from the bottom left of Figure 6.3.7-1 while the second one starting from the top left of the figure “emerges” following a number of measurements.

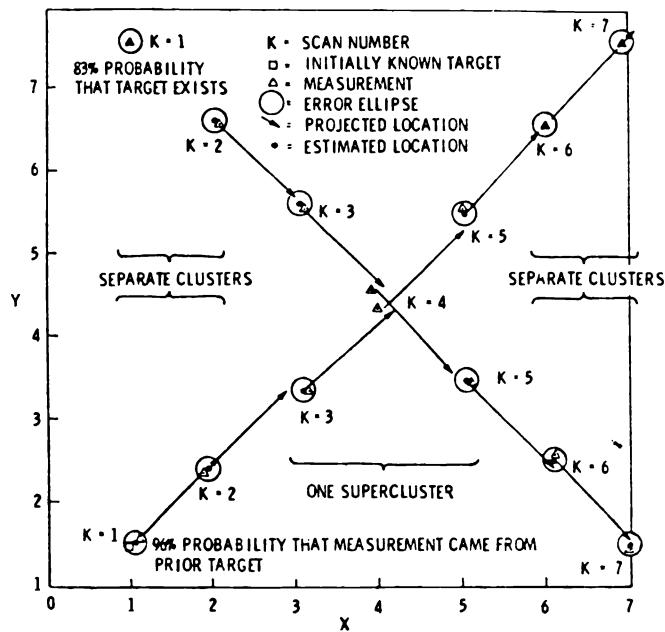


Figure 6.3.7-1: Crossing tracks.

6.3.7 Example of Crossing Tracks

Prior to time $k = 3$ the clusters corresponding to two targets are separated; however, at time $k = 3$ the two clusters become overlapping, forming a supercluster. This happens because the lower measurement at $k = 3$ could be associated with the tentative upper target if the measurement at $k = 2$ is from clutter since the velocity variance is then still the prior (equal to 1). After computing the posterior probability, however, this hypothesis becomes negligible and is eliminated. Thus, at $k = 3$, after processing the measurements, the supercluster is eliminated.

At $k = 4$ a supercluster is formed again. By this time the tentative target (coming from the top) is confirmed but there is an association uncertainty: with 60% probability it has the top measurement and with 40% the lower one.

At $k = 5$ there is no new association uncertainty but the hypotheses pertaining to time $k = 4$ are still unresolved: the probabilities are 54% and 46%, respectively. This is a case where later data did not reduce a past association uncertainty: in fact the ambiguity increased.

By time $k = 6$ the difference in the hypotheses at $k = 4$ becomes irrelevant since the target estimates are practically the same. From then on the target clusters separate and there are two separate track maintenance problems.

6.3.8 Example of a High Target Density Environment

6.3.8 Example of a High Target Density Environment

In this example, illustrated in Figure 6.3.8-1, there are 5 targets but only 4 have been initialized; the fifth will be initialized by the data association procedure.

The initial position and velocity estimates for the 4 known targets are indicated in the figure together with the measurements obtained during the next 6 sampling times.

The process and measurement noises had variances $q = r = 0.4$.

The measurements are shown grouped according to the most probable hypothesis. There were 16 hypotheses with non-negligible probability.

The unknown target is formed by the data association procedure from measurements 2, 8 and 14.

The general grouping of the measurements, even though it is time-varying, corresponds to the actual targets.

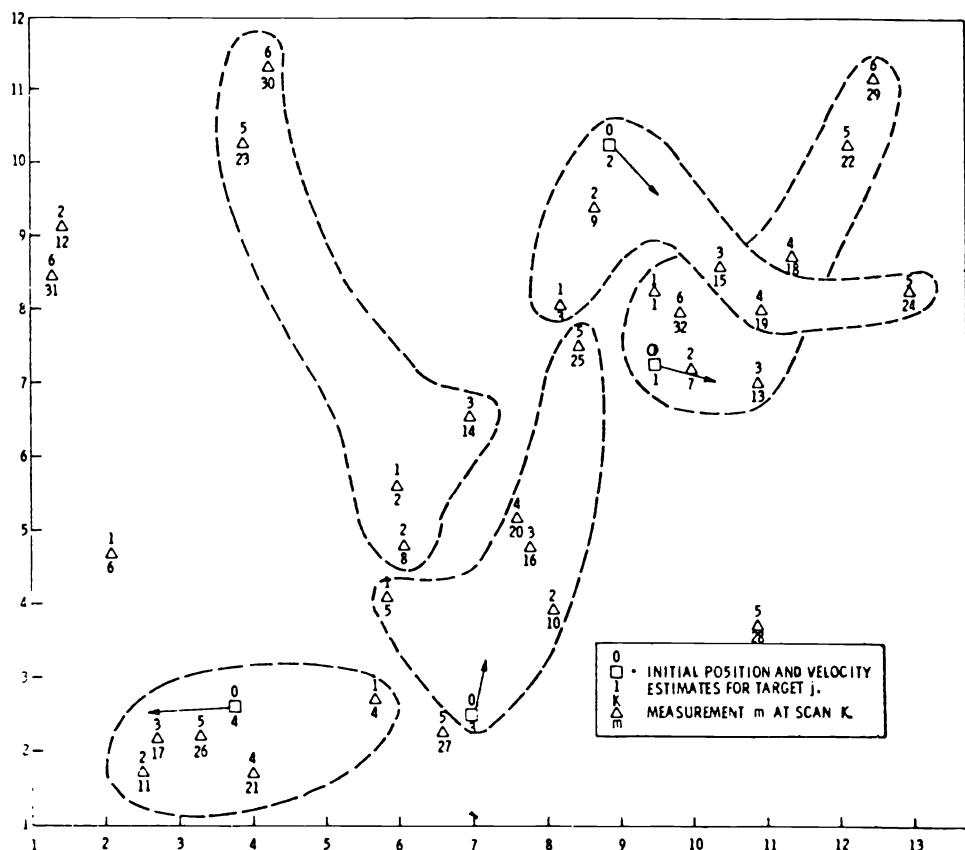


Figure 6.3.8-1: Tracking in a dense target environment.

6.3.9 Correct Association Probability

The factors affecting tracking performance are the environmental variables such as the **extraneous true target density** λ_{ET} , the **false target (clutter) density** λ_{FT} , and the validation gate size, determined by the innovation covariance S .

The probability of making a **correct association** between a target and the measurement originated from it — defined here as the situation where the correct measurement was closer to the predicted measurement than any extraneous measurement — is a key factor in tracking performance.

From the point of view of a given target, the spatial density of **extraneous measurements** is

$$\lambda = \lambda_{FT} + P_D \lambda_{ET} \quad (6.3.9-1)$$

A two-dimensional innovation covariance matrix of the following form is considered

$$S = I\sigma^2 \quad (6.3.9-2)$$

The **density-variance product** can be defined as

$$\rho = \lambda\sigma^2 \quad (6.3.9-3)$$

Then the expected number of extraneous measurements in the 1σ circle is $\pi\rho$ because the area of the “ g -sigma” circle is $\pi g^2\sigma^2$ and the expected number of extraneous measurements in it is $\pi g^2\sigma^2\lambda$.

Figure 6.3.9-1 shows the probability that, at a given sampling time, the correct measurement will be “correctly associated,” as a function of the “density-variance” product ρ . The target detection probability was taken as $P_D = 0.9$, both for the target of interest as well as for the new ones.

For $\rho = \lambda\sigma^2 = 0.1$ the probability of **correct association**, as defined above, is down to about 0.65. This corresponds to a density of 0.3 per 1σ region, 2.7 per 3σ region ($\gamma = g^2 = 9$) and 4.8 per 4σ region ($\gamma = g^2 = 16$).

These curves are *for a single scan* and they do not reflect the performance of a filter over more than one scan. Nevertheless, as observed in Subsection 3.4.6, the upper limit of “trackability” for the PDAF was around 5 false measurements per $\gamma = 16$ region, which is close to the above value of $\rho = 0.1$.

A more detailed discussion on this topic can be found in Ch. 7 of [Bar92].

6.3.9 Correct Association Probability

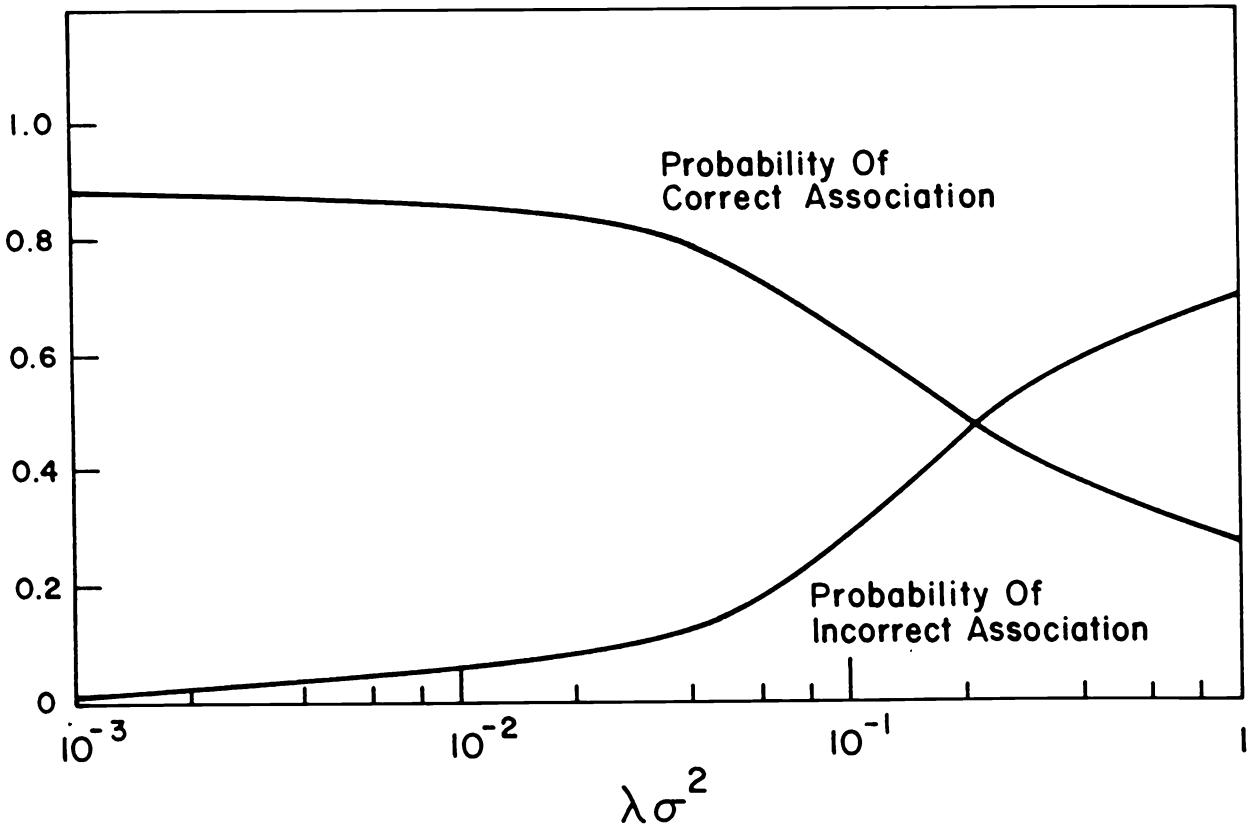


Figure 6.3.9-1: The probabilities of “correct” and “incorrect” data association vs. normalized extraneous measurement density.

6.3.10 Monte Carlo Runs

6.3.10 Monte Carlo Runs

A surveillance area of 10×10 was considered. The false target density was $\lambda_{FT} = 0.01$. The total target density was $\lambda_{TT} = 0.05$ with a fraction $f_{KT} = 0.8$ as known targets at the initial time. The detection probability was $P_D = 0.9$.

The density of the new targets was taken as

$$\lambda_{NT} = (1 - f_{KT})\lambda_{TT}P_D \quad (6.3.10-1)$$

In the cases simulated the **hypothesis dropping threshold** α was taken as 0.01, 0.1 and 0.5. In the last case only the most likely hypothesis was retained. The “density-variance” parameter ρ , defined in (6.3.9-3), ranged over the values 0.01, 0.03 and 0.1.

Each data point from Table 6.3.10-1 is the result of 10 runs over 10 sampling times.

Scenario Parameters	Number of Hypotheses	Percent Targets Tracked	Percent False Targets	Normalized Error
$q = r = 0.04, \alpha = 0.01 (\rho = 0.01)$	10.52	97.3	2.9	0.920
$q = r = 0.04, \alpha = 0.1 (\rho = 0.01)$	7.67	97.1	3.8	0.922
$q = r = 0.04, \alpha = 0.5 (\rho = 0.01)$	4.40	81.0	0.6	0.870
$q = r = 0.12, \alpha = 0.01 (\rho = 0.03)$	13.65	93.6	6.7	1.053
$q = r = 0.12, \alpha = 0.1 (\rho = 0.03)$	8.04	92.6	6.0	1.052
$q = r = 0.12, \alpha = 0.5 (\rho = 0.03)$	4.40	75.4	7.0	1.062
$q = r = 0.40, \alpha = 0.01 (\rho = 0.10)$	16.41	89.1	7.7	1.055
$q = r = 0.40, \alpha = 0.10 (\rho = 0.10)$	7.90	85.2	11.8	1.071
$q = r = 0.40, \alpha = 0.50 (\rho = 0.10)$	4.40	75.0	6.5	1.043
$q = r = 0.04, \alpha = 0.10 (\rho = 0.01)$ (double λ_{NT})	7.76	99.3	5.8	0.937
$q = r = 0.12, \alpha = 0.10 (\rho = 0.03)$ (double λ_{NT})	8.05	94.3	6.2	1.052
$q = r = 0.40, \alpha = 0.10 (\rho = 0.10)$ (double λ_{NT})	7.81	90.8	12.1	1.098

Table 6.3.10-1: Results of Monte Carlo simulations.

As the density-variance product increases, the difficulty of correct measurement-to-target association obviously increases. Furthermore, for low threshold $\alpha = 0.01$ there is little “pruning” of the hypotheses and their number is quite large.

The normalized error for position (divided by 2) stays around its theoretical value of 1, indicating good consistency of the estimator.

In the last three cases shown in the Table the filter used a twice higher value of λ_{NT} than the one calculated from (6.3.10-1); in spite of this, the filter seems to be quite robust.

6.3.11 The MHT — Summary

Assumptions of the MHT:

- There is an *unknown number of targets* each with the same state equation
- Tracks have not necessarily been initialized
- At each sampling time any measurement could have originated from:
 - an established track (a sequence of previous measurements) or
 - a new target or
 - a false alarm
- There can be at most one measurement from a target
- Target detections occur independently according to a known probability
- A measurement could have originated from at most one target
- New targets can appear at each sampling time
- False measurements appear according to a Poisson process
- The *origin of each sequence of measurements* is considered.

The MHT Algorithm Steps

- The hypotheses at the current time are obtained from
 - the set of hypotheses at the previous time, augmented with
 - all the feasible associations of the current measurements
- The probability of each new hypothesis is evaluated assuming
 - measurements associated with a track are Gaussian distributed around the predicted location of the corresponding track's measurement
 - false measurements are uniformly distributed in the surveillance region and appear according to a fixed rate Poisson process
 - new targets are uniformly distributed in the surveillance region (or according to some other pdf) and appear according to a Poisson process with a certain rate (or from a “pool of undetected targets”)
- The state estimate corresponding to each hypothesized track is obtained from a standard filter
- An elaborate hypothesis management is needed
- The selection of the most probable hypothesis amounts to an exhaustive search over the set of feasible hypotheses.

Hypothesis Management

The evaluation of the probabilities of hypotheses is done within *clusters* — sets of tracks that are “connected” by “common” measurements.

Clusters can merge into *superclusters* or separate into subsets which then become independent clusters.

Each “new target” not in an existing cluster becomes a separate cluster.

The number of hypotheses is reduced by

- dropping hypotheses with probability below a threshold
- combining hypotheses that have *common measurements in a sliding window* — they have nearly the same state estimates.

Remarks

While the MHT is “optimal” in the sense that it makes *no overt approximations or simplifications*, in order to implement it, many *hidden approximations and simplifications* have to be made, which leads to *loss of optimality*.

Therefore, this method is not a panacea and has to be treated like any other approach — cautiously and with a clear understanding of all of its assumptions.

The performance evaluation of the MHT — for which even one run might be quite expensive — is to date possible only via Monte Carlo runs. Some bounding techniques that rely on a reduced number of runs have been proposed in [Dau90] (see also Chapter 6 of [Bar92]).

The extension of the MHT to multiple models, to include the situation of maneuvering targets, can be found in Chapter 3 of [Bar90]. The complex hypothesis management can be handled using tree data structures (see Chapter 3 of [Bar90]).

6.3.12 Multiple Hypothesis Tracking Overview

The **Multiple Hypothesis Tracking (MHT)** approach considers

- the association of *sequences of measurements* and evaluates
- the probabilities of *all the association hypotheses*.

This leads to a *complexity that increases exponentially with time* and appropriate techniques have to be used to limit the number of hypotheses under consideration:

- clustering to reduce the combinatorial complexity
- pruning of low probability hypotheses
- merging of similar hypotheses.

Target-Oriented vs. Measurement-Oriented

The PDA/JPDA are of the *target-oriented* type, i.e., the probability that a measurement belongs to an established target is evaluated.

The MHT approach is *measurement-oriented* in the sense that the probability that

- an established target, or
- a new target

gave rise to a certain measurement sequence is obtained.

The latter feature allows *inclusion of track initiation* for new targets within the framework of the algorithm.

Hypothesis Generation

The set of hypotheses at the current time is generated by augmenting the previous set with each new measurement with *all the feasible associations*.

The feasible associations for a current measurement are

1. It is the continuation of a previous history — a sequence of measurements hypothesized as having originated from a target, or, possibly, an established track
2. It is a false alarm
3. It is a new target.

In each new hypothesis, to each track (history) one can associate at most one current measurement, which has to fall in its validation region.

The MHT Algorithm Steps

- Each hypothesis probability evaluation assumes
 - measurements associated with a track are Gaussian distributed around the predicted location of the corresponding track's measurement
 - false measurements are uniformly distributed in the surveillance region and appear according to a Poisson process with a certain rate
 - new targets are uniformly distributed in the surveillance region and appear according to a Poisson process with a certain rate
- The state estimate corresponding to each hypothesized track is obtained from a standard filter
- An elaborate hypothesis management is needed
- The selection of the most probable hypothesis amounts to an exhaustive search over the set of feasible hypotheses.

Hypothesis Management

The evaluation of the probabilities of hypotheses is done within *clusters* — sets of tracks that are “connected” by “common” measurements.

Clusters can merge into *superclusters* or separate into subsets which then become independent clusters.

Each “new target” not in an existing cluster becomes a separate cluster.

The number of hypotheses is reduced by

- dropping hypotheses with probability below a threshold
- combining hypotheses that have *common measurements in a sliding window* — they have nearly the same state estimates.

Tree data structures are used in hypothesis management.

Figure 6.3.12-1 summarizes one cycle of the MHT algorithm.

6.3.12 Multiple Hypothesis Tracking Overview

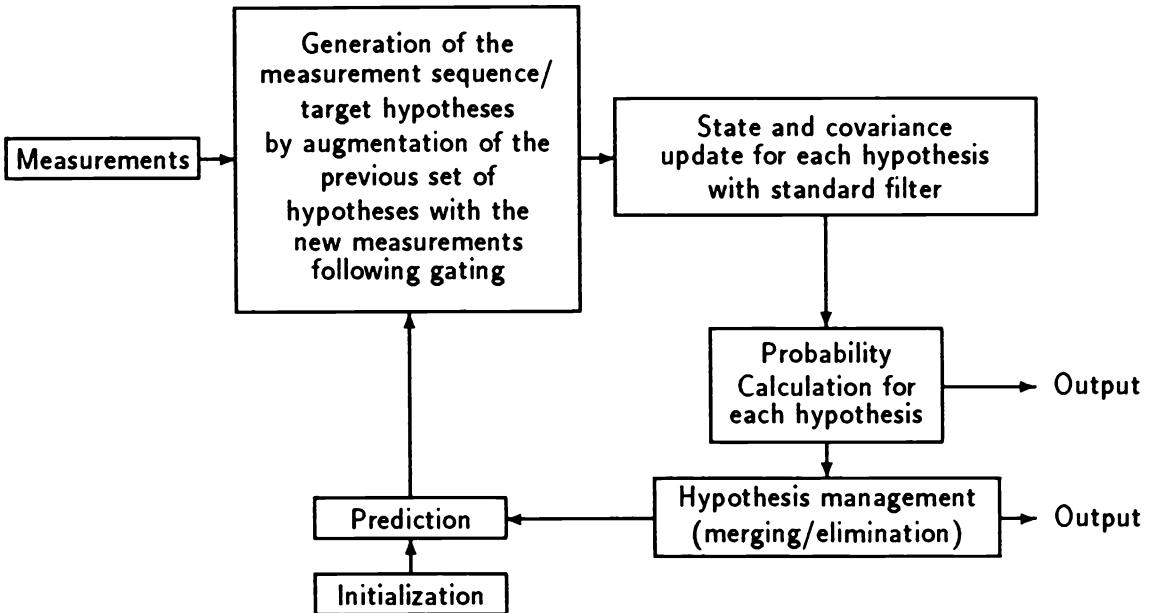


Figure 6.3.12-1: One cycle of the MHT.

Remarks

While the MHT is “optimal” in the sense that it makes *no overt approximations or simplifications*, in order to implement it, *hidden approximations and simplifications* have to be made.

Therefore, this method is not a panacea and has to be treated like any other approach — cautiously and with a clear understanding of all of its assumptions.

The performance evaluation of the MHT — for which even one run might be quite expensive — is to date possible only via Monte Carlo runs. Some bounding techniques that rely on a reduced number of runs have been proposed in [Dau90] (see also Chapter 6 of [Bar92]).

Extension

The inclusion of multiple motion models for maneuvering targets appears in Chapter 3 of [Bar90].

6.4 MULTIPLE TARGET TRACKING WITH POSSIBLY UNRESOLVED MEASUREMENTS

6.4.1 Introduction

When two targets are close enough in the measurement space they will give only one **merged (unresolved) measurement** due to the inherent finite **resolution** capability of any signal processor/detector.

The JPDAF can track targets with measurements whose origin is:

- One of the targets in track or
- None of the targets — false alarm or clutter.

The possibility that a measurement originated from the merging of the detections from two targets will be accounted in the extension of the JPDA, called **JPDA with merged measurements (JPDAM)**, which uses a probabilistic model for the merged measurements from two targets. Then the state estimation is presented with a modification to use the “apportioned” merged measurement.

Assumptions

The tracks of the targets of interest have been initialized.

Two targets are considered with generic dynamic equation

$$x_t(k+1) = F_t(k)x_t(k) + v_t(k) \quad t = 1, 2 \quad (6.4.1-1)$$

with zero-mean white process noises with covariance $Q_t(k)$, mutually uncorrelated.

The correct measurement corresponding to target t , *when resolved*, is

$$z_t(k) = H_t(k)x_t(k) + w_t(k) \quad (6.4.1-2)$$

with the measurement noise zero-mean, white with covariance $R_t(k)$.

6.4.2 The Model for the Merged Measurements

6.4.2 The Model for the Merged Measurements

Assumptions

- Merging of two measurements occurs if the (noisy) measurements from the two targets fall in the same resolution cell
- The location of the merged measurement depends on the relative strength of the two signals
- The components of the measurement vector are uncorrelated.

The n -dimensional measurement vector from target t is denoted as

$$z_t = [z_{t1} \ \cdots \ z_{tn}]' \quad t = 1, 2 \quad (6.4.2-1)$$

They are assumed Gaussian, mutually uncorrelated with predicted values

$$\hat{z}_t = [\hat{z}_{t1} \ \cdots \ \hat{z}_{tn}]' \quad (6.4.2-2)$$

and associated innovation covariance matrices

$$S_t = \text{diag}[\sigma_{ii}^2] \quad (6.4.2-3)$$

where the time arguments have been omitted for simplicity.

Two measurements merge if they fall in the same n -dimensional **resolution cell** C of size δ_i for component i of the measurement,

$$C \triangleq \{z : k_i\delta_i < z_{ti} < (k_i + 1)\delta_i, i = 1, \dots, n\} \quad (6.4.2-4)$$

where k_i is an integer.

The signal powers are denoted by s_t , $t = 1, 2$, and

$$\alpha \triangleq \frac{s_1}{s_1 + s_2} \quad (6.4.2-5)$$

which yields the **relative strength** (of target 1), with $0 < \alpha < 1$.

Merged Measurement Model

The merged measurement is modeled as the weighted linear combination

$$z_0 = \alpha z_1 + (1 - \alpha) z_2 \triangleq [z_{01} \ \cdots \ z_{0n}]' \quad (6.4.2-6)$$

6.4.2 The Model for the Merged Measurements

The pdf of the Merged Measurement

The pdf of the merged measurement (6.4.2-6) is obtained using Bayes' formula

$$\begin{aligned} p(z_0|z_1, z_2 \in C) &= p(z_0) \frac{P\{z_1, z_2 \in C|z_0\}}{P\{z_1, z_2 \in C\}} \\ &\triangleq p(z_0) \frac{Q_M(z_0)}{P_M} \\ &\triangleq g_{1,2}(z_0) \end{aligned} \quad (6.4.2-7)$$

where

- $p(z_0)$ is the pdf of the linear combination (6.4.2-6) of the measurements
- Q_M is the conditional merging probability (conditioned on the value of the measurement z_0)
- P_M is the unconditional merging probability.

Lengthy derivations presented in Subsection 6.4.8 yield the pdf (6.4.2-7) as approximately Gaussian with moments

$$\bar{z}_{ai} = \hat{z}_{0i} - \frac{\kappa_i \sigma_{0i}^2 \hat{w}_i}{\sigma_{wi}^2} e^{-\frac{\delta_i}{4\sigma_{wi}}} \quad (6.4.2-8)$$

$$\bar{\sigma}_{ai}^2 = \sigma_{0i}^2 - \frac{\kappa_i^2 \sigma_{0i}^4}{\sigma_{wi}^2} e^{-\frac{\delta_i}{2\sigma_{wi}}} \quad (6.4.2-9)$$

where

$$\hat{z}_{0i} \triangleq \alpha \hat{z}_{1i} + (1 - \alpha) \hat{z}_{2i} \quad (6.4.2-10)$$

$$\sigma_{0i}^2 \triangleq \alpha^2 \sigma_{1i}^2 + (1 - \alpha)^2 \sigma_{2i}^2 \quad (6.4.2-11)$$

$$\kappa_i \triangleq [\alpha \sigma_{1i}^2 - (1 - \alpha) \sigma_{2i}^2] \sigma_{0i}^{-2} \quad (6.4.2-12)$$

$$\hat{w}_i \triangleq \hat{z}_{1i} - \hat{z}_{2i} \quad (6.4.2-13)$$

$$\sigma_{wi}^2 \triangleq \sigma_{1i}^2 + \sigma_{2i}^2 \quad (6.4.2-14)$$

6.4.3 The Data Association

At the current time k one has the set of validated measurements

$$Z(k) = \{z_j(k)\}_{j=1}^m \quad (6.4.3-1)$$

in the union of the validation regions of the two targets of interest.

Association probabilities will be computed only for the latest set of measurements conditioned upon

$$Z^k = \{Z(k), Z^{k-1}\} \quad (6.4.3-2)$$

The **validation matrix** (at the current time k)

$$\Omega = [\omega_{jt}] \quad j = 1, \dots, m; \quad t = 0, 1, \dots, N_T \quad (6.4.3-3)$$

consists of binary elements indicating if measurement j has been validated for target t . Index $t = 0$ denotes false alarm (clutter) origin, which is possible for each measurement. This matrix summarizes all the feasible individual (*marginal*) measurement-to-target associations.

A **joint association event** θ is represented by a matrix consisting of the units from Ω corresponding to the associations *assumed* in it

$$\hat{\Omega}(\theta) = [\hat{\omega}_{jt}(\theta)] \quad (6.4.3-4)$$

Feasible events have to satisfy the following conditions:

- The **detection indicator** for target $t = 1, \dots, N_T$

$$\delta_t(\theta) \triangleq \sum_{j=1}^m \hat{\omega}_{jt}(\theta) \leq 1 \quad (6.4.3-5)$$

- The **target association indicator** for measurement $j = 1, \dots, m$

$$\tau_j(\theta) \triangleq \sum_{t=1}^{N_T} \hat{\omega}_{jt}(\theta) \leq 2 \quad (6.4.3-6)$$

- The **resolution indicator**

$$\rho(\theta) \triangleq \begin{cases} 1 & \text{if } \tau_j(\theta) \leq 1 \quad \forall j \\ 0 & \text{otherwise} \end{cases} \quad (6.4.3-7)$$

6.4.3 The Data Association

If the resolution indicator is unity then there are no unresolved measurements.

For example, consider the following validation matrix for two measurements ($j = 1, 2$) and two targets ($t = 0$ [clutter origin], 1, 2), illustrated in Figure 6.4.3-1.

$$\Omega = \begin{matrix} & t & 0 & 1 & 2 \\ j & \begin{matrix} 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 \end{matrix} \end{matrix}$$

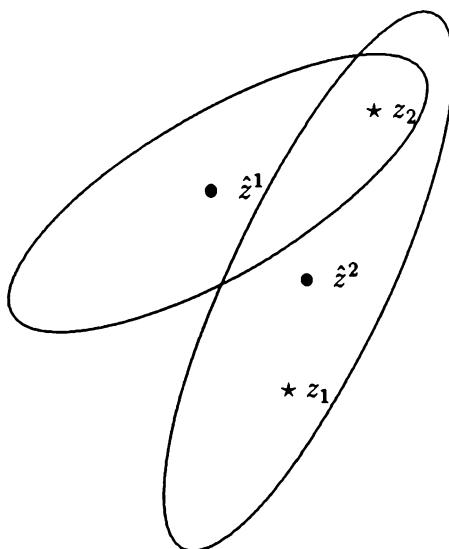


Figure 6.4.3-1: Two targets with one possible merged measurement.

The feasible association events, characterized by the corresponding matrix $\hat{\Omega}$, are in this case:

$$\hat{\Omega}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{\Omega}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \hat{\Omega}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (6.4.3-8)$$

$$\hat{\Omega}_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \hat{\Omega}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{\Omega}_6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad (6.4.3-9)$$

The association event represented by $\hat{\Omega}_6$, the only one with two units in the same row ($j = 2$), corresponds to assuming this measurement as having originated from both targets and measurement 1 from clutter (column 0).

6.4.4 The Association Probabilities

6.4.4 The Association Probabilities

The **joint association event probabilities** for $N_T = 2$ targets and m measurements with a possible *merged (unresolved) measurement* are, from Subsection 6.4.9, given by

$$P(\theta|Z^k) = \frac{\lambda^\phi}{c'} \prod_{j:\tau_j=2} g_{1,2}[z_j(k)] \prod_{j:\tau_j=1} f_{t_j}[z_j(k)] \cdot \prod_{t=1}^2 P_{D_t}^{\delta_t} (1 - P_{D_t})^{1-\delta_t} P_M^{1-\rho} (1 - P_M)^\rho \quad (6.4.4-1)$$

where c' is the normalization constant.

The following **marginal event probabilities** are then obtained for target t :

- No measurement is associated with it (event θ_{0t})

$$\beta_{0t}(k) = \sum_{\theta} P\{\theta|Z^k\} [1 - \delta_t(\theta)] \quad (6.4.4-2)$$

- The **resolved measurement** j is associated with it (event θ_{jt})

$$\beta_{jt}(k) = \sum_{\theta} P\{\theta|Z^k\} \hat{\omega}_{jt}(\theta) \rho(\theta) \quad (6.4.4-3)$$

- The **unresolved measurement** j is associated with it (event $\bar{\theta}_{jt}$)

$$\bar{\beta}_{jt}(k) = \sum_{\theta} P\{\theta|Z^k\} \hat{\omega}_{jt}(\theta) [1 - \rho(\theta)] \quad (6.4.4-4)$$

6.4.5 The State Estimation

The state estimation is carried out according to the decoupled approach discussed in Sub-section 6.2.5.

For each target the set of *marginal events*, whose probabilities have been obtained, are *mutually exclusive and exhaustive*.

The MMSE estimate — the conditional mean — of the state of target t can be written by decomposing it according to these events as follows:

$$\hat{x}^t(k|k) \triangleq \hat{x}^t(k|k-1)\beta_{0t}(k) + \sum_{j=1}^m [\hat{x}_j^t(k|k)\beta_{jt}(k) + \bar{x}_j^t(k|k)\bar{\beta}_{jt}(k)] \quad (6.4.5-1)$$

where

- $\hat{x}^t(k|k-1)$ is the estimate at k if no measurement is associated with target t (i.e. the prediction from $k-1$)
- $\hat{x}_j^t(k|k)$ is the updated state with measurement j , assumed *resolved*
- $\bar{x}_j^t(k|k)$ is the updated state with measurement j , assumed *merged (unresolved)*.

Update with a Resolved Measurement

The update with a *resolved* measurement is done in the standard manner

$$\hat{x}_j^t(k|k) = \hat{x}^t(k|k-1) + W^t(k)[z_j(k) - \hat{z}^t(k|k-1)] \quad (6.4.5-2)$$

$$W^t(k) = P^t(k|k-1)H^t(k)'S^t(k)^{-1} \quad (6.4.5-3)$$

$$S^t(k) = H^t(k)P^t(k|k-1)H^t(k)' + R(k) \quad (6.4.5-4)$$

The covariance associated with the state of target t conditioned on measurement j being *correct and resolved* is

$$\hat{P}^t(k|k) = [I - W^t(k)H^t(k)]P^t(k|k-1) \quad (6.4.5-5)$$

6.4.5 The State Estimation

Update with a Merged Measurement

The update with a measurement assumed *merged* is done as follows.

The normal approximation with the mean with components (6.4.2-8) and variances given in (6.4.2-9) is used for the pdf of such a measurement:

$$p[z_j(k)|Z^{k-1}, \bar{\theta}_{jt}(k)] = \mathcal{N}[z_j(k); \bar{z}^t(k|k-1), \bar{S}^t(k)] \quad (6.4.5-6)$$

where the predicted merged measurement has moments derived in Subsection 6.4.2.

For the calculation of the filter gain in this case it will be assumed that

$$\text{cov}[x^t(k), z_j(k)|Z^{k-1}, \bar{\theta}_{jt}(k)] = \alpha^t P^t(k|k-1) H^t(k)' \quad (6.4.5-7)$$

where α^t is the **relative signal strength** of target t as in (6.4.2-5).

Then the gain and the updated state are

$$\bar{W}^t(k) = \alpha^t W^t(k) \quad (6.4.5-8)$$

$$\boxed{\bar{x}_j^t(k|k) = \hat{x}^t(k|k-1) + \bar{W}^t[k_j(k) - \bar{z}^t(k|k-1)]} \quad (6.4.5-9)$$

The covariance associated with this state conditioned on measurement j being *from target t but merged* is

$$\bar{P}^t(k|k) = [I - \bar{W}^t(k) H^t(k)] P^t(k|k-1) \quad (6.4.5-10)$$

The covariance associated with the combined estimate (6.4.5-1) is

$$\boxed{\begin{aligned} P^t(k|k) &= \beta_{0t}(k) P^t(k|k-1) + \sum_{j=1}^m \beta_{jt}(k) \hat{P}^t(k|k) + \bar{\beta}_{jt}(k) \bar{P}^t(k|k) \\ &\quad + \sum_{j=1}^m \beta_{jt}(k) [\hat{x}_j^t(k|k) \hat{x}_j^t(k|k)' - \hat{x}^t(k|k) \hat{x}^t(k|k)'] \\ &\quad + \sum_{j=1}^m \bar{\beta}_{jt}(k) [\bar{x}_j^t(k|k) \bar{x}_j^t(k|k)' - \hat{x}^t(k|k) \hat{x}^t(k|k)'] \end{aligned}} \quad (6.4.5-11)$$

6.4.6 Application to Crossing Targets

The Scenario

The following problem was simulated:

- Two targets modeled as nearly constant velocity objects moving in a plane
- Initial speed $500m/s$ for each, with courses of 80° and 100°
- The trajectories cross at the middle of the $35s$ tracking period, resulting in severe mutual interference
- Position-only measurements taken at intervals $T = 1s$ with additive noise with $\sqrt{R_{ii}} = 150m$
- Clutter density $1/km^2$
- The process noise entering into the velocity had standard deviation of 2% from the speed
- The targets' signal strengths were equal and their detection probabilities are 0.99
- The resolution cell size was, in each coordinate (four options),

$$\delta_i = n\sqrt{R_{ii}} \quad n = 0, 1, 2, 3, 4 \quad (6.4.6-1)$$

Figure 6.4.6-1 illustrates the performance of the PDA, JPDA and JPDAM, respectively, for $\delta_i = 600m$ — a large resolution cell (low resolution situation).

Figures 6.4.6-2 and 6.4.6-3 compare the tracking capability and position errors for these filters for various resolution cell sizes.

6.4.6 Application to Crossing Targets

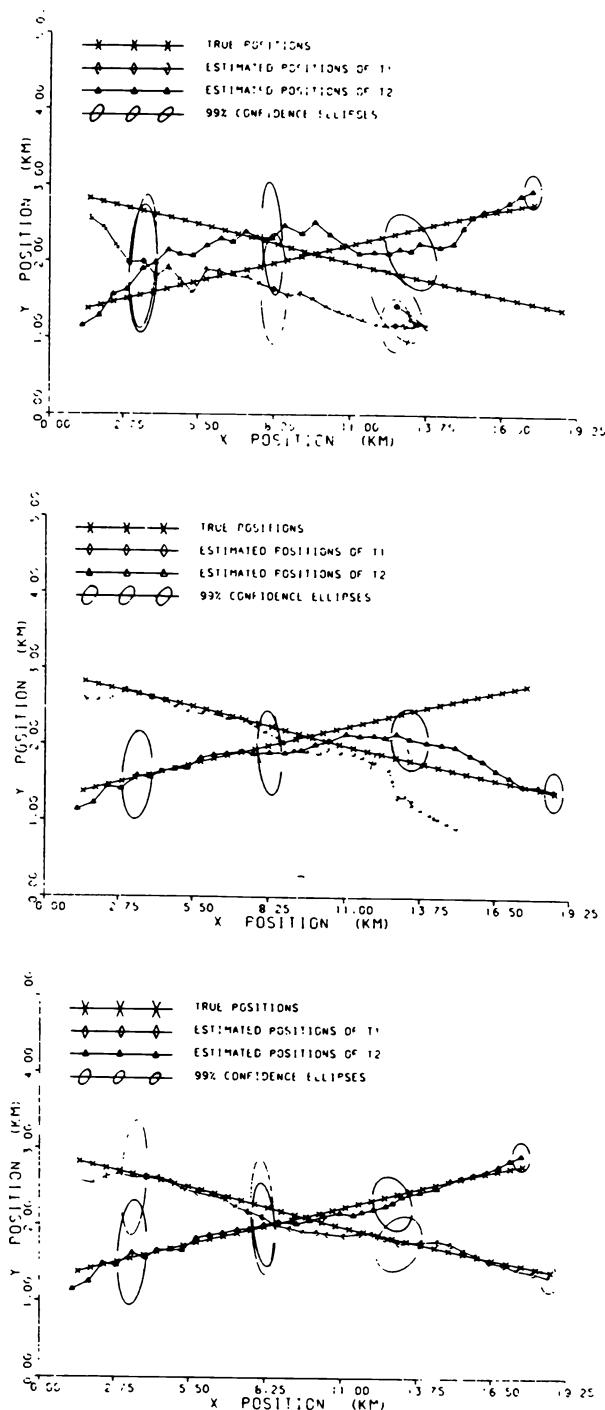


Figure 6.4.6-1: True vs. estimated trajectories — one sample run: (top) PDA; (middle) JPDA; (bottom) JPDAM.

6.4.6 Application to Crossing Targets

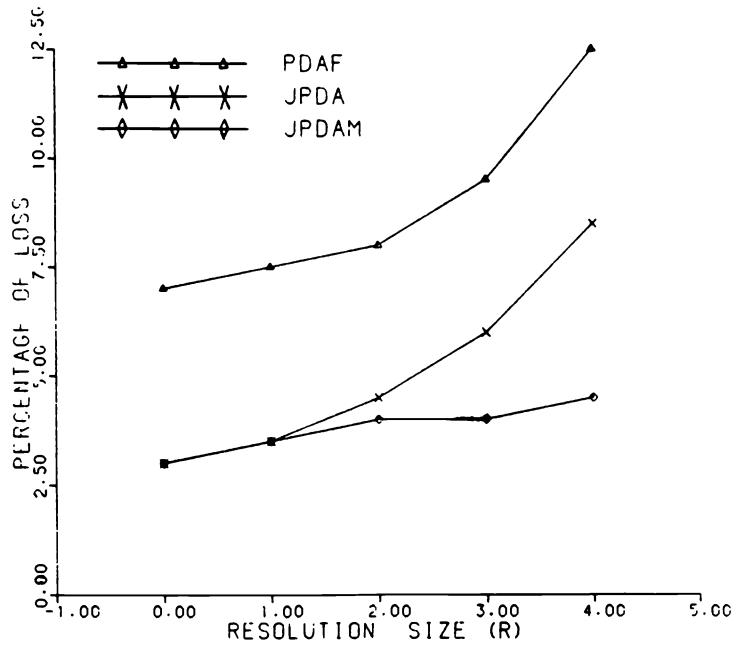


Figure 6.4.6-2: Comparison of the tracking capability of the three filters from 100 runs.

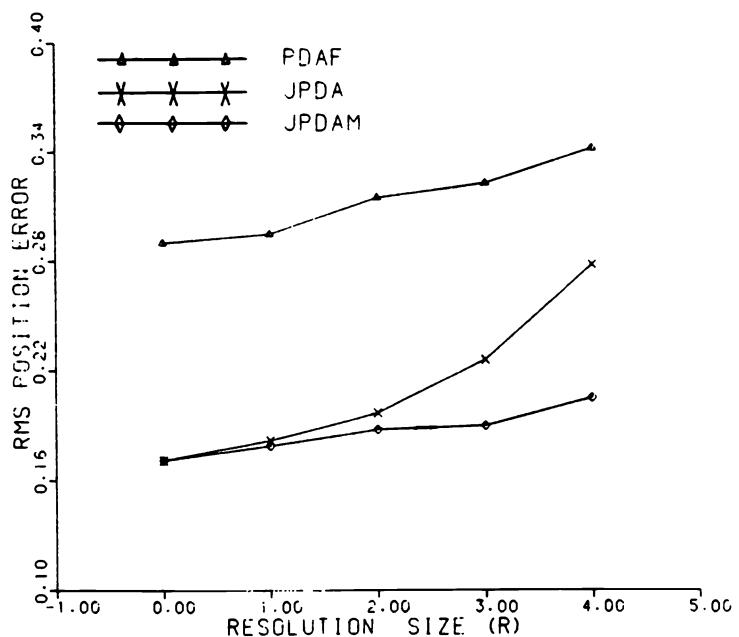


Figure 6.4.6-3: Position estimation errors for the three filters — sample average from 100 runs.

6.4.7 The JPDAM — Summary

The measurements from two targets merge if they fall in the same multidimensional resolution cell.

The pdf of the location of a merged measurement has been derived by modeling it as a *linear combination* of the two noisy measurements with weightings given by the relative signal strengths.

The association of the measurements is done by evaluating the probabilities of all the feasible *joint association events* for the latest set of measurements.

These feasible association events are obtained from the *measurement validation matrix*, by assuming that:

- a measurement could have originated from
 - a single target
 - two targets (unresolved)
 - no target (false-alarm/clutter)
- a target can give rise to
 - one measurement
 - no measurement

From the probabilities of *joint association events* one obtains the probabilities of the *marginal* (individual measurement to individual target) associations.

The state update is done as in the (decoupled) PDAF with each measurement weighted by its association probability to the target under consideration with the following difference:

The *filter gain* is

- the standard one if the measurement is assumed *resolved*
- modified (smaller) if the measurement is assumed to have originated from two targets — *merged (unresolved)* — because in this case it carries less information about each single target.

Alternatively, a coupled filter could be used — in this case the update is done directly based on the joint association events.

6.4.8 The PDF of the Merged Measurements

The pdf of

$$z_0 = \alpha z_1 + (1 - \alpha) z_2 \quad (6.4.8-1)$$

without the condition that the measurements fall in the same cell is normal with mean

$$\hat{z}_0 = \alpha \hat{z}_1 + (1 - \alpha) \hat{z}_2 \triangleq [\hat{z}_{01} \ \cdots \ \hat{z}_{0n}]' \quad (6.4.8-2)$$

and covariance

$$S_0 = \alpha^2 S_1 + (1 - \alpha)^2 S_2 = \text{diag}[\sigma_{0i}^2] \quad (6.4.8-3)$$

With these notations this unconditional pdf is

$$p(z_0) = \mathcal{N}(z_0; \hat{z}_0, S_0) = \prod_{i=1}^n \mathcal{N}(z_{0i}; \hat{z}_{0i}, \sigma_{0i}^2) \triangleq \prod_{i=1}^n p(z_{0i}) \quad (6.4.8-4)$$

The difference between two measurements

$$w \triangleq z_1 - z_2 \triangleq [w_1 \ \cdots \ w_n]' \quad (6.4.8-5)$$

has the pdf

$$p(w) = \mathcal{N}(w; \hat{w}, S_w) = \prod_{i=1}^n \mathcal{N}(w_i; \hat{w}_i, \sigma_{wi}^2) \quad (6.4.8-6)$$

where

$$\hat{w} = \hat{z}_1 - \hat{z}_2 = [\hat{w}_1 \ \cdots \ \hat{w}_n]' \quad (6.4.8-7)$$

$$S_w = S_1 + S_2 = \text{diag}[\sigma_{wi}^2] \quad (6.4.8-8)$$

Denote

$$\omega_i^+ \triangleq \frac{\hat{w}_i + \delta_i}{\sigma_{wi}} \quad (6.4.8-9)$$

$$\omega_i^- \triangleq \frac{\hat{w}_i - \delta_i}{\sigma_{wi}} \quad (6.4.8-10)$$

$$\omega_i \triangleq \frac{\hat{w}_i}{\sigma_{wi}} \quad (6.4.8-11)$$

The **prior merging probability** is the product of the corresponding probabilities P_{M_i} for each component

$$P_M \triangleq P\{z_1, z_2 \in C\} = \prod_{i=1}^n P\{z_{1i}, z_{2i} \in C_i\} \triangleq \prod_{i=1}^n P_{M_i} \quad (6.4.8-12)$$

6.4.8 The PDF of the Merged Measurements

The *prior merging probability for component i* is [CB84]

$$P_{M_i} = \frac{1}{2}[\text{erf}(\omega_i^+) - \text{erf}(\omega_i^-)] + \frac{\hat{w}_i}{2\delta_i}[\text{erf}(\omega_i^+) + \text{erf}(\omega_i^-) - 2\text{erf}(\omega_i)] \\ + \frac{\sigma_{w_i}}{\delta_i(2\pi)^{1/2}}[e^{-(\omega_i^+)^2/2} + e^{-(\omega_i^-)^2/2} - 2e^{-(\omega_i)^2/2}] \quad (6.4.8-13)$$

where

$$\text{erf}(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_{-x}^x e^{-z^2/2} dz \quad (6.4.8-14)$$

The *conditional merging probability* (conditioned on z_0) is the product of the corresponding probabilities Q_{M_i} for each component

$$Q_M(z_0) \triangleq P\{z_1, z_2 \in C | z_0\} = \prod_{i=1}^n P\{z_{1i}, z_{2i} \in C_i | z_{0i}\} \triangleq \prod_{i=1}^n Q_{M_i}(z_{0i}) \quad (6.4.8-15)$$

To obtain the above, the conditional pdf of w given z_0 is needed. This follows by observing that

$$w = z_1 - z_2 = \frac{z_0 - z_2}{\alpha} \quad (6.4.8-16)$$

Then it follows that

$$p(w|z_0) = \prod_{i=1}^n \mathcal{N}(w_i; m_i, \tau_i^2) \quad (6.4.8-17)$$

Using the expression of the conditional mean of one Gaussian random variable in terms of another and denoting with κ_i the corresponding “gain,” one has

$$m_i \triangleq E[w_i | z_{0i}] = E[w_i] + \kappa_i(z_{0i} - \hat{z}_{0i}) = \hat{z}_{1i} - \hat{z}_{2i} + [\alpha\sigma_{1i}^2 - (1 - \alpha)\sigma_{2i}^2]\sigma_{0i}^{-2}(z_{0i} - \hat{z}_{0i}) \quad (6.4.8-18)$$

The conditional variance is

$$\tau_i^2 \triangleq \text{var}[w_i | z_{0i}] = \sigma_{1i}^2 + \sigma_{2i}^2 - \kappa_i^2\sigma_{0i}^2 = \sigma_{1i}^2 + \sigma_{2i}^2 - [\alpha\sigma_{1i}^2 - (1 - \alpha)\sigma_{2i}^2]^2\sigma_{0i}^{-2} \quad (6.4.8-19)$$

Define

$$\mu_i^+ \triangleq \frac{m_i + \delta_i}{\tau_i} \quad (6.4.8-20)$$

$$\mu_i^- \triangleq \frac{m_i - \delta_i}{\tau_i} \quad (6.4.8-21)$$

$$\mu_i \triangleq \frac{m_i}{\tau_i} \quad (6.4.8-22)$$

Similarly to (6.4.8-13), one obtains *the conditional merging probability* for component i as

$$Q_{M_i}(z_{0i}) = \frac{1}{2}[\text{erf}(\mu_i^+) - \text{erf}(\mu_i^-)] + \frac{m_i}{2\delta_i}[\text{erf}(\mu_i^+) + \text{erf}(\mu_i^-) - 2\text{erf}(\mu_i)] \\ + \frac{\tau_i}{\delta_i(2\pi)^{1/2}}[e^{-(\mu_i^+)^2/2} + e^{-(\mu_i^-)^2/2} - 2e^{-(\mu_i)^2/2}] \quad (6.4.8-23)$$

6.4.8 The PDF of the Merged Measurements

Thus, the pdf of the merged measurement (6.4.2-7) is obtained as

$$g_{12}(z_0) = \prod_{i=1}^n p(z_{0i}) \frac{Q_{M_i}(z_{0i})}{P_{M_i}} \quad (6.4.8-24)$$

where the terms on the r.h.s. above are given by (6.4.8-4), (6.4.8-23) and (6.4.8-13), respectively.

The estimation of the state requires the *predicted mean and variance of the merged measurement*.

The mean associated with the pdf (6.4.8-24) is [CB84]

$$\bar{z}_{0i} = \hat{z}_{0i} + \frac{\kappa_i \sigma_{0i}^2}{2 P_{M_i} \delta_i} [\operatorname{erf}(\omega_i^+) + \operatorname{erf}(\omega_i^-) - 2\operatorname{erf}(\omega_i)] \quad (6.4.8-25)$$

where notations (6.4.8-9)–(6.4.8-11) have been used and κ_i is defined in (6.4.2-12).

The variance associated with (6.4.8-24) is

$$\bar{\sigma}_{0i}^2 = \sigma_{0i}^2 (1 + d_i) - (\bar{z}_{0i} - \hat{z}_{0i})^2 \quad (6.4.8-26)$$

where, using definitions (6.4.8-9)–(6.4.8-11),

$$d_i \triangleq \frac{\kappa_i^2 \sigma_{0i}^2}{2 P_{M_i} \delta_i \sigma_{w_i}} (2/\pi)^{1/2} [e^{-(\omega_i^+)^2/2} + e^{-(\omega_i^-)^2/2} - 2e^{(\omega_i)^2/2}] \quad (6.4.8-27)$$

An alternative to using the (quite complicated) pdf (6.4.8-24) of the merged measurement is a *Gaussian approximation* of it with the moments as given by (6.4.8-25) and (6.4.8-26). However these equations are still too complicated.

As shown in [CCB86] the following, simple to implement, approximation of the moments (6.4.8-25) and (6.4.8-26) is very accurate

$$\bar{z}_{0i} \approx \bar{z}_{ai} \triangleq \hat{z}_{0i} - \frac{\kappa_i \sigma_{0i}^2 \hat{w}_i}{\sigma_{w_i}^2} e^{-c \frac{\delta_i}{\sigma_{w_i}}} \quad (6.4.8-28)$$

$$\bar{\sigma}_{0i}^2 \approx \bar{\sigma}_{ai}^2 \triangleq \sigma_{0i}^2 - \frac{\kappa_i^2 \sigma_{0i}^4}{\sigma_{w_i}^2} e^{-2c \frac{\delta_i}{\sigma_{w_i}}} \quad (6.4.8-29)$$

where c is a coefficient determined experimentally.

With $c = 0.25$ approximation (6.4.8-28) for the mean has about 1% error, while approximation (6.4.8-29) yields about 3% error for the standard deviation.

6.4.9 Evaluation of the Association Probabilities

6.4.9 Evaluation of the Association Probabilities

The **joint association event probabilities** for $N_T = 2$ targets and m measurements with a possible **merged (unresolved) measurement** are evaluated. Using Bayes' formula, the conditional probability of a joint event at time k is

$$P\{\theta|Z^k\} = \frac{1}{c} p[Z(k)|\theta, Z^{k-1}] P\{\theta|Z^{k-1}\} \quad (6.4.9-1)$$

where c is the normalization constant.

The first term on the r.h.s. above is

$$p[Z(k)|\theta, Z^{k-1}] = \prod_{j=1}^m p[z_j(k)|\theta, Z^{k-1}] \quad (6.4.9-2)$$

The origin-conditioned pdf of a measurement is

$$p[z_j(k)|\theta, Z^{k-1}] = \begin{cases} g_{1,2}[z_j(k)] & \text{if } \tau_j(\theta) = 2 \\ f_{t_j}[z_j(k)] & \text{if } \tau_j(\theta) = 1 \\ V^{-1} & \text{if } \tau_j(\theta) = 0 \end{cases} \quad (6.4.9-3)$$

where

- $g_{1,2}$ is the pdf of the merged measurement from targets 1 and 2 at time k , given their relative power,
- f_{t_j} is the pdf of the measurement from target t_j , to which measurement j is associated,
- V is the volume of the surveillance region in which a measurement, if false, is assumed uniformly distributed.

The second term on the r.h.s. of (6.4.9-1) is the *prior to the current time* probability of a joint event.

The total number of false measurements in event θ under consideration is

$$\phi(\theta) = \sum_{j=1}^m \{1 - \min[1, \tau_j(\theta)]\} \quad (6.4.9-4)$$

Then the number of events in which the *same set of targets is detected* is given by the number of permutations of the m measurements taken as the number of target detections $m - \phi(\theta)$, i.e.,

$$P_{m-\phi}^m = \frac{m!}{\phi!} \quad (6.4.9-5)$$

6.4.9 Evaluation of the Association Probabilities

With this, the prior probability of a joint event is

$$P\{\theta|Z^{k-1}\} = \frac{\phi!}{m!} \prod_{t=1}^2 \left[(P_D^t)^{\delta_t} (1 - P_D^t)^{1-\delta_t} \right] (P_M)^{1-\rho} (1 - P_M)^\rho \mu_F(\phi) \quad (6.4.9-6)$$

where

- P_D^t is the probability of detection of target t ,
- P_M is the prior (to the current time) probability that two measurements are *merged*, given in (6.4.8-12),
- $\mu_F(\cdot)$ is the pmf of the number of *false* measurements in the surveillance region of volume V , which can be assumed as Poisson distributed with spatial density λ , in which case

$$\mu_F(\phi) = e^{\lambda V} \frac{(\lambda V)^\phi}{\phi!} \quad (6.4.9-7)$$

Combining (6.4.9-2), (6.4.9-3) and (6.4.9-6) with (6.4.9-7) into (6.4.9-1) yields the final expression of the probabilities of the joint events (6.4.4-1).

6.5 BIBLIOGRAPHICAL NOTES

The JPDAF from Section 6.2 was first presented in [Bar74] but it used incorrect prior probabilities. This was corrected in [BFS80] and the final version of this algorithm is from [FBS80, FBS83]. The implementation of the JPDAF, with some simplifications and changes, in a microprocessor for real-time application is presented in [Fit86] and Chapter 2 of [Bar90]. Another real-time implementation is discussed in [BW87], where the idea of the JPDACF (Subsection 6.2.6) was put forward.

The “measurement-oriented” Multiple Hypothesis Tracker from Section 6.3 is based on [Rei77, Rei79a]. The applications of this algorithm to targets moving over terrain and ocean surveillance were discussed in [Rei79a, Rei79b], respectively. Further work that generalized this approach is [MCWT83, MC85, MCTW86]. A “track-oriented” approach combined with multiple models has been presented in [WKBW84, KW85] and Chapter 3 of [Bar90]. Combinatorial issues related to the multiple hypotheses filter are discussed in [NCS87b, NCS87a]. Other issues related to MHT, including the “coordinated presentation” of the results, are discussed in Chapter 8 of [Bar90].

The explicit use of the state pdf in measurement association for track formation was reported in [She82]; the propagation of the state pdf for bearings-only observations using “containment regions” is described in [BS86].

The systematic treatment of unresolved targets has started with [TW81]. Further modeling work was done in [NB81], which was generalized in [CB84] and applied to tracking, resulting in the JPDAM. Section 6.4 is based on [CB84, CB86].

Chapter 7

MULTIPLE TARGETS IN CLUTTER: NON-BAYESIAN APPROACHES

7.1 INTRODUCTION

7.1.1 The Non-Bayesian Approach: an Assignment Problem

The non-Bayesian approach is characterized by a **hard association**, e.g., **assignment** of a sequence of measurements together or of a measurement to a track (a sequence of measurements already associated), based on a certain decision criterion.

There are several possible decision criteria. A decision or assignment criterion can deal with

- the cumulative set of measurements from the beginning (track formation)
- measurements from a sliding window (to be assigned to an existing track — previously associated measurement sequence up to the time just before the window)
- measurements from the latest time (to be assigned to an existing track — a previously associated measurement sequence up to the previous time).

7.1.1 The Non-Bayesian Approach: an Assignment Problem

The set of measurements from one sampling time (scan or frame) constitute a *list*¹.

The assignment of a set of measurements from k times (scans or frames) into sequences amounts to having k lists (one for each time) and finding sequences with (at most) one element from each list. This is a **k -D (k -dimensional) assignment problem**, which is **NP-hard** for $k > 2$ — its numerical complexity is *not polynomial* in the size of the problem (the length of the lists). The complexity is factorial in the size of the problem.

A 2-D assignment problem, which requires “matching” of items from 2 lists (assignment of one item from the first list to one item from the second list) has polynomial (actually cubic) complexity in the size of the problem. An efficient algorithm for this 2-D assignment problem, called the **modified auction** can be found in [PDBW92].

There are several decision criteria that can be used in obtaining the assignment:

- The likelihood function of a track — the pdf of the sequence of measurements under consideration (as in Section 3.3). This is not a global approach — it can yield more than one assignment for a measurement.
- The joint likelihood function of a set of tracks — the joint pdf of all the sequences of measurements under consideration (see Section 7.2). This approach is global and amounts to a k -D assignment problem.
- The sum of “distances” between the latest measurements and the previously established tracks (log-likelihood functions or normalized innovations). This approach (global, but limited to the latest measurements) is a 2-D assignment problem.

¹Actually, for **scanning sensors**, each measurement within a scan has its own **time tag** within the sampling interval. Only for **staring sensors** do all the measurements from a frame share the same time tag. Explicit accounting for the different time tags of the measurements from the same scan is carried out in Section 8.7.

7.1.2 Outline

An algorithm that evaluates the *marginal likelihood* of each *sequence of measurements* (track) was discussed in Section 3.3. The extension of this approach that assigns measurements to tracks by considering simultaneously all the feasible partitions of the measurements via the *joint likelihood function* of all the tracks is presented in Section 7.2. The application of the joint ML assignment to the track formation for several ballistic objects observed from a *single* orbiting satellite based passive sensor is also discussed.

A technique for initiating tracks in a multitarget environment, based on a simple logic combined with polynomial fitting, is presented in Section 7.3.

Section 7.4 presents a *composite logic* for track formation and shows how one can evaluate the system-level performance — target track acceptance probability and false track probability — based on the single-look target detection probability and the per cell per look false alarm probability.

7.1.3 Summary of Objectives

7.1.3 Summary of Objectives

Present the non-Bayesian approaches — without prior probabilistic model of the environment:

- The joint **maximum likelihood (ML)** approach — considers simultaneously all the feasible partitions of the measurements into the tracks
- A technique for initiating tracks in a multitarget environment, based on a simple logic combined with polynomial fitting.

Apply the joint ML technique to the track formation of several ballistic objects using data from a single orbiting satellite based passive sensor.

Evaluate the system-level performance

- target track detection probability
- false track probability

for a composite logic initiation technique based on

- target detection probability
- false alarm probability.

7.2 A JOINT LIKELIHOOD FUNCTION METHOD FOR TRACK FORMATION VIA ASSIGNMENT

7.2.1 The Model and Assumptions

In this method it is assumed that there are a number of targets and each target's state evolves according to the *same* state equation as in (3.1.3-1).

The observation from a target is given by the standard measurement equation (3.1.3-2).

The measurements not having originated from any target are modeled as i.i.d. with uniform pdf V^{-1} , where V is the hypervolume of the surveillance region.

The number of targets is not known.

The set of measurements at time k is

$$Z(k) = \{z_i(k)\}_{i=1}^{m(k)} \quad (7.2.1-1)$$

This can be seen as the “snapshot” of the whole surveillance region at time k .

The cumulative set (actually, a sequence of sets) of measurements from the whole surveillance region up to and including time k is

$$Z^k = \{Z(j)\}_{j=1}^k \quad (7.2.1-2)$$

The total number of measurements in the above set is

$$N = \sum_{j=1}^k m(j) \quad (7.2.1-3)$$

7.2.2 Partitioning of the Measurements

7.2.2 Partitioning of the Measurements

A sequence of measurements up to time k is denoted as

$$Z^{k,l} = \{z_{i,l}(j)\}_{j=1}^k \quad (7.2.2-1)$$

and the corresponding event

$$\theta^{k,l} = \{Z^{k,l} \text{ is a true track}\} \quad l = 1, \dots, L \quad (7.2.2-2)$$

Only the sequences (7.2.2-1) whose likelihood functions are “acceptable” are considered. This implies that the negative log-likelihood function corresponding to each **candidate track** (7.2.2-1) is below a threshold as discussed in Section 3.3 — it is acceptable if considered alone (i.e., ignoring the competition).

Denote by

$$\phi = \{Z^{k,l_i}\}_{i=0}^I \quad (7.2.2-3)$$

a **feasible partition** of the set Z^k into acceptable tracks $Z^{k,l}$. The set $Z^{k,0}$ contains the measurements not associated into any track in the partition under consideration.

The feasibility of the partition requires that

- (i) each measurement belong to a track

$$Z^k = \bigcap_{i=0}^I Z^{k,l_i} \quad (7.2.2-4)$$

and

- (ii) each measurement belong to one track only

$$Z^{k,l_i} \cap Z^{k,l_j} = \emptyset \quad (7.2.2-5)$$

Figures 7.2.2-1 and 7.2.2-2 illustrate some of the possible partitions.

The problem thus consists of associating one item from each of the k “lists” (scans or frames or snapshots) given in (7.2.1-1).

7.2.2 Partitioning of the Measurements

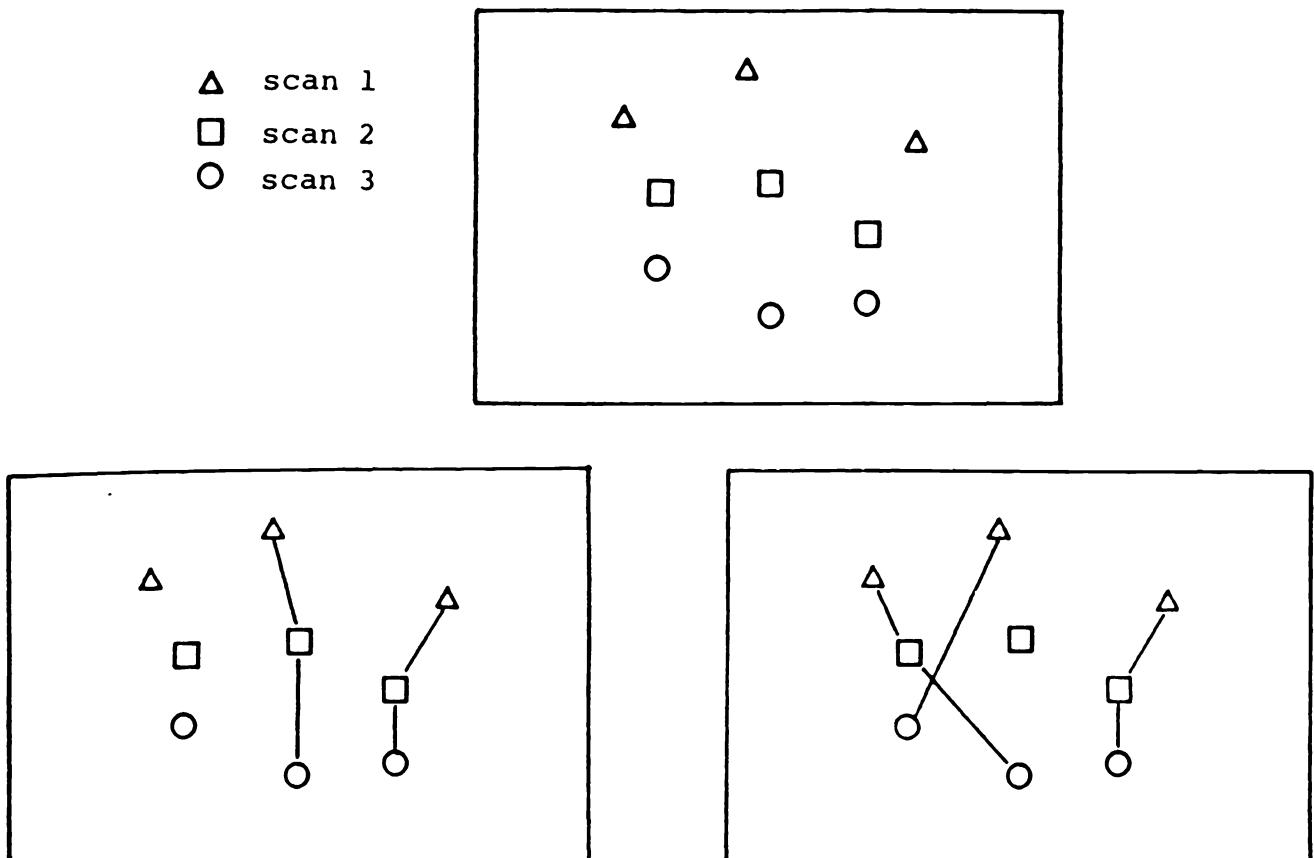


Figure 7.2.2-1: Some feasible partitions.

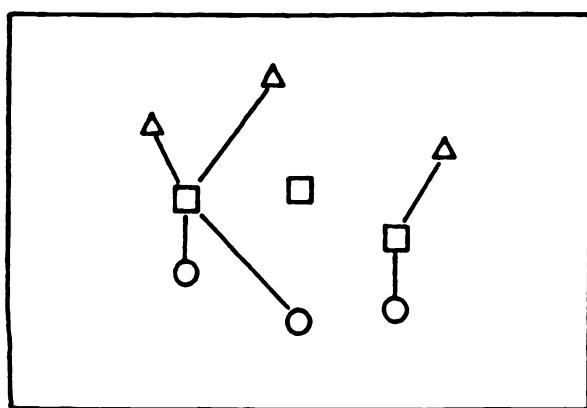


Figure 7.2.2-2: An unfeasible partition.

7.2.2 Partitioning of the Measurements

Remark

The **partition** of a data set amounts to **assignment** of the measurements to various tracks. The **dimension of the assignment problem** is k — one has to associate items from k lists into sequences of measurements, with (at most) one from each list and at most one assignment for a measurement.

The Maximum Likelihood Criterion for Partitioning

Corresponding to partition ϕ one has the event

$$\theta(\phi) = \{\text{partition } \phi \text{ is true}\} \quad (7.2.2-6)$$

The set of all feasible partitions is

$$\Phi = \{\phi\} \quad (7.2.2-7)$$

The **most likely partition** of the data set Z^k into tracks is obtained by *maximizing the joint likelihood function of all the measurements* over all the feasible partitions into acceptable tracks:

$$\max_{\phi \in \Phi} p[Z^k | \theta(\phi)] \quad (7.2.2-8)$$

In the sequel, the maximization (7.2.2-8) is shown to be equivalent to a constrained zero-one programming (assignment) problem.

Interpretation of the ML Criterion

The candidate tracks are supposed to be acceptable if considered alone, i.e., regardless of the *competition for measurements among the tracks*.

The joint ML criterion *accounts for this competition* by considering only feasible partitions.

7.2.3 The Joint Likelihood Function and the Likelihood Ratio

7.2.3 The Joint Likelihood Function and the Likelihood Ratio

The **negative log-likelihood function** $\lambda^{k,l}$ of each acceptable track $Z^{k,l}$ is

$$\boxed{\lambda^{k,l} = -\log p[Z^{k,l}|\theta^{k,l}]} \quad (7.2.3-1)$$

Consider the following **log-likelihood ratio**

$$\boxed{\tilde{\lambda}^{k,l} = -\log \frac{p[Z^{k,l}|\theta^{k,l}]}{p[Z^{k,l}|\text{all false}]} = \lambda^{k,l} + N_l \log V^{-1}} \quad (7.2.3-2)$$

where N_l is the number of measurements in the sequence $Z^{k,l}$.

Note that (7.2.3-2) is a (physically) *dimensionless quantity*.

Denote the **likelihood ratio vector** with elements (7.2.3-2) as

$$\tilde{\lambda} = [\tilde{\lambda}^{k,1} \dots \tilde{\lambda}^{k,L}]' \quad (7.2.3-3)$$

Let the **partition indicator vector** ρ be a binary vector of dimension L that indicates which of the acceptable tracks belong to a feasible partition. To each partition ϕ corresponds a vector ρ .

Let the **track indicator vector** ψ^l be a binary vector of dimension N , the total number of measurements in the set Z^k , with unity elements corresponding to the measurements that belong to the track $Z^{k,l}$ and the others zero.

Define the **measurement indicator matrix** A as the $N \times L$ matrix whose columns are the track indicator vectors

$$A = [\psi^1 \dots \psi^L] \quad (7.2.3-4)$$

The Feasibility Conditions as Constraints

The feasibility conditions (7.2.2-4) and (7.2.2-5) for a partition ϕ can be expressed in terms of the corresponding partition indicator vector ρ and the measurement indicator matrix A as the *inequality constraints*

$$A\rho \leq \mathbf{1} \quad (7.2.3-5)$$

where $\mathbf{1}$ is the N -vector of ones.

7.2.3 The Joint Likelihood Function and the Likelihood Ratio

The Log-Likelihood Function and the Partition Vector

The negative log-likelihood function of all the measurements is, for the partition ϕ under consideration,

$$-\log p[Z^k|\theta(\phi)] = \sum_{L=1}^L \rho_l \lambda^{k,l} + \lambda^{k,0}(\phi) \quad (7.2.3-6)$$

where ρ_l is the l -th element of ρ and the likelihood function of the unassociated measurements is

$$\begin{aligned} \lambda^{k,0}(\phi) &= -N_0 \log V^{-1} \\ &= -(N - \sum_{l=1}^L \rho_l N_l) \log V^{-1} \end{aligned} \quad (7.2.3-7)$$

Now (7.2.3-6) can be rewritten as

$$\begin{aligned} -\log p[Z^k|\theta(\phi)] &= \sum_{l=1}^L \rho_l (\lambda^{k,l} + N_l \log V^{-1}) - N \log V^{-1} \\ &= \sum_{l=1}^L \rho_l \tilde{\lambda}^{k,l} - N \log V^{-1} \\ &= \rho' \tilde{\lambda} - N \log V^{-1} \end{aligned} \quad (7.2.3-8)$$

Since the last term in (7.2.3-8) is a constant, the *maximization* of the likelihood function is equivalent to the *minimization* of the inner product of

- the partition vector and
- the likelihood ratio vector.

Therefore, the maximum likelihood set of tracks is obtained from

$$\boxed{\min_{\rho} \rho' \tilde{\lambda}} \quad (7.2.3-9)$$

subject to the following constraint on the partition indicator vector ρ

$$\boxed{A\rho \leq \mathbf{1}} \quad (7.2.3-10)$$

This has converted the track formation problem into a **zero-one programming** problem.

Remarks

1. While the zero-one programming (or assignment) problem is not a trivial one computationally, standard packages can be used for modest size problems. The computational requirements, however, can become prohibitive, since the complexity of this combinatorial optimization problem increases, if the dimension of the assignment problem is greater than 2, approximately as the factorial of the size of the problem — it is **NP-hard**, i.e., it has **non-polynomial complexity**.
2. The above development assumed *unity probability of detection*; otherwise incomplete sequences of measurements have to be considered as well. The extension to the situation where $P_D < 1$ is discussed in the next subsection.
3. This is a batch procedure suitable for track initialization of several targets.
4. The formulation of this problem of

- association of measurements from a *single sensor across time*

is equivalent to the problem of

- association of measurements from *several sensors from the same time*.

For details, as well as a fast numerical technique (for a 3-D assignment problem), see [PDBW92] and Section 8.6.

The most general multidimensional assignment problem is when one has to carry out

- association of measurements from *several sensors from arbitrary times*.

This is discussed in [DPBY94] and, briefly in Subsection 8.9.1.

7.2.4 Extension to Incomplete Tracks

A possibly **incomplete track** is characterized by the **detection indicator sequence**

$$\delta^{k,l} = [\delta^l(1) \ \dots \ \delta^l(k)]' \quad (7.2.4-1)$$

where

$$\delta^l(j) = \begin{cases} 1 & \text{if track } \theta^{k,l} \text{ is detected at time } \kappa \\ 0 & \text{otherwise} \end{cases} \quad (7.2.4-2)$$

The **likelihood function of a possibly incomplete track** is the mixed pdf-probability of the measurements and the detection sequence

$$P[Z^{k,l}, \delta^{k,l} | \theta^{k,l}] = P[Z^{k,l} | \theta^{k,l}] \prod_{\kappa=1}^k (P_D)^{\delta^l(\kappa)} [(1 - P_D)]^{1 - \delta^l(\kappa)} \quad (7.2.4-3)$$

The **likelihood ratio of a possibly incomplete track** is, similarly to (7.2.3-2),

$$\tilde{\lambda}^{k,l} = -\log \frac{p[Z^{k,l}, \delta^{k,l} | \theta^{k,l}]}{p[Z^{k,l} | \text{all false}]} \quad (7.2.4-4)$$

Remarks

One cannot compare likelihood functions with different number of measurements because they have *different physical dimensions*. However, likelihood ratios as in (7.2.4-4) are *dimensionless*, and, thus, they can be used to compare sequences of measurements of different lengths.

Further extensions of the likelihood function approach (that include new targets, track duration) can be found in [Bla86]. However, the general formulation of the problem cannot be recast in a zero-one programming form.

7.2.5 Track Formation for Ballistic Objects from a Single Passive Sensor

The following problem is considered:

- A unknown number of missiles on ballistic trajectories are observed by a passive sensor on an orbiting satellite
- The sensor yields azimuth and elevation observations with a certain target detection probability (less than unity) and has false alarms
- A small number of frames (or scans) are available from this sensor
- It is desired to find the number of targets and their state estimates — position and velocity.

In order to estimate the entire 6-dimensional state of such a target, the minimum number of measurements (of dimension 2) needed is thus 3. The key to **target state observability** is the motion of the orbiting satellite — the numerical conditioning of the problem is, in general, still marginal.

The association problem for s sets of measurements from one sensor (or from several sensors) yields an s -D assignment problem. In practice, in order to obtain reliable association in the present single sensor problem, the number of frames or scans (measurements over time) needed is larger than 3.

Thus an s -D assignment algorithm is necessary in order to solve this problem. Such an algorithm, which relies on a series of successive relaxations and Lagrangian multiplier updates, is presented in [DPB92a].

The assignment algorithm which solves this **multiple target track formation from a single passive sensor**, also applies to the problem of **measurement association from multiple synchronous sensors**.

Example

A cluster of 5 targets are located at $t_0 = 0$ at $[7000 \ 2000 \ 2000]$ (in km, in Earth centered rotating coordinates) with an average intertarget separation of 400. The target velocities are $[2.1 \ 4.5 \ -2.8]$ (in km/s) plus a random term $\mathcal{U}(-0.5, 0.5)$ in each component.

The sensor is on an orbiting satellite at $[9000 \ 0 \ 0]$ with a velocity of $[0 \ 6 \ 2.9]$ at t_0 . The period between frames is $T = 10$ s and all the measurements are taken at the same time, i.e., one has a **staring sensor**. The problem can be solved similarly for a **scanning sensor**.

The sensor field of view is 10° , with 3000×3000 resolution cells and the azimuth and elevation measurement accuracies are $1/3$ of a cell — $20\mu\text{rad}$.

The target detection probability is 0.95 and the false alarm probability is $2 \cdot 10^{-7}$ per cell, yielding on the average 2 false reports per frame.

Table 7.2.5-1 gives the simulation results for various numbers of frames. The search was started at the initial estimate obtained from Laplace's method based on the first 3 measurements.

Number of frames s	5	6	8	10
Association accuracy	66%	88%	94%	94%
Position RMS error	194	126	48	26
Velocity RMS error	0.54	0.36	0.12	0.03

Table 7.2.5-1: Simulation results (100 Monte Carlo runs).

Conclusion

In spite of the poor conditioning of this problem, a minimum of 8 frames (total time span of 70s) yield reliable results in this example.

7.2.6 Track Formation with Joint Likelihood Function — Summary

Assumptions

- Tracks are to be formed from sets of measurements taken at several sampling times
- At each sampling time the number of measurements is arbitrary
- There is an unknown number of targets
- Each target's state evolves according to the same equation
- The detection probability of each hypothesized target is unity (extension to $P_D < 1$ also available)
- Measurements that did not originate from a target are independently and uniformly distributed in the surveillance region.

The algorithm

- All the (individually) acceptable tracks are obtained
- A *feasible partition* of the *set of acceptable tracks* is a subset that contains tracks with no common measurements — it is assumed that a measurement can belong to at most one track
- The search for the *most likely feasible partition* is done by maximizing the *joint likelihood function of all the measurements*
- This maximization is equivalent to a zero-one programming (assignment) with a linear constraint that reflects the partition feasibility requirement.

The formulation of this single-sensor, multiple-points-in-time problem is equivalent to the multiple-sensor, single-point-in-time association problem.

7.3 A LOGIC-BASED MULTITARGET TRACK INITIATOR

7.3.1 Track Start-Up

In this approach tracks are started by associating measurements from the first scan (or “frame” for an optical sensor) to each candidate from the second scan using an **acceptance region** or **association region**.

This region accounts for the measurement noise variance and the motion of the target, characterized by a maximum and a minimum speed, denoted as v_{\max}^l and v_{\min}^l , respectively, for coordinate l .

Measurement Association from the First Two Scans

Let $z_i^l(k)$ be component l of measurement i at time $k = 1, 2$. The **distance vector** d_{ij} between measurements $z_i(1)$ and $z_j(2)$ is defined as having components $l = 1, \dots, n_z$

$$d_{ij}^l = [z_j^l(2) - z_i^l(1) - v_{\max}^l \Delta]^+ + [z_j^l(2) - z_i^l(1) - v_{\min}^l \Delta]^- \quad (7.3.1-1)$$

where Δ is the time interval between the scans and

$$[x]^+ \triangleq \max(x, 0) \quad (7.3.1-2)$$

$$[x]^- \triangleq \max(-x, 0) \quad (7.3.1-3)$$

Assuming the measurement errors to be independent, normal and zero-mean with covariance $R_i(k)$, define the following normalized distance squared

$$D_{ij} = d_{ij}' [R_i(1) + R_j(2)]^{-1} d_{ij} \quad (7.3.1-4)$$

The test for associating $z_i(1)$ to $z_j(2)$ is then

$$D_{ij} \leq \gamma \quad (7.3.1-5)$$

where the threshold γ for “probability of miss” α is such that

$$\chi_{n_z}^2 (1 - \alpha) = \gamma \quad (7.3.1-6)$$

A “General/Parallel” Search for Track Formation

Following the association of measurements from the first two scans, a “general/parallel” search shown in Figure 7.3.1-1 is used to associate (“correlate”) measurements from subsequent scans. The main features of this search are:

- It relies on polynomial fitting as described later
- It discards tracks with missed detections above a certain number
- Each unassociated new detection is used to start a new track.

The General Search

1. Starting with an **initiator** — a measurement (detection) from the first scan (frame) — an acceptance region is set up for the second scan according to (7.3.1-5). For every measurement from the second scan falling into this region a potential track is set up, i.e., one utilizes **track split**.
2. For every potential track, consisting of two measurements, a straight line extrapolation (first order polynomial) is used to the third sampling time. The size of the acceptance region for measurements from the third scan is determined as in (3.5.3-4) of [BL93], possibly increased by allowing for some acceleration.
3. In the third scan, if no measurement is found in the acceptance region corresponding to a potential track, then this track is dropped. If more than one measurement is validated for a potential track, then this track is split. A prediction into the fourth scan can be done with a second order polynomial or again with a first order polynomial if the estimated acceleration is statistically insignificant, as discussed in Section 3.6.1 of [BL93].
4. This procedure is continued for 5–8 scans, depending on target density. At that point, hopefully, only a few potential tracks will be left. If this is not the case, this procedure might not be suitable.
5. The final track selection is done then by testing the residual for the polynomial fit via the chi-square test as in Section 3.6.1 of [BL93]. The order of the polynomial depends on the specific application.

Remark

An additional technique, called “pattern match method,” can be used to resolve ambiguities in associating measurements. This pertains primarily to the track continuation stage and it will be discussed later.

7.3.1 Track Start-Up

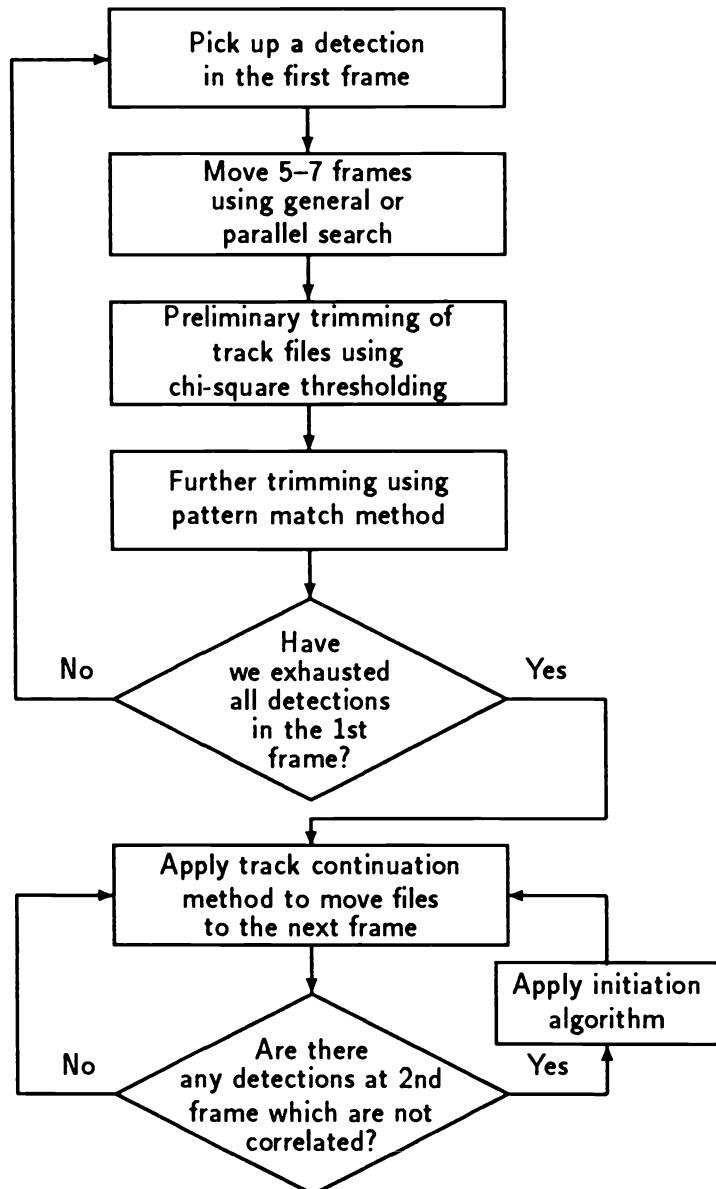


Figure 7.3.1-1: Track initiation logic.

7.3.1 Track Start-Up

The Parallel Search

6. The **parallel search** assumes that the targets of interest in the neighborhood of an initiator move approximately with the same velocity — this can reduce the search for other potential tracks. The acceptance region in the second scan corresponding to another initiator under consideration is set up using the velocity estimate from an established track, together with its uncertainty (possibly augmented to account for target velocity variability).
7. After all the “parallel” tracks have been found, all the remaining unassociated measurements from the first scan are used as initiators for a general search.

Following the initiation procedure, the track continuation procedure, to be described later, is applied. As illustrated in Figure 7.3.1-2, while the continuation algorithm is applied to the initiated tracks, the initiation procedure is applied in a **sliding window** mode on the unassociated measurements.

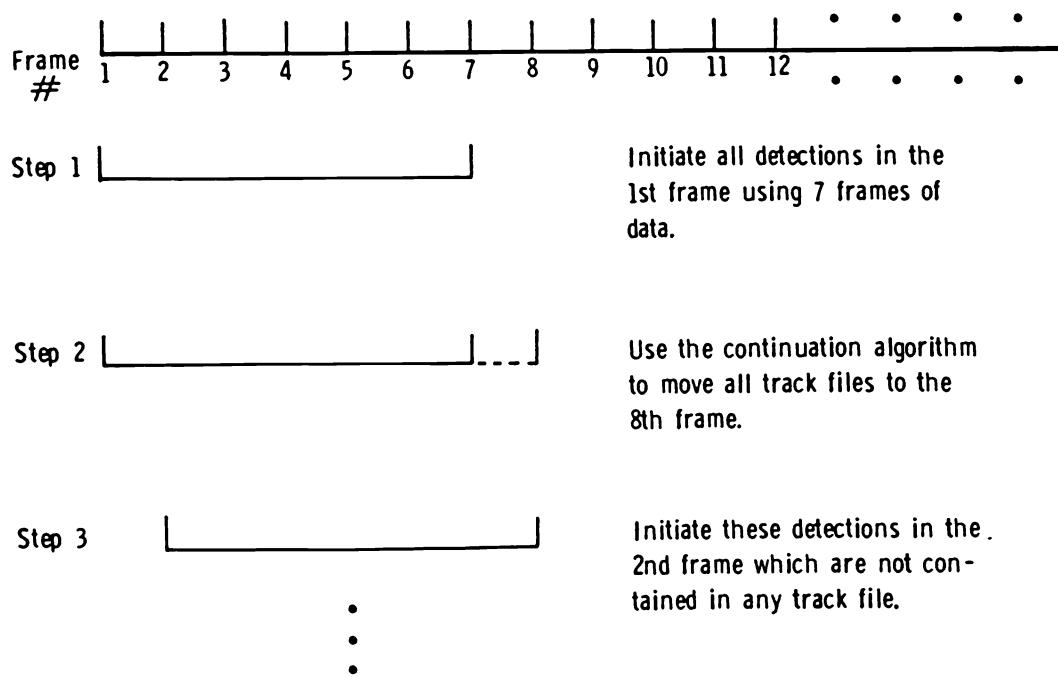


Figure 7.3.1-2: Sliding window initiation logic.

7.3.2 Track Continuation

The track continuation procedure consists of

- (i) prediction
- (ii) measurement association (assignment)
- (iii) updating.

The prediction and updating are done with a Kalman filter, using target equations of motion, i.e., ***precision filtering***.

The measurement association procedure to be described next consists of a relatively simple logic to overcome the ambiguities one can encounter in practical situations.

Figure 7.3.2-1 illustrates some typical situations in the track continuation process:

- In Case 1 the track is split if more than one measurement falls in the validation region.
- If the validation region is empty, the track is extrapolated but it will be dropped if a pre-determined number of consecutive validation regions are also empty, which is illustrated in Cases 1 and 2.
- As shown in Case 3, a track is split but, if for a number of scans the resulting tracks have the same validated measurements, then they are merged.

Ambiguity Resolution

Two procedures for resolving ambiguities in associating measurements validated for two different tracks are discussed next. They are of the “maximum likelihood” type and rely on time windows of

- length 1 for immediate resolution
- length 2 for deferred or delayed resolution,

as illustrated in Figure 7.3.2-2.

7.3.2 Track Continuation

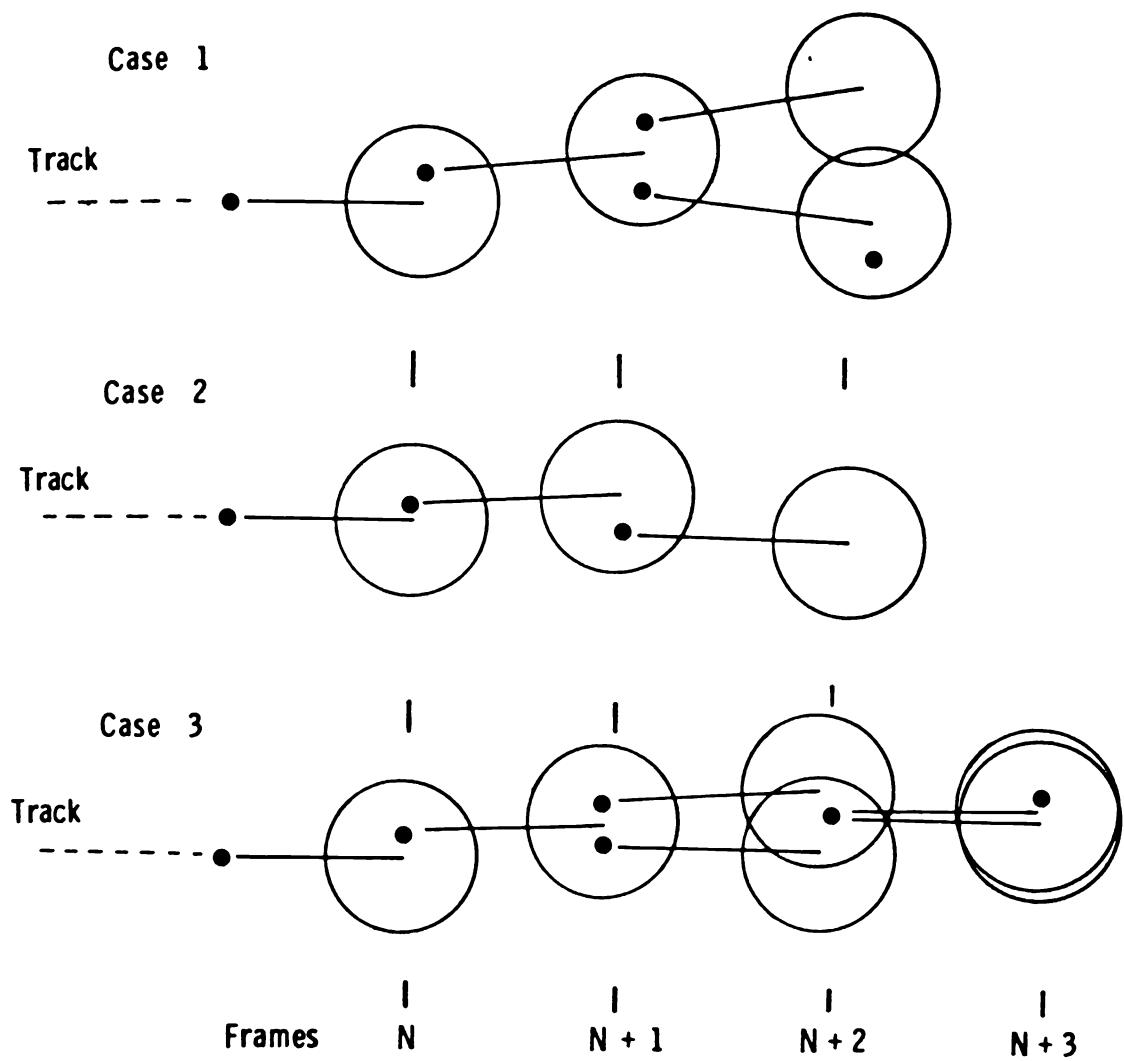
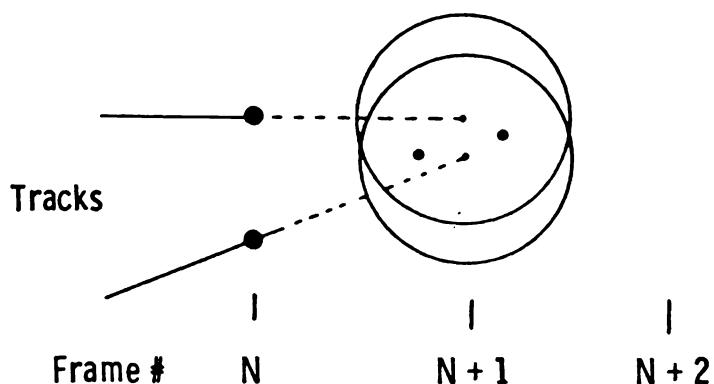


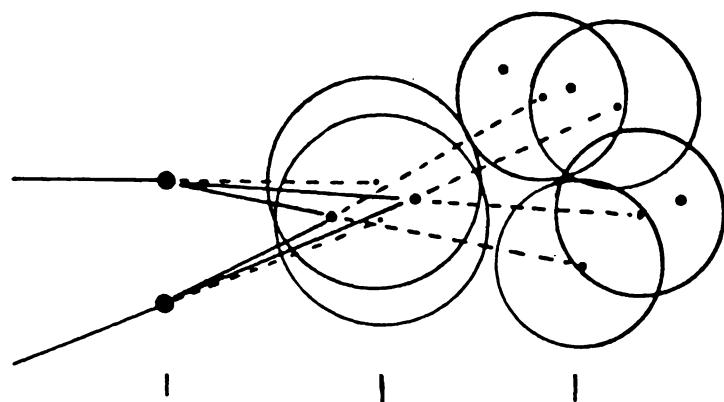
Figure 7.3.2-1: Typical situations encountered in track continuation.

7.3.2 Track Continuation

Immediate Resolution



Delayed Resolution



- + Using Interpolated Estimate to Resolve the Ambiguity of the Previous Frame

Figure 7.3.2-2: Track continuation ambiguity resolution.

7.3.2 Track Continuation

Immediate Ambiguity Resolution

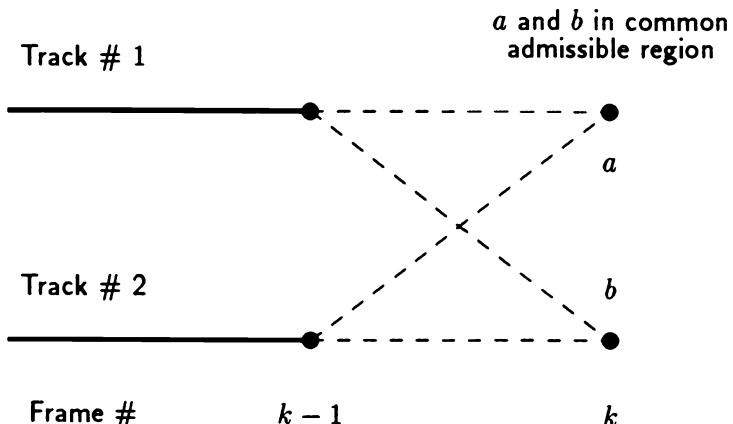
The **immediate resolution** method is illustrated in more detail in Figure 7.3.2-3. Under the assumption of Gaussian pdf for the innovations, the likelihood function of associating measurement a to track 1 and measurement b to track 2 is

$$\Lambda(\theta_{a1}, \theta_{b2}) = ce^{-\frac{1}{2}(D_{1a} + D_{2b})} \quad (7.3.2-1)$$

where

$$D_{1a} = \nu'_{a,1} S_1^{-1} \nu_{a,1} \quad (7.3.2-2)$$

is the (normalized) distance squared from measurement a to track 1. In the above, $\nu_{a,1}$ denotes the innovation corresponding to measurement a if it belongs to track 1 and S_1 is the innovation covariance for track 1.



Pattern match matrix:

	Measurements	
	<i>a</i>	<i>b</i>
Track # 1	D_{1a}	D_{1b}
Track # 2	D_{2a}	D_{2b}

Figure 7.3.2-3: Immediate ambiguity resolution.

The most likely association (based on only one scan — immediate resolution of ambiguity) is the one with the minimum sum of distances as can be seen from (7.3.2-1). Therefore, using the “pattern match matrix” illustrated in Figure 7.3.2-3, one has to choose the *pair of distances* with the smallest sum.

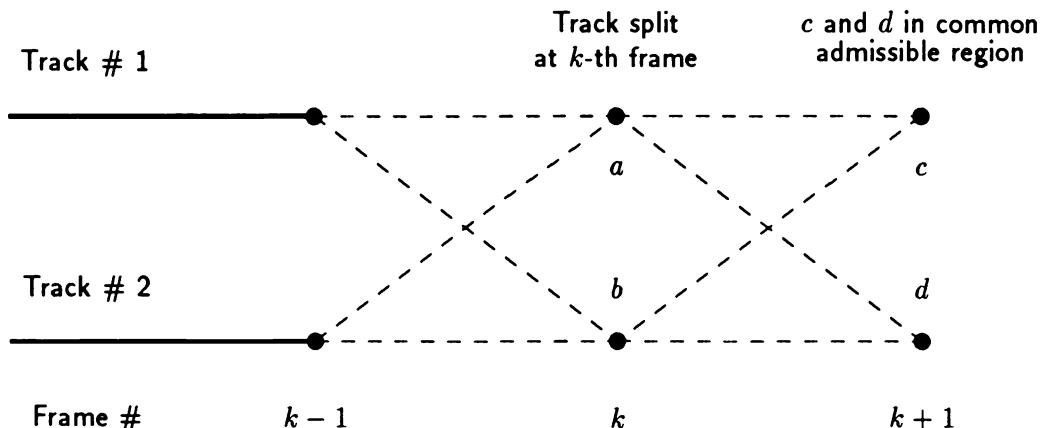
7.3.2 Track Continuation

Deferred Ambiguity Resolution

The one-step **deferred ambiguity resolution** technique is illustrated in Figure 7.3.2-4. There is a “two-layered pattern match matrix” in this case that consists of the distances defined as follows:

$$D_{1ac} = \nu'_{a,1} S_1^{-1} \nu_{a,1} + \nu'_{c,1a} S_{1a}^{-1} \nu_{c,1a} \quad (7.3.2-3)$$

where $\nu_{a,1}$ and S_1 are as before; $\nu_{c,1a}$ is the innovation corresponding to measurement c w.r.t. the track made up of track 1 continued with measurement a and S_{1a} is the corresponding covariance matrix.



Two-layer pattern match matrix:

		For measurement c		For measurement d	
		Measurements		Measurements	
		a	b	a	b
Track # 1	D_{1ac}	D_{1bc}		D_{1ad}	D_{1bd}
Track # 2	D_{2ac}	D_{2bc}		D_{2ad}	D_{2bd}

Figure 7.3.2-4: Deferred ambiguity resolution.

Similarly to (7.3.2-1), the most likely association is the one with total “distance,” defined in (7.3.2-3), being the smallest.

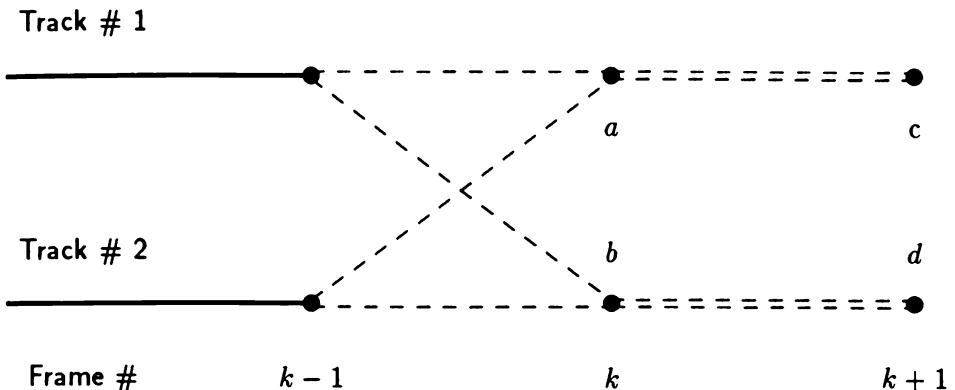
7.3.2 Track Continuation

Simplified Deferred Ambiguity Resolution

Since the number of entries to be considered for deferred ambiguity resolution when there are many track files can become large, a further simplification can be considered. Namely, denote

$$X_{1a} = \min(D_{1ac}, D_{1ad}) \quad (7.3.2-4)$$

Then one has only a “single layered pattern match matrix” as shown in Figure 7.3.2-5.



Pattern match matrix (to resolve measurements of k -th frame using measurements up to $(k+1)$ -th frame):

		Measurements	
		a	b
Track # 1	X_{1a}	X_{1b}	
	X_{2a}	X_{2b}	

X_{ij} : The smallest residual of track i going through measurement j (of the k -th frame) to end at the $(k+1)$ -th frame.

Figure 7.3.2-5: Simplified deferred ambiguity resolution.

The procedure is then to choose the feasible pair with the smallest sum. This procedure is simpler than the full deferred resolution algorithm and only slightly more complicated than the immediate resolution approach; performancewise, however, it is close to the former.

7.3.3 An Exoatmospheric Ballistic Missile Defense Application

7.3.3 An Exoatmospheric Ballistic Missile Defense Application

The Scenario

A passive sensor located at an altitude of 5 km scans a field of 30° width in azimuth and 16° in elevation.

There are 825 targets observed over 16 sampling times.

Figure 7.3.3-1 illustrates the targets as they appear in the measurement space. These angular observations were generated by a functional model of the sensor and the signal processor that accounts for target intensity variations, background and receiver noise, and limited sensor resolution.

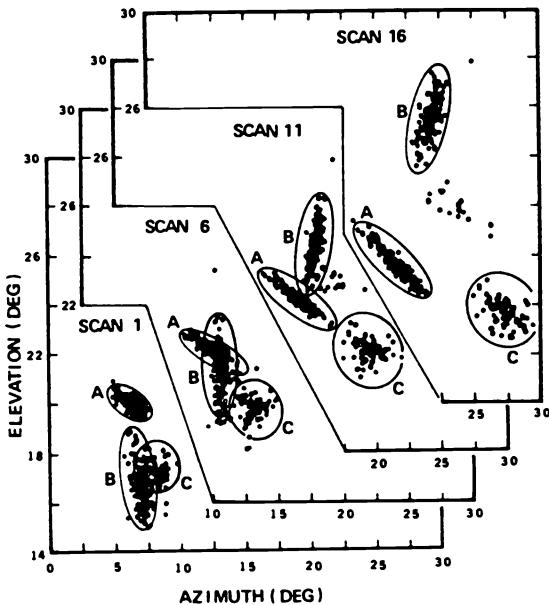


Figure 7.3.3-1: Targets in the measurement space.

Figure 7.3.3-2 shows the number of targets detected (resolved or unresolved from other targets). The number of resolved measurements was about 775, while the total number of measurements was about 800. The difference, about 25, were merged measurements, caused by (mostly) two neighboring targets, which indicates that all targets were eventually detected.

7.3.3 An Exoatmospheric Ballistic Missile Defense Application

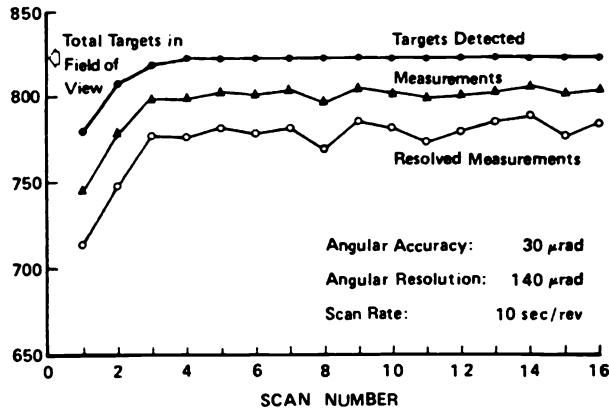


Figure 7.3.3-2: Target detection summary.

System Level Performance

The system performance evaluation was done by evaluating the following performance number of a track file, called **track purity**

$$p = \frac{M_t}{M_{TF}} \quad (7.3.3-1)$$

where

- M_t = largest number of measurements in the track file that originated from the same target,
- M_{TF} = total number of measurements in the track file.

From this, one can define the **fraction of correct associations**

$$p_c(x) = \frac{N_c(x)}{N_T} \quad (7.3.3-2)$$

where

- $N_c(x)$ = the number of track files with purity greater than or equal to x ,
- N_T = total number of targets.

Similarly, the **fraction of false associations** — with purity *below* x — is

$$p_f(x) = 1 - \frac{N_c(x)}{N_T} \quad (7.3.3-3)$$

where

- N_{TF} = total number of track files.

7.3.3 An Exoatmospheric Ballistic Missile Defense Application

Performance vs. Sensor Parameters

Table 7.3.3-1 shows five sets of sensor parameters, defined by the angular measurement accuracy and resolution.

For these five cases the overall tracking performance is illustrated in Figure 7.3.3-3. For fixed sensor accuracy (cases A, B, C) the performance degrades as the sensor's resolution capability degrades. Similarly, for fixed resolution (cases B, D, E) the performance degrades as the sensor accuracy decreases.

Sensor Parameters	Test case				
	A	B	C	D	E
Angular Accuracy (μrad)	15	15	15	30	50
Angular Resolution (μrad)	80	140	200	140	140

Table 7.3.3-1: Test cases.

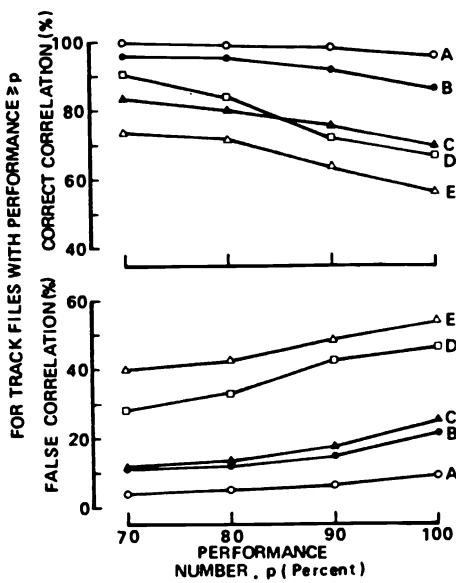


Figure 7.3.3-3: Overall tracking performance.

7.3.4 A Logic Based Initiation — Summary

Assumptions

- There is an *unknown number of targets* to be tracked in the presence of *false alarms*
- Tracks are to be initiated from sequences of associated measurements of a certain length
- Due to resolution limitations a track can be a cluster of neighboring targets that can become resolved as they get closer to the sensor.

The algorithm

- For each “initiator” (first measurement) a validation region is set up for the next time based on prior target motion information
- Tracks are split whenever there is more than one measurement in the validation region
- Prediction is done using polynomial fit (kinematic model)
- Order of polynomial is determined by goodness of fit
- Tracks with empty validation regions over a predetermined number of samples are dropped
- Association ambiguities arising from “common” measurements in the intersection of validation regions are resolved by an approximate maximum likelihood approach (“pattern match method”) using the measurements from
 - the last scan only, or
 - the last two scans — deferred ambiguity resolution
- After (the magical number of) 7 scans a goodness of fit test for the polynomial fit is used to eliminate unlikely sequences — those that remain are used to initialize precision tracking
- The velocity information from an initial association can be used to reduce the validation gate of other initiators if one can assume that the targets move approximately in parallel
- This initiation procedure is used over a sliding window for every measurement not associated with a track — this allows
 - initializing newly detected targets
 - following real targets that “split”
 - picking up “lost” targets

7.3.4 A Logic Based Initiation — Summary

- The track continuation via precision filtering
 - uses dynamic model for target motion with numerical integration for prediction if necessary
 - resolves ambiguities with the same approximate ML technique and updates the state estimate with a standard filter (other options possible).

7.4 EVALUATION OF A COMPOSITE LOGIC FOR TRACK FORMATION: A CASE STUDY

7.4.1 Introduction

A **track formation (initiation)** procedure using a **composite logic**, suitable for a high data rate sensor, is considered. A high data rate sensor (relative to the target motion) is one in which target detections that occur in consecutive looks (scans) are at most a few resolution cells apart.

The goal is to evaluate a class of track formation logics based upon **system requirements**, and to determine the **system target capacity** — the maximum number of targets the system can handle without interference among them.

The system requirements are

- (i) the **target track detection probability**
- (ii) the **false track probability**.

An **M out of N logic**, denoted as M/N , is a test which stipulates that an event (association of a measurement from a “validation gate” to a track) must occur at least M times in N consecutive sampling times.

An n block composite logic is a functional composition of n M_i/N_i logics. The 2 block logic, denoted as $M_1/N_1 \times M_2/N_2$, is the composition $M_2(M_1/N_1)/N_2$.

The purpose of this section is to present a **performance prediction** technique that

- does not require simulations and
- can account in a realistic way for the effect of the clutter

in terms of

- P_D , the **target detection probability per look** (single scan), and
- P_{FA} , the **false alarm probability per look per cell**.

7.4.1 Introduction

For the simplest tracking scenario of a single target in the absence of false alarms, standard Bernoulli sums can be used to get

- (a) P_{DT} — the **target track detection probability** in the absence of false alarms, in terms of P_D .

The other quantities of interest are

- (b) \tilde{P}_{DT} — the **target track detection probability in the presence of false alarms**, a function of P_D and P_{FA} ,
- (c) P_{FT} — the **false track probability** in the absence of targets, a function of P_{FA} ,
- (d) \tilde{P}_{FT} — the **false track probability in the neighborhood of a target**, a function of P_{FA} and P_D .

To obtain the last three quantities above, the so-called **Common Gate-History (CGH) algorithm** is used. The CGH algorithm greatly reduces computational and storage requirements, and avoids the need for simulations, which would be costly due to low-probability events.

A plot of \tilde{P}_{DT} vs. \tilde{P}_{FT} , called a **System Operating Characteristic (SOC)**, is made, where values of P_D and P_{FA} are varied to conform to the **Receiver Operating Characteristic (ROC)** corresponding to the system's **Signal to Noise Ratio (SNR)**. This allows choice of the **detector operating point** such as to satisfy the overall system requirements.

The CGH algorithm is then used to infer the overall system capacity in terms of the maximum number of targets that can be tracked with a certain reliability.

7.4.2 The Association Gates

Detections in the sensor's field-of-view, assumed here to be a radar, are assumed to be resolved into N_R range and N_F Doppler cells. Thus the total number of **resolution cells** is

$$N_c = N_R N_F \quad (7.4.2-1)$$

A target **association (acceptance) gate** consists of a number of resolution cells around its predicted location in the measurement space.

Target Motion Model

The following two assumptions are made:

- (i) The predicted location of the target in the measurement space (the center of the gate) is assumed here to be the *last target detection*
- (ii) The size of the association (acceptance) gate is determined by the target dynamics and depends only on the elapsed time since the last association.

Denote by $g(\omega_l)$ the **association gate size** (in cells) that has grown for ω_l looks (sampling intervals) since the *last association*. This number ω_l , which determines the size of the gate, will be called the **gate index**.

The size of the gate is determined by using the *target's maximum velocity and acceleration*. Thus any range-Doppler cell that the target could be in (given the cell of the latest detection) will be part of the gate. Therefore, if the target is detected, it will be in its gate.

For the case study considered, Table 7.4.2-1 gives the postulated gate growth sequence that reflects the targets' maneuverability.

If on any look (sample time) a detection falls in the gate, then it is associated with the detection that established the gate.

Unassociated detections are those detections that occur outside acceptance gates. These detections are used to initiate new track files.

7.4.2 The Association Gates

Number of looks since last detection (gate index)	Range cells on each side	Total range cells	Doppler cells on each side	Total Doppler cells	Gate size
1	1	3	3	7	21
2	1	3	6	13	39
3	2	5	9	19	95
4	2	5	12	25	125
5	2	5	15	31	155
6	3	7	18	37	259
7	3	7	21	43	301
8	4	9	24	49	411
9	4	9	27	55	495
10	4	9	30	61	549
11	5	11	33	67	737

Table 7.4.2-1: Growth of gates in case study.

7.4.3 Single Target in the Absence of False Alarms

7.4.3 Single Target in the Absence of False Alarms

For an $M_1/N_1 \times M_2/N_2$ composite logic the **target track detection probability** for a single target in the absence of false alarms is

$$P_{DT} = P_{M_2/N_2} = \sum_{j=M_2}^{N_2} \binom{N_2}{j} (P_{M_1/N_1})^j (1 - P_{M_1/N_1})^{N_2-j} \quad (7.4.3-1)$$

i.e., a **Bernoulli sum**, where

$$P_{M_1/N_1} = \sum_{i=M_1}^{N_1} \binom{N_1}{i} P_D^i (1 - P_D)^{N_1-i} \quad (7.4.3-2)$$

Figure 7.4.3-1 gives, for a $2/8 \times 4/4$ logic, a plot of P_{DT} vs. P_D . For example, to obtain $P_{DT} = 0.8$, one needs $P_D \approx 0.46$.

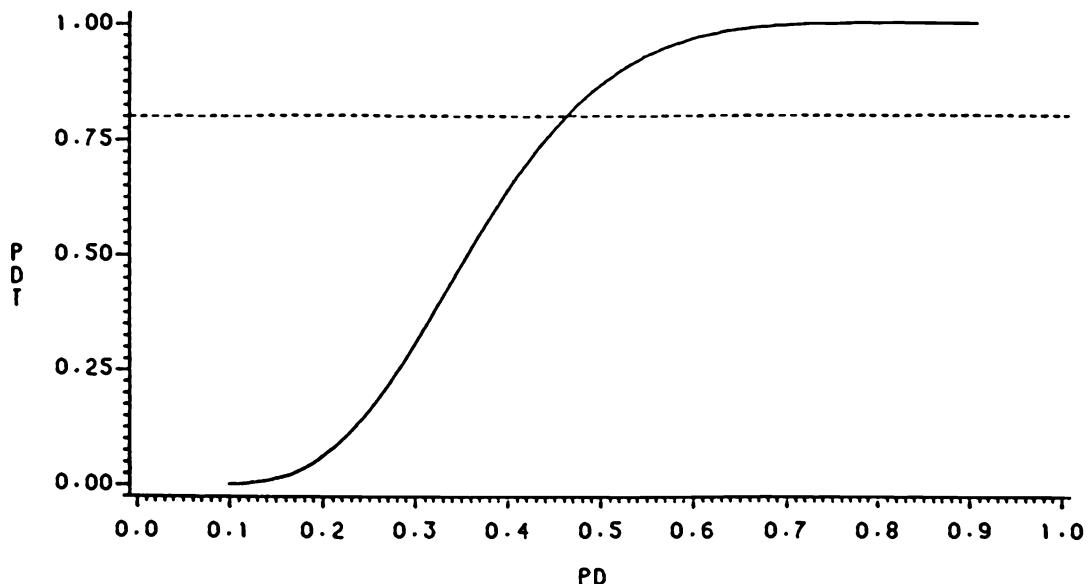


Figure 7.4.3-1: Track detection probability vs. single-look target detection probability for a $2/8 \times 4/4$ composite logic.

Note the “S-shape” of the above curve (sigmoid): medium values on the x -axis are mapped into large values on the y -axis, while small values are mapped into very small ones — this reflects the “discrimination power” of the logic.

7.4.3 Single Target in the Absence of False Alarms

Synchronous vs. Asynchronous Logic

If the counting of the composite logic is synchronized to start at a predetermined initial time and if the target is illuminated by the radar beam for the duration of the counting, then the above equations yield an exact relationship between P_D and P_{DT} .

If, however, the logic is triggered by an unassociated detection — **detection-triggered logic** — then the relationship between P_D and P_{DT} obtained from (7.4.3-2) is only approximate. Nevertheless, the difference is of the order of a few percent only. All the remaining sections are for detection-triggered logics.

Effect of False Alarms

The values obtained from (7.4.3-2) are good in a tracking environment where the target tracks are not affected significantly by false alarms (i.e., low probability of false alarms per look per cell, P_{FA}). In this case one would expect that the P_D required to give a desired P_{DT} , as per (7.4.3-2), would be close to the P_D necessary in the actual tracking environment.

Thus (7.4.3-2) can be used for *preliminary selection of the composite logic* to use in more realistic and more complicated analyses.

7.4.4 Evaluation in a Realistic Environment

7.4.4 Evaluation in a Realistic Environment

One approach to evaluate situations with false returns is to enumerate every possible track sequence that will satisfy the composite logic requirements. For the case of false alarms in the absence of targets, this approach is not too burdensome computationally.

However, when considering the more complicated scenario of a single target in the presence of false returns, the computational load of this (brute force) approach increases tremendously.

Table 7.4.4-1 shows the number of possible tracks for a small sample of composite logics. There is a large factor of increase when going from environment E_1 to environment E_2 .

environment	logic	1/1	2/2	$2/2 \times 1/2$	$2/8 \times 4/4$
E_1 — false alarms in the absence of targets		1	1	3	247^4
E_2 — Single target in the presence of false alarms		2	4	24	6544^4
Increase factor $E_1 \rightarrow E_2$		2	4	8	4.9×10^5

Table 7.4.4-1: Number of possible tracks for some composite logics.

Because of the excessive computation required to evaluate the performance in environment E_2 , an approximate method, which greatly reduces the computational load, is described in the sequel.

7.4.5 The Common Gate-History Algorithm

7.4.5 The Common Gate-History Algorithm

For the scenario of a single target in the presence of false alarms a track can have any one of the following makeups:

- (i) Detections from the target only,
- (ii) Detections from the target and from false alarms,
- (iii) Detections from false alarms only.

Given these possible track matchups, the evaluation of a two stage composite logic of the type $M_1/N_1 \times N_2/N_2$ can be done based on a sufficient statistic of a track, which consists of

- (a) the number ω_l of time steps since the *last detection* (target or false alarm),
- (b) the number ω_{lt} of time steps since the *last target only detection*,
- (c) the count λ of detections within stage 1 of the logic (logic status; for other logics more than one status might be required).

Under assumptions to be introduced later, the **gate-history vector**

$$\omega = [\omega_l \ \omega_{lt} \ \lambda] \quad (7.4.5-1)$$

is the *sufficient statistic* for the $2/8 \times 4/4$ logic. Note that $\lambda \leq 2$ because of the saturation of the $2/8$ logic stage.

This vector evolves according to a Markov chain. The states and transition probabilities can be set up manually (and then the chain evolution follows from standard equations) or, in an *automated manner* shown next.

7.4.5 The Common Gate-History Algorithm

Automatic Generation of the Markov Chain

Using the sufficient statistic ω , the algorithm

- sequentially generates the feasible values of ω ,
- follows their evolution and
- lumps all tracks with a *common* ω .

In view of this, it is called the **Common Gate-History (CGH)** algorithm.

The following events that occur within its gates are considered:

A_1 — No detection

A_2 — Target only detection (TGT)

A_3 — False alarm (FA)

A_4 — Target detection and a false alarm (TGT&FA). Assuming that a track splitting algorithm is used, then this event yields a true track as well as a false track.

The events with more than one false alarm in a gate are assumed to occur with negligible probabilities if P_{FA} is low enough (10^{-3} or less for the gates presented in Table 7.4.2-1; if one has to work with a larger value of P_{FA} , then the algorithm should be augmented to include more events).

This is summarized by the assumption

(AS1): Events A_1, \dots, A_4 are exhaustive.

Outline of the CGH Algorithm

The CGH evaluation algorithm works by continuing each track at each look according to the above four possible events.

The track probability at look (scan) i is multiplied by the probability of the event that continues this track at look $i + 1$ (to give the updated track probability at look $i + 1$) and then lumping (addition of the probabilities of) all the tracks with a common ω is performed.

This leads to a Markov chain with a manageable number of states, which are generated automatically.

7.4.5 The Common Gate-History Algorithm

Illustration of the CGH Algorithm Operation

Table 7.4.5-1 illustrates a detection-triggered logic. The status is shown at look 1 (time t_1), just prior to look 2 (time t_2^-), at t_2 , and prior to look 3 (time t_3^- before, and after lumping).

At look 1 the two events that could start a track file are TGT and FA.

For the one with target detection, $\omega_l(t_2^-) = \omega_{lt}(t_2^-) = 1$ and $\lambda = 1$. For the track started with a false alarm, $\omega_{lt} = F$ where F is a *flag* indicating that there has not been a target detection in this track file. This is later used to evaluate the probability that a track started by a false alarm will “pick up” a target.

The two track files which exist at look 1 are continued at look 2 by the occurrence of the events A_1 through A_4 .

		t_1	t_2^-	t_2	t_3^- before lumping	t_3^- after lumping
TGT	$\omega = [1, 1, 1]$	A_1 (No detection):		$\omega = [2, 2, 1] \rightarrow (1)$	(1): $\omega = [2, 2, 1]$	
		A_2 (TGT):		$\omega = [1, 1, 2] \rightarrow (3)$		
		A_3 (FA):		$\omega = [1, 2, 2] \rightarrow (2)$	(2): $\omega = [1, 2, 2]$	
		A_4 (TGT&FA):		$\omega = [1, 2, 2] \rightarrow (2)$	(3): $\omega = [1, 1, 2]$	
FA	$\omega = [1, F, 1]$	A_1 (No detection):		$\omega = [2, F, 1] \rightarrow (4)$	(4): $\omega = [2, F, 1]$	
		A_2 (TGT):		$\omega = [1, 1, 2] \rightarrow (3)$		
		A_3 (FA):		$\omega = [1, F, 2] \rightarrow (5)$		
		A_4 (TGT&FA):		$\omega = [1, F, 2] \rightarrow (5)$	(5): $\omega = [1, F, 2]$	

Table 7.4.5-1: CGH algorithm — track continuation and lumping by sufficient statistics (common ω) for evaluation of \tilde{P}_{FT} .

After the second look there are 8 different tracks. However, some of these tracks have the same sufficient statistics and, after lumping, there are only 5 track files left. At t_3^+ if the tracks are not lumped there will be 32 different track files but after lumping only 9 track files remain.

7.4.5 The Common Gate-History Algorithm

The CGH Algorithm: the Key to Feasibility

For the $2/8 \times 4/4$ scheme, over the first eight looks there are 6,544 tracks that satisfy the $2/8$ logic. However, with the CGH algorithm, each of these tracks will correspond to one of *only 35 possible* ω — this was determined by the CGH algorithm when automatically generating the chain.

Probabilities of the Gate Events

The gate size (number of range and Doppler cells) for an ω_l -step prediction is denoted as $g(\omega_l)$.

The “no detection” probability in gate $g(\omega_l)$ is

$$P\{A_1\} = \left[1 - \frac{g(\omega_l)}{g(\omega_{lt})} P_D\right] (1 - P_{FA})^{g(\omega_l)} \quad (7.4.5-2)$$

where the ratio of actual gate $g(\omega_l)$ to the size it should have been, $g(\omega_{lt})$, *reduces* P_D to a *lower effective value*.

The probability of the target only detection is

$$P\{A_2\} = \frac{g(\omega_l)}{g(\omega_{lt})} P_D (1 - P_{FA})^{g(\omega_l)} \quad (7.4.5-3)$$

The probability of one false alarm and no target detection in gate $g(\omega_l)$ is (assuming the probability of more than one false alarm to be negligible)

$$P\{A_3\} = \left[1 - (1 - P_{FA})^{g(\omega_l)}\right] \left[1 - \frac{g(\omega_l)}{g(\omega_{lt})} P_D\right] \quad (7.4.5-4)$$

Finally, the probability of one false alarm and target detection in gate $g(\omega_l)$ is

$$P\{A_4\} = \left[1 - (1 - P_{FA})^{g(\omega_l)}\right] \frac{g(\omega_l)}{g(\omega_{lt})} P_D \quad (7.4.5-5)$$

Note that these four probabilities do sum up to unity.

7.4.5 The Common Gate-History Algorithm

Description of the CGH Algorithm

For event A_1 the updating of ω is done by incrementing ω_l and ω_{lt} by 1 (the last detection and last target detection are one more look further back in time) and by not changing λ .

For event A_2 , since there has just been a target detection then $\omega_l = \omega_{lt} = 1$ and λ is incremented by 1.

For event A_3 , since there has just been a false detection, $\omega_l = 1$; however, ω_{lt} gets incremented by 1 because the detection was not from the target and λ is incremented by 1.

For event A_4 there is a simultaneous occurrence of a false alarm and the target detection. Assuming that a track splitting algorithm is used, then event A_4 yields a true track as well as a false track.

In view of this, the following additional assumptions are used

(AS2a): In computing the false track probability \tilde{P}_{FT} , tracks continued with event A_4 (one FA and target detected) are lumped with the tracks that continue with event A_3 (one FA only).

(AS2b): In computing the target track detection probability \tilde{P}_{DT} , tracks continued with event A_4 are lumped with the tracks that continue with event A_2 (target detection only).

Using *one of these assumptions at a time*, the CGH algorithm will yield

- under AS2a realistic false track probability, and
- under AS2b realistic target track detection probability.

For the $2/8 \times 4/4$ scheme, λ is used to check for the satisfaction of the first stage in the logic (2/8) requirement. The 2/8 requirement covers eight looks which are made in a time span denoted by T_d .

A track file (sequence of detections) starts with a new unassociated detection for a detection triggered logic. Upon initialization, $\lambda = 1$.

A track file with no detections within the *current* T_d will have $\lambda = 0$. A track file with one detection within the current T_d will have $\lambda = 1$. A track file with two or more detections within the current T_d will have $\lambda = 2$, at which point the counter is “saturated.” At the end of the current T_d only those track files with $\lambda = 2$ will be allowed to continue to the subsequent T_d . At the start of the next T_d period, λ is reset to zero.

The 4/4 requirement is satisfied by track files which have satisfied four consecutive 2/8 requirements.

This methodology can be extended to other logics with suitable modifications.

7.4.5 The Common Gate-History Algorithm

Summary

The CGH algorithm generates automatically the states of the Markov chain modeling the track formation.

The inputs P_D and P_{FA} are used to calculate the probabilities of track continuation events A_1 through A_4 . A track probability is found by multiplying the probabilities of the events that make up the track.

At each look the tracks are lumped by adding the track probabilities of the track files with a common gate history vector ω . This keeps the number of possible states of the Markov chain within reasonable computational limits.

7.4.6 False Tracks in the Absence of Targets

7.4.6 False Tracks in the Absence of Targets

The CGH algorithm can be used to evaluate the **false track probability in the absence of targets**, P_{FT} , in terms of the probability of a false alarm per look per cell, P_{FA} , by setting $P_D = 0$.

General Analysis of Composite Logic Schemes

After the preliminary relationship between P_D and P_{DT} is obtained for a particular logic, as shown in Subsection 7.4.3, use is made of the ROC curve with P_D determined from the desired value of P_{DT} , to obtain the corresponding P_{FA} .

Consider the ROC curve for the **Swerling I detection model**²

$$\text{SNR} = 10 \log \left(\frac{\log P_{FA}}{\log P_D} - 1 \right) \quad \text{dB} \quad (7.4.6-1)$$

shown in Figure 7.4.6-1 for $\text{SNR} = 9\text{dB}$.

The P_{FA} value obtained from the ROC is used to obtain P_{FT} using the CGH algorithm. If an acceptable value of P_{FT} is found, then the analysis can proceed to the more complicated case of a single target in the presence of false alarms.

This requires caution because a logic which yields an acceptable P_{DT} and P_{FT} for a given SNR *may not exist*. It is because of this that the CGH algorithm is valuable since it is quick and requires no simulations.

For the $2/8 \times 4/4$ logic, Figure 7.4.6-2 gives a plot of P_{FT} vs. P_{FA} obtained from the CGH algorithm with $P_D = 0$, and with the gate growth sequence of Table 7.4.2-1.

For example, from Figure 7.4.6-1, the detection threshold setting corresponding to $P_D = 0.46$ (obtained in Section 7.3) yields $P_{FA} = 9 \cdot 10^{-4}$, which, in turn, gives, from Figure 7.4.6-2, $P_{FT} = 10^{-6}$.

²This is based on the signal amplitude fluctuation model discussed in Section 4.6.

7.4.6 False Tracks in the Absence of Targets

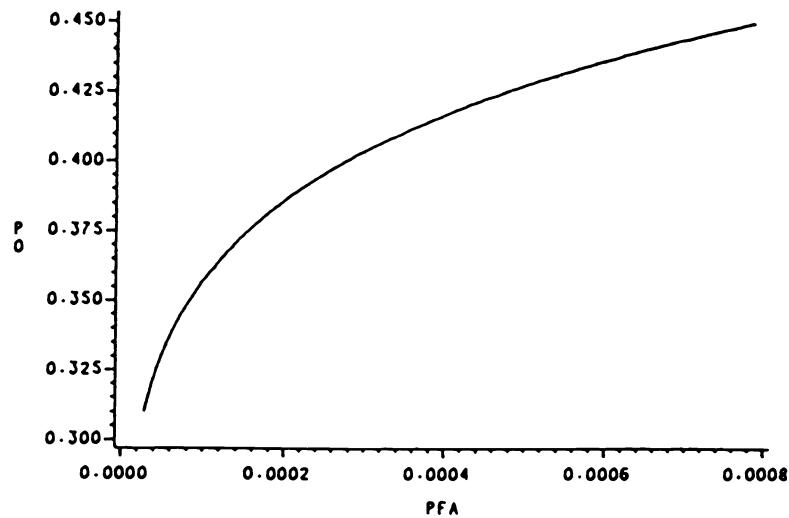


Figure 7.4.6-1: The ROC curve for $\text{SNR} = 9\text{dB}$.

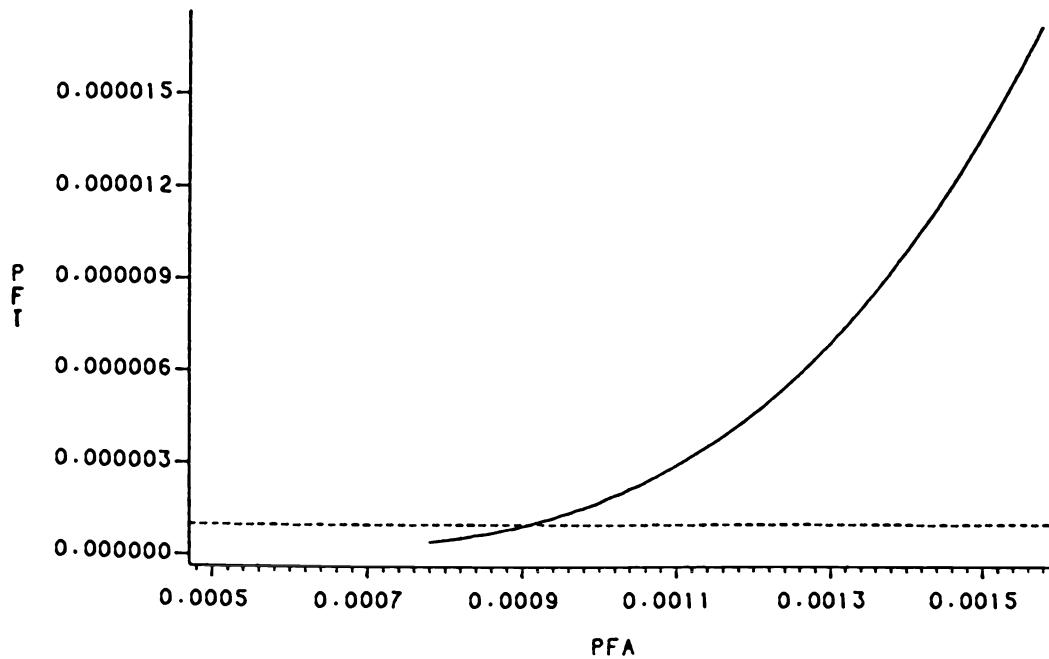


Figure 7.4.6-2: P_{FT} vs. P_{FA} for the $2/8 \times 4/4$ composite logic.

7.4.7 Single Target in the Presence of False Alarms

7.4.7 Single Target in the Presence of False Alarms

With a range of P_D and P_{FA} selected from the ROC as the inputs to the CGH algorithm one can obtain corresponding values of

- \tilde{P}_{DT} , the **target track probability in the presence of false alarms**, and
- \tilde{P}_{FT} , the **false track probability in the neighborhood of a target**.

For the $2/8 \times 4/4$ logic a **target track** is defined as

Any sequence of detections that satisfies the logic and *at least one detection per eight looks is from the target*.

Any other sequence that satisfies the logic will be called a **false track**.

Remarks

The above definition will yield \tilde{P}_{DT} different from P_{DT} obtained using Bernoulli sums because it can be “contaminated” by false alarms. However, since the false alarm rate is low, the effect of false alarm contamination on target tracks should be negligible.

The value of \tilde{P}_{FT} obtained in this evaluation of a single target in the presence of false alarms can differ significantly from P_{FT} obtained in the absence of targets. This is due to the fact that in the neighborhood of target tracks, *false tracks can be “spawned” from a target*: false tracks occur with a higher probability due to target detections.

Caution must be used when interpreting the false track results: P_{FT} obtained from the analysis of Subsection 7.4.6 will be valid only in regions of the tracking environment where there are no targets. The value of \tilde{P}_{FT} , pertains *only to the vicinity of targets*.

Evaluation for a Single Target in the Presence of False Alarms

To obtain \tilde{P}_{DT} , the CGH algorithm

- (a) computes the probability of all detection sequences that satisfy the logic using assumption AS2b ($TGT\&FA \rightarrow TGT$),
- (b) separately stores the probability of target tracks according to the definition adopted.

To obtain \tilde{P}_{FT} , the CGH algorithm

- (a) computes \tilde{P}_{DT} under assumption AS2a ($TGT\&FA \rightarrow FA$),
- (b) subtracts it from the total probability of all detection sequences that satisfy the logic.

7.4.7 Single Target in the Presence of False Alarms

A set of \tilde{P}_{DT} and \tilde{P}_{FT} values are plotted against each other to form the **System Operating Characteristic (SOC)** curve.

The resulting SOC can then be used for deciding

1. which logic scheme will best meet the system requirements for target and false track probabilities,
2. which point on the ROC curve (detection threshold) should be used,
3. whether the available SNR can meet the system requirements.

A plot of the ROC for $\text{SNR} = 10\text{dB}$ and the corresponding SOC for the $2/8 \times 4/4$ logic are given in Figure 7.4.7-1. Figure 7.4.7-2 presents the $\text{SNR} = 12\text{dB}$ case.

The operating points are labeled to indicate corresponding points on the two plots. Notice that, in the SOC, as P_D and P_{FA} increase, both \tilde{P}_{DT} and \tilde{P}_{FT} increase. However, as P_{FA} becomes greater than 10^{-3} , Figure 7.4.7-1 indicates that \tilde{P}_{DT} starts to decrease.

The decrease in \tilde{P}_{DT} is explained as follows. From (7.4.5-3) one can see that the *effective* P_D for event A_2 (target only) depends upon the ratio $g(\omega_l)/g(\omega_{lt})$. An increase in P_{FA} decreases $g(\omega_l)$ because false returns are occurring more often. Although an increase in P_D decreases $g(\omega_{lt})$ as well, however, since the rate of increase of P_{FA} is much larger than the rate of increase of P_D , the ratio $g(\omega_l)/g(\omega_{lt})$ decreases. Thus, the *effective* P_D goes down and the *target track detection probability decreases*.

If one wants \tilde{P}_{DT} to be 0.8 or greater, then, as seen from Figures 7.4.7-1 and 7.4.7-2, a minimum P_D of 0.45 is needed. However, since there is a sharp knee in the SOC, it would be better to have a P_D of 0.5 or greater.

Overall System Load

Note that \tilde{P}_{FT} is the probability of a false track in the vicinity of a true track. The quantity P_{FT} is the probability of a false track starting from a resolution cell without “help” from a neighboring target. For a practical situation one can have, say, $\tilde{P}_{FT} \approx 10^{-2}$ and $P_{FT} \approx 10^{-6}$.

The **expected number of false tracks in the surveillance region** is

$$E(N_{FT}) = P_{FT}N_c + \tilde{P}_{FT}N_T \quad (7.4.7-1)$$

where

N_c — number of resolution cells in the field of view

N_T — number of targets.

The above can be used to obtain the computational load for the environment under consideration.

7.4.7 Single Target in the Presence of False Alarms

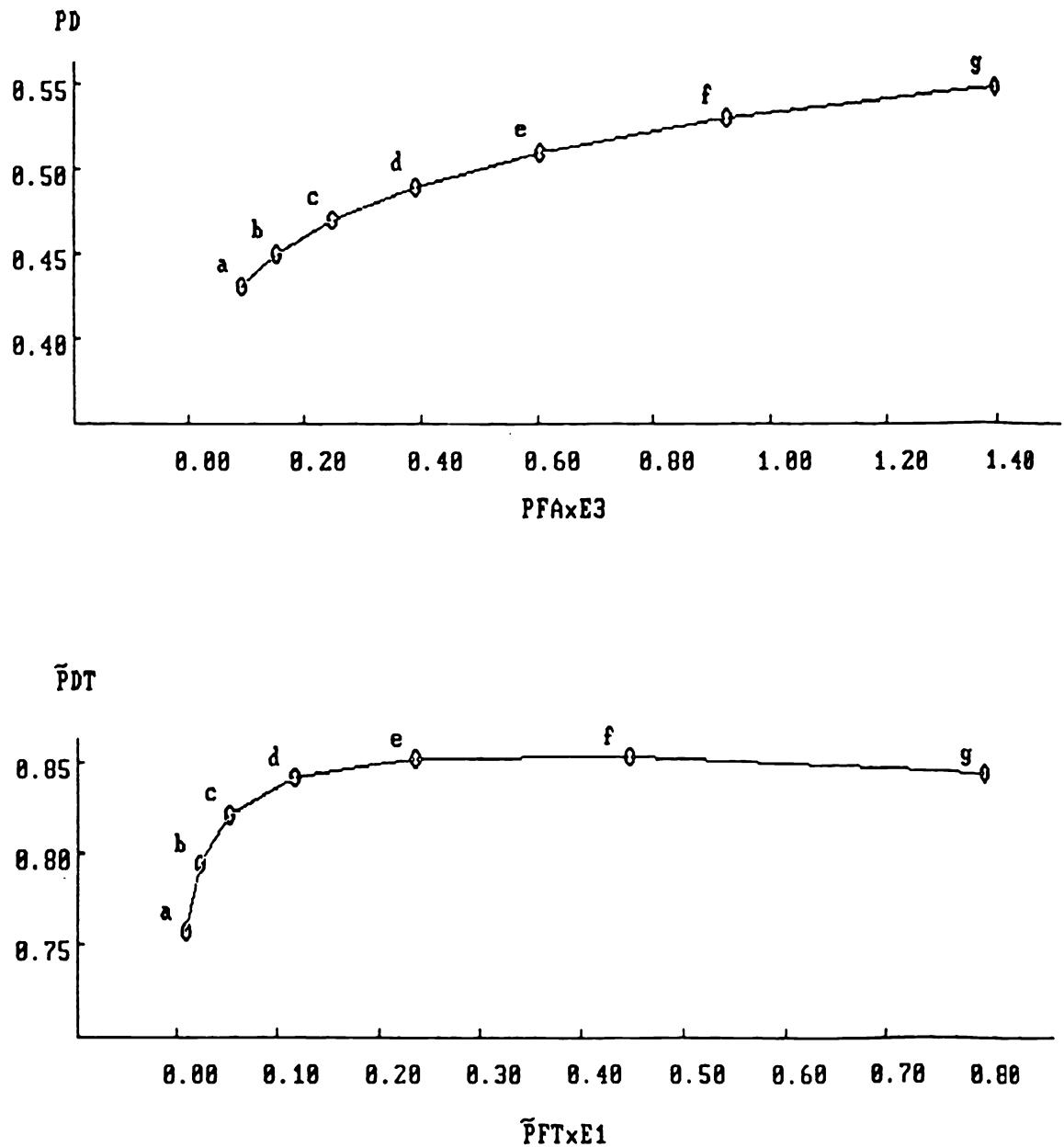


Figure 7.4.7-1: The ROC for SNR = 10dB and the SOC for the $2/8 \times 4/4$ logic.

7.4.7 Single Target in the Presence of False Alarms

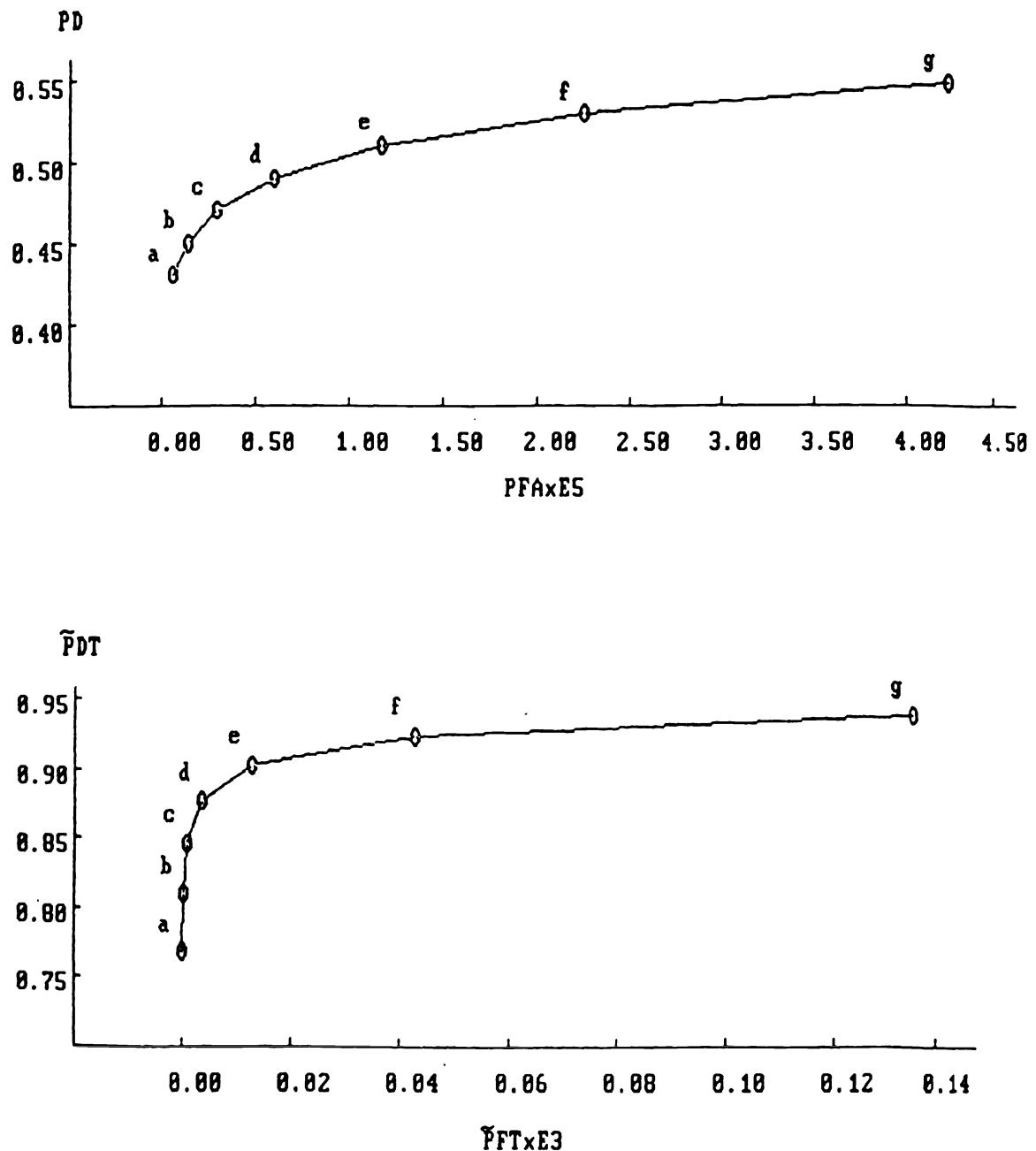


Figure 7.4.7-2: The ROC for SNR = 12dB and the SOC for the $2/8 \times 4/4$ logic.

7.4.8 Multiple Targets and False Alarms

7.4.8 Multiple Targets and False Alarms

The **system capacity** — the number of targets per unit area that can be handled — can be determined by setting a **non-interference condition** for neighboring targets with a certain (high) probability. This then yields a **minimum target separation**.

The approach consists of

- (a) evaluation of the pmf of the **gate index** ω_l (number of looks since the last association, which determines the gate size) for a target track in the presence of false alarms, and
- (b) setting a **non-overlap** condition for the gates of neighboring targets, which guarantees no interference.

The CGH algorithm can yield the pmf of the gate index for a single target with false alarms.

Calculation of the Gate Index pmf

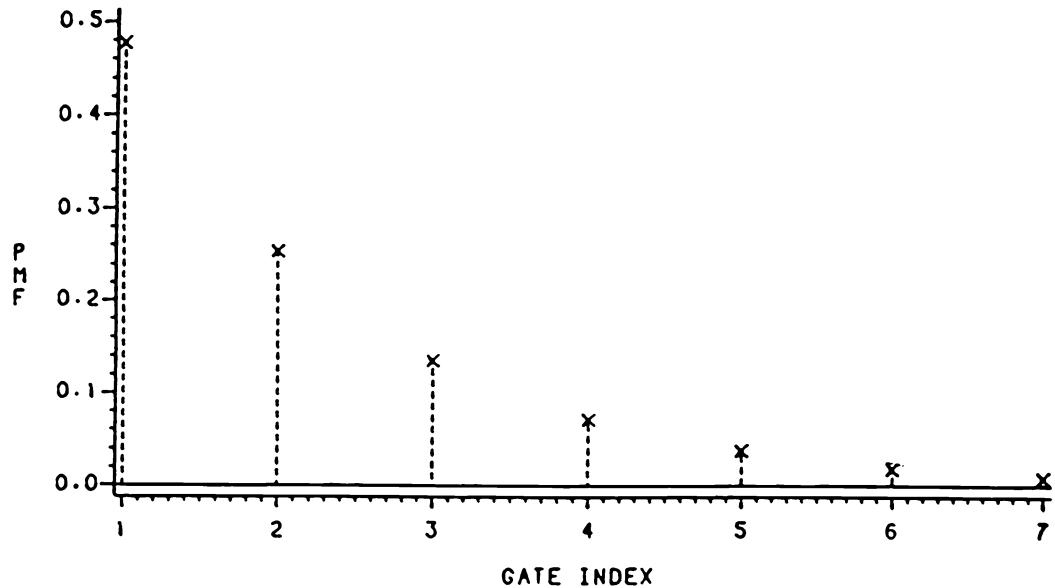
The probability of a gate index being equal to a given number (at a particular look) is found by summing the track probabilities of all tracks (both target and false tracks) with ω_l equal to this number, then dividing this sum by the total probability of all tracks.

Figure 7.4.8-1 gives the pmf of the gate index for the $2/8 \times 4/4$ logic for $P_D = 0.45$ and $P_{FA} = 10^{-3}$ at look 8 and look 32. The pmf at looks 16 and 24 is essentially the same as at look 32.

For these particular looks (8, 16, 24, 32) the largest gate index is seven because tracks with gate index larger than seven would not have passed the $2/8$ requirement, and thus are discarded.

7.4.8 Multiple Targets and False Alarms

LOOK 8.



LOOK 32.

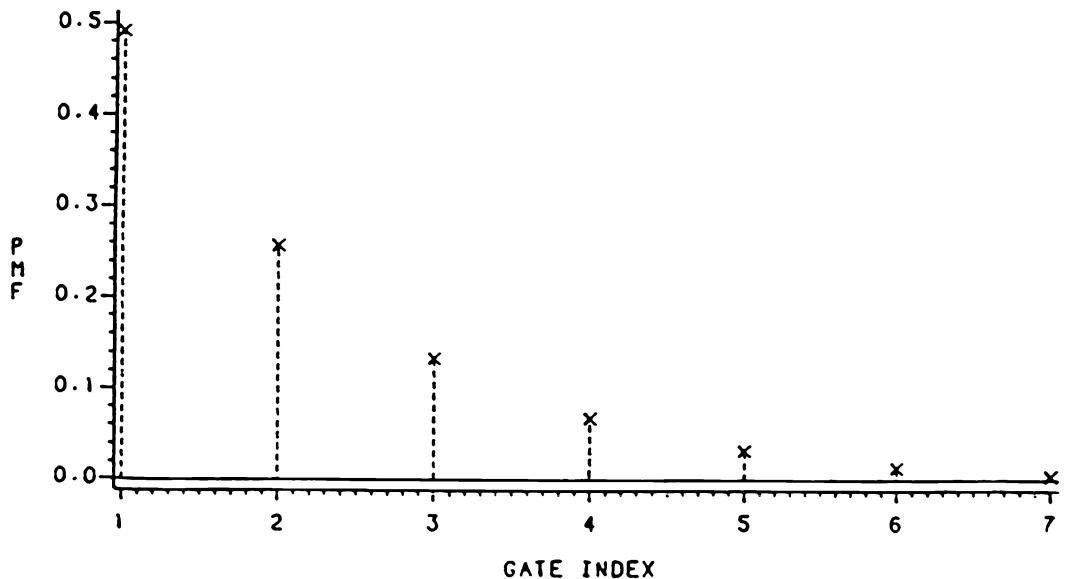


Figure 7.4.8-1: Gate index pmf.

7.4.8 Multiple Targets and False Alarms

Inference on Minimum Target Separation

The probability of overlap in range and Doppler is obtained by noting that there must be overlap in both coordinates for a gate overlap to occur.

The following notations will be used in the sequel:

- $i \triangleq$ gate index (time since last detection)
- $\rho(i) \triangleq$ number of range cells on each side of a gate, detailed in Table 7.4.2-1. For example, with $i_A = 3$, one has $\rho(i_A) = 2$, i.e., target A has a validation gate spread in range of $2\rho + 1 = 5$ cells
- $\delta(i) \triangleq$ number of Doppler cells on each side of a gate. For example, with $i_B = 2$, one has $\delta(i_B) = 6$, i.e., target B has a gate spread in Doppler of $2\delta + 1 = 13$ cells
- $x \vee y \triangleq \max(x, y)$
- $s_r \triangleq$ range separation (number of range cells between the two targets, i.e., if one is in cell j , the other is in cell $j + s_r$)
- $s_d \triangleq$ Doppler separation.

With s_r and s_d given, the probability of two targets overlapping in range and Doppler for one look is:

$$P_{OV} = \sum_{i_A=1}^{i_{\max}} \sum_{i_B=i_0}^{i_{\max}} P(i_A)P(i_B) \quad (7.4.8-1)$$

where $P(i)$ is the probability of gate index equal to i (gate index pmf), and

$$i_0(s_d, s_r, i_A) = i_r(s_r, i_A) \vee i_d(s_d, i_A) \quad (7.4.8-2)$$

$$i_r(s_r, i_A) = \arg \min_i [\rho(i_A) + \rho(i) \geq s_r] \quad (7.4.8-3)$$

$$i_d(s_d, i_A) = \arg \min_i [\delta(i_A) + \delta(i) \geq s_d] \quad (7.4.8-4)$$

In the above i_0 is the minimum gate index for target B such that, for given separation (s_d, s_r) and gate index i_A of target A , the gates overlap; i_{\max} is the largest possible value of the gate index in the logic considered (this is 13, based on detections at looks 1, 2, 15 and 16 in two adjacent groups of 8 looks).

7.4.8 Multiple Targets and False Alarms

Calculation of the Non-Overlap Probability

Once P_{ov} is found then $1 - P_{ov}$ gives the probability of **non-overlap** for one look.

The product of $1 - P_{ov}$ over several looks gives the non-overlap probability of two adjacent targets over these looks.

Table 7.4.8-1 gives the probability of non-overlap over looks 9 through 16. Thus for a Doppler separation of 36 cells or more, regardless of the range separation, there will be a high non-overlap probability between adjacent targets over looks 9–16.

Notice also that for a range cell separation of 6 or more there will be a high non-overlap probability regardless of the Doppler separation. Thus, once there is enough separation in either range or Doppler then a high non-overlap probability is guaranteed.

s_r	0	2	4	6
s_d				
36	0.95	0.95	0.95	0.96
24	0.56	0.56	0.56	0.95
12	0.01	0.01	0.30	0.95
0	0.00	0.00	0.30	0.95

Table 7.4.8-1: Probability of non-overlap over looks 9 through 16.

7.4.9 Composite Logic Evaluation — Summary

The preliminary evaluation of the track initiation based on a composite 2-stage logic in a surveillance system consisted of the evaluation of

- (1) the target track detection probability in the absence of false alarms, in terms of the per look target detection probability P_D ,
- (2) the false track probability in the absence of targets, a function of the per look per cell false alarm rate, P_{FA} .

This was followed by evaluation of the situation with target and false alarms, with the quantities of interest being

- (3) the target track detection probability in the presence of false alarms
- (4) the false track probability in the presence of a target, both being functions of P_D and P_{FA} .

The first quantity, the target track detection probability in the absence of false alarms, could be obtained using standard Bernoulli sums.

To obtain the remaining three quantities, the *Common Gate-History Algorithm (CGH)* was used.

The CGH Algorithm

- Generates automatically the states of the Markov chain modeling the track formation process and follows their evolution
- Greatly reduces computational and storage requirements compared to a brute-force approach
- Avoids the need for simulations, which would be costly due to low-probability events.

Assumptions of the CGH Algorithm

- The validation gate size (number of resolution cells) depends on
 - target maneuverability
 - elapsed time since the last association
- The probability of more than one false alarm in a gate is negligible.

The CGH algorithm has the feature of accounting for the reduced effective target detection probability due to the false alarms.

The *System Operating Characteristic (SOC)* was introduced as the plot of

- the target track detection probability in the presence of false alarms vs.
- the false track probability,

where values of P_D and P_{FA} are varied to conform to the ROC corresponding to the system's SNR.

System Optimization

This technique allows selection of the detector's operating point such as to satisfy the overall system requirements.

System Evaluation

The CGH algorithm can be used to infer the *overall system capacity* in terms of the maximum number of targets that can be tracked with a certain reliability — this was done by setting a “non-overlap” condition between the gates of neighboring targets.

7.5 BIBLIOGRAPHICAL NOTES

The joint likelihood approach to track formation presented in Section 7.2 is based partially on [Mor77], where the unity target detection probability case was considered. The use of the likelihood ratio, which is necessary to handle the practical situation with $P_D < 1$, was introduced in Chapter 7 of [Bar90].

A score function approach based on likelihood function for measurement-to-measurement association was developed in [SB75b] and can be found in [Bla86]. It considers targets with $P_D \leq 1$, new targets and track terminations, i.e., it is more general than [Mor77]; however, it does not lend itself to easy maximization. It has been applied in a sequential manner using the “nearest neighbor approach” to practical problems [Bla86]. Another score function approach, for discrete-valued measurements, was developed in [GMB85]. The problem of multitarget tracking with multiple models was considered in [PE79]. More recent work on this problem, in the context of multisensor tracking, can be found in [PDBW90] (see Section 8.8), together with a new numerical optimization technique to find the most likely set of measurements associated across multiple passive sensors.

The logic-based track initiation technique of Section 7.3 is based on [CDY84]. The “pattern matching” method used in this approach is a particular case of the Munkres general assignment problem [BL71b, BL71a] to minimize a sum of distances (a 2-D assignment). The response of sliding-window M/N initiation logic schemes — probability of initiation of a target and expected number of scans to initialize a target track (in the absence of false alarms) — are presented in [Cas76, MO78, FS85]; the same technique was used to evaluate the false track probability, but under the rather restrictive assumption of fixed size gates. A Bayesian approach consisting of a single block logic called the “Adaptive Track Promotion Logic” was described in [Hol77]; this logic relies on the ratio of the probability of a track being valid to its probability of being invalid given a sequence of detections and performance evaluation at a system level can be done only via simulations.

Section 7.4 is based on [BCL90].

Chapter 8

MULTISENSOR TRACKING AND DATA FUSION

8.1 INTRODUCTION

8.1.1 Outline

When more than one sensor is used to obtain measurements, there are several possible **configurations for information processing** — the sequence in which the data association and tracking are carried out.

The various types of configuration possible are discussed in Section 8.2.

In a **distributed (decentralized) configuration** where each sensor has its information processing system, each such system has a number of tracks. A major question is how to decide whether two tracks from different systems represent the *same target* — the problem of **track to track association**.

If it is decided that two such tracks represent the same target, then one can combine the corresponding estimates — this is called **track fusion**. If done appropriately, track fusion will yield *more accurate estimates* than single sensor based tracks.

Section 8.3 deals with the association and fusion of tracks under the assumption that they have *independent state estimation errors*.

As shown in Section 8.4, tracks obtained from the same target by independent sensors are **dependent**. The methodology for obtaining the corresponding statistical dependence is presented and the implications are illustrated.

8.1.1 Outline

These techniques of fusion can be used at arbitrary times and they are simple, but not optimal.

Multisensor feature estimation using a Bayesian approach, also within a decentralized configuration, is discussed in Section 8.5.

Distributed approaches to multisensor tracking that are optimal for synchronized sensors and nearly optimal for asynchronous sensors are discussed in Section 8.6.

A case study of multisensor air traffic surveillance with a centralized information processing configuration is presented in Section 8.7 using real data from two radars.

A centralized configuration multisensor data association algorithm, suitable for large scale problems and heterogeneous sensors, and which yields a nearly optimal maximum likelihood assignment, is presented in Section 8.8.

The localization of a number of emitters using a large number of direction finders is described in Section 8.9.

Decentralized vs. Distributed System

In a **decentralized system** the various decision makers (or estimators or controllers) do not all share the same information at a given time.

In a **distributed system** the computation/sensing is done at different locations with “local” or “global” information.

These terms are sometimes used interchangeably.

8.1.2 Summary of Objectives

8.1.2 Summary of Objectives

Present the possible configurations for information processing in multisensor systems.

Analyze the association and fusion of tracks accounting for the dependence of their state estimation errors. Quantify the state estimate correlation for the same target at different sensors.

Discuss decentralized multisensor feature estimation and distributed state estimation.

Describe a case study on real data from two radars for air traffic surveillance

Present an assignment algorithm, suitable for centralized configuration multisensor data association, which yields a nearly optimal ML solution

Describe an assignment algorithm for emitter localization from a large number of direction finders.

8.2 MULTISENSOR TRACKING CONFIGURATIONS

8.2.1 Introduction

When more than one sensor is used to obtain measurements, there are several possible **configurations for information processing** — the sequence in which the data association and tracking are carried out.

The following types of configuration are discussed:

1. **Type I configuration:** Single sensor situation, which serves as a baseline (Subsection 8.2.2)
2. **Type II configuration:** Single sensor tracking followed by **track to track association and fusion** (Subsection 8.2.3)
3. **Type III configuration:** Measurement to measurement association across sensors with all the measurements from the same time (the sensors are assumed perfectly synchronized), i.e., **static association**, followed by **central dynamic association and tracking** (Subsection 8.2.4)
4. **Type IV configuration:** Completely **centralized association and tracking** (Subsection 8.2.5)
5. A **hierarchical hybrid configuration** (Subsection 8.2.6).

8.2.2 Single Sensor Tracking

The flowchart of a tracking system using a single sensor, designated as **Type I configuration**, is depicted in Figure 8.2.2-1.

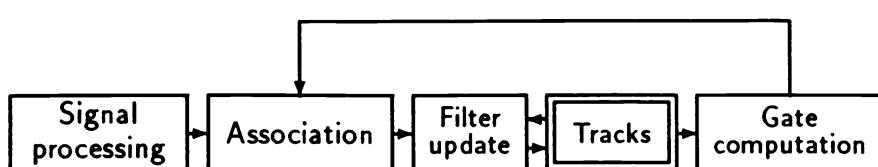


Figure 8.2.2-1: Type I configuration — single sensor tracking.

8.2.3 Single Sensor Tracking Followed by Track Fusion

8.2.3 Single Sensor Tracking Followed by Track Fusion

In this **distributed configuration**, called Type II, each sensor has its own information processor — **local data associator/tracker** — and yields full tracks.

In the **Type IIa configuration**, shown in Figure 8.2.3-1, there is no feedback from the fusion center to the local trackers, while in the **Type IIb configuration**, shown in Figure 8.2.3-2, the local trackers utilize *feedback from the center*.

Type IIa Configuration

Two implementations of this configuration, consisting of

- A statistical test of the tracks whether they belong to the same target — **track to track association** — followed by
- **track fusion**

which can be done at *arbitrary times* — “on demand” — are discussed in Sections 8.3 and 8.4; both are suboptimal, but the latter is nearly optimal.

The implementation of the track fusion presented in Subsection 8.6.1 yields the **optimal global estimate**, but requires *fusion at every sampling time*.

The **fusion of feature estimates** is discussed in Section 8.5.

Type IIb Configuration

In this configuration each local tracker has the global information and this yields the **optimal global estimates if the feedback is available at every sampling time**.

An implementation of this scheme is discussed in Subsection 8.6.2.

8.2.3 Single Sensor Tracking Followed by Track Fusion

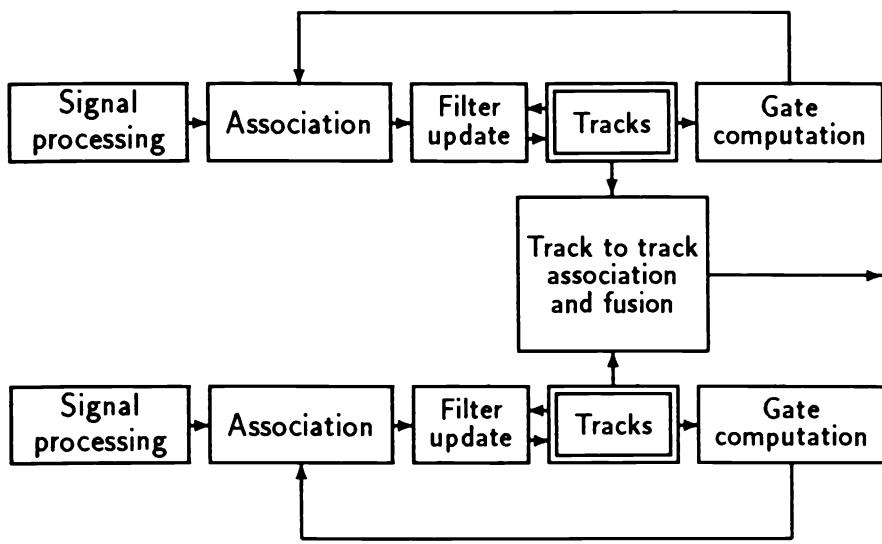


Figure 8.2.3-1: Type IIa configuration — single sensor tracking followed by track fusion without feedback.

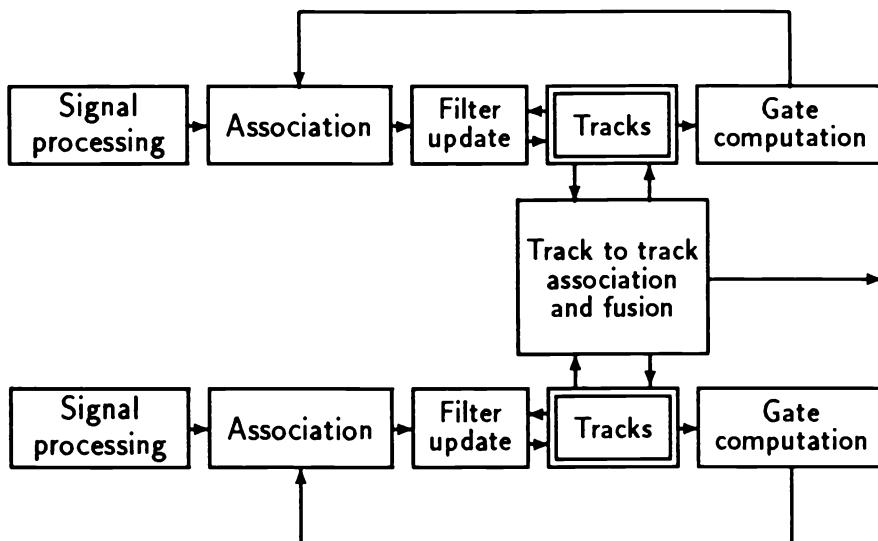


Figure 8.2.3-2: Type IIb configuration — single sensor tracking followed by track fusion with feedback.

8.2.4 Static Intersensor Association Followed by Central Processing

8.2.4 Static Intersensor Association Followed by Central Processing

In this **Type III configuration** there are two stages:

1. The measurements from the various sensors (assumed to be from the same time, i.e., the sensors are assumed **synchronized**) are first associated — this is a **static intersensor association** that yields “supermeasurements”;
2. Then these supermeasurements are processed by a (central) **dynamic association and tracking algorithm**.

Figure 8.2.4-1 shows such a configuration for 3 sensors. As an example, these sensors yield only LOS measurements (i.e., incomplete position observations) that are associated to obtain full position estimates (supermeasurements). Such an algorithm is discussed in Section 8.8.

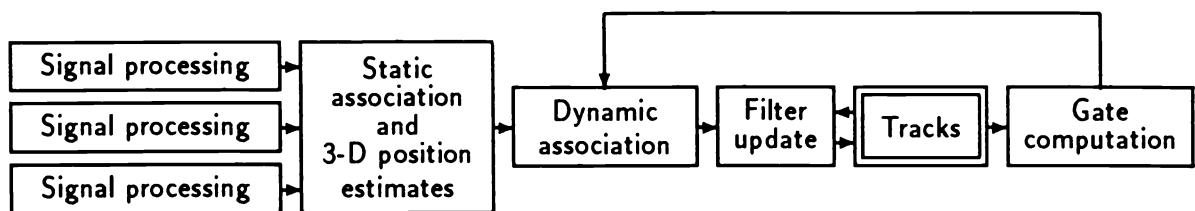


Figure 8.2.4-1: Type III configuration — static followed by dynamic association.

Remarks

The synchronicity assumption for the sensors is critical. Ad-hoc techniques to “move” the measurement times to a common time are possible but questionable.

For fixed sources detected by a **direction finder**, the use of such detections from a window in time is acceptable. The association of detections from a window is discussed in Section 8.9.

8.2.5 Centralized Tracking

8.2.5 Centralized Tracking

The **Type IV configuration**, which is the **centralized tracking**, is depicted in Figure 8.2.5-1.

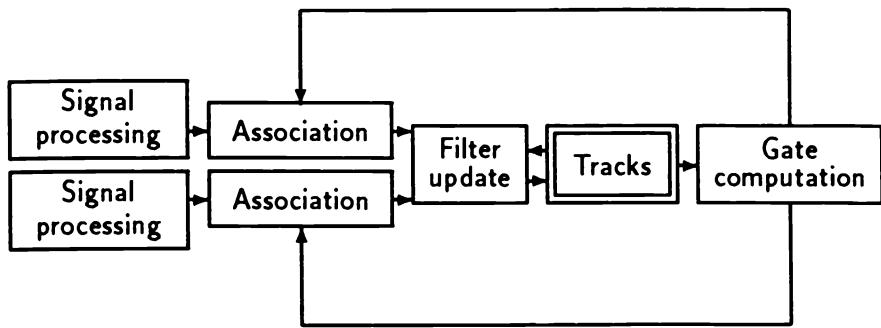


Figure 8.2.5-1: Type IV configuration — centralized multisensor tracking.

Such an configuration was used in Section 4.5 with a sequential implementation of the association/update across the sensors.

8.2.6 A Hierarchical Hybrid Configuration for Multisensor-Multisite Tracking

8.2.6 A Hierarchical Hybrid Configuration for Multisensor-Multisite Tracking

The following 3-level hierarchical situation is considered here:

- There are several sensors of diverse nature at a site (platform)
- There are several sites, each estimating the targets in its area
- The estimation results of each site are to be fused at a central fusion location.

Figure 8.2.6-1 depicts the structure of such a system with a **hybrid configuration**.

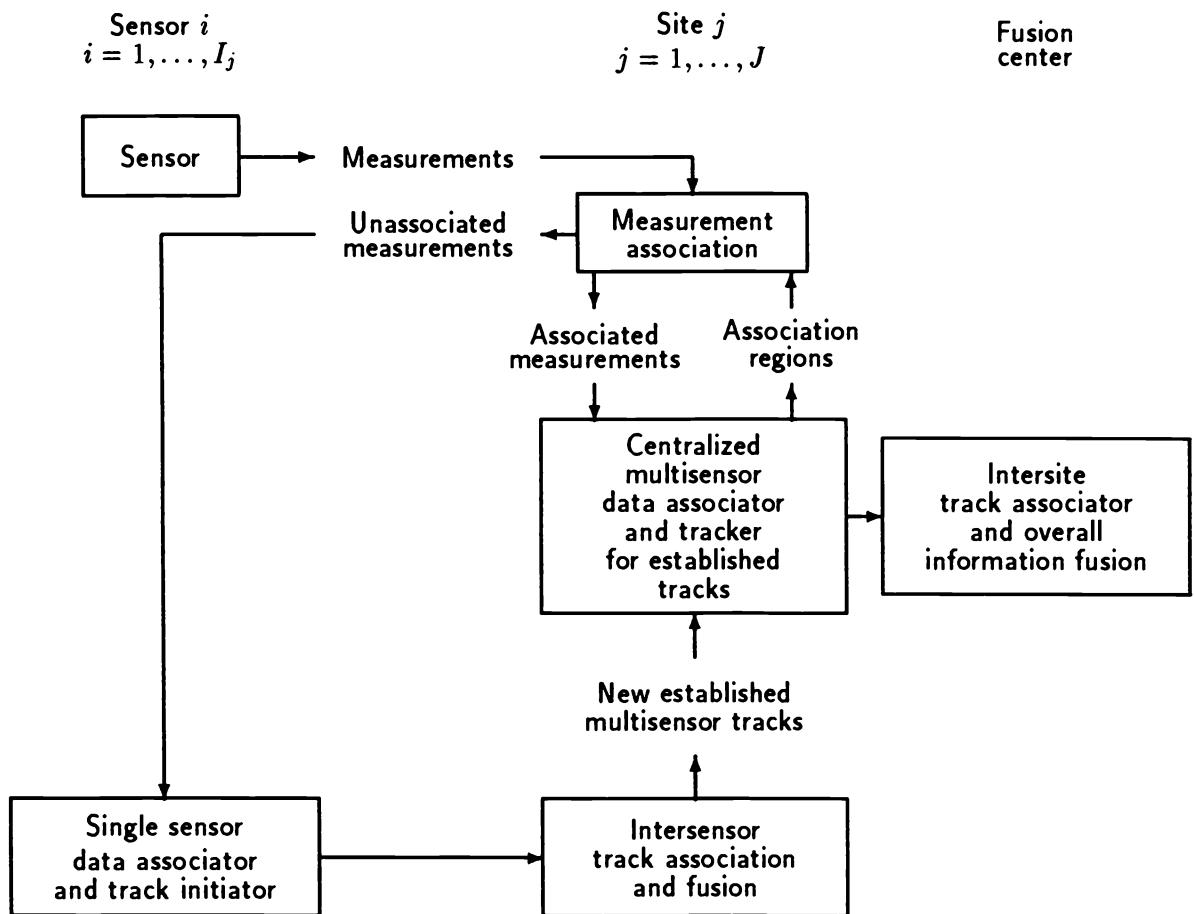


Figure 8.2.6-1: A multisensor-multisite tracking system for multiple targets.

8.2.7 Multisensor Configurations — Summary

8.2.7 Multisensor Configurations — Summary

Several types of configuration of multisensor tracking have been discussed:

- Type I: Single sensor situation, which serves as a baseline
- Type II: Single sensor tracking followed by track to track association and fusion
 - Type IIa: without feedback of the fused tracks to the local tracker
 - ★ suboptimal, but not by much
 - ★ economical, since it can be carried out “on demand”
 - Type IIb: with feedback of the fused tracks to the local tracker
 - ★ optimal
 - ★ more expensive — has to be done at every sampling time
- Type III: Measurement to measurement association across sensors (static association) followed by central dynamic association and tracking
- Type IV: Completely centralized association and tracking
- A hierarchical hybrid configuration.

The criticality of sensor synchronicity for the Type III configuration has been pointed out.

8.3 TESTING AND FUSION OF INDEPENDENT TRACKS

8.3.1 Introduction

This section presents an implementation of the **track to track association and fusion** from the **Type IIa configuration**.

It is desired first to test the hypothesis that two tracks pertain to the same target. The optimal test requires using the entire data base (the sequences of measurements that form the tracks) through the present time k and is not practical. In view of this, the test to be presented is based only on the most recent estimates from the tracks.

Let $\hat{x}^i(k)$ be the estimated state of a target by sensor i with its own information processor. Assume that one has an estimate $\hat{x}^j(k)$ from sensor j , corresponding to the same time.

Both can be current estimates or one can be a prediction as long as they pertain to the same time (the second time argument has been omitted for simplicity).

The corresponding covariances are denoted as $P^m(k)$, $m = i, j$.

Assumption

The state estimation errors

$$\tilde{x}^i(k) = x^i(k) - \hat{x}^i(k) \quad (8.3.1-1)$$

$$\tilde{x}^j(k) = x^j(k) - \hat{x}^j(k) \quad (8.3.1-2)$$

where x^i and x^j are the corresponding true states, are *independent*. This is the **error independence assumption**.

Remark

As shown in Section 8.4, these errors are *dependent*.

8.3.2 The Test for Track Association

8.3.2 The Test for Track Association

Denote the difference of the two estimates as

$$\hat{\Delta}^{ij}(k) = \hat{x}^i(k) - \hat{x}^j(k) \quad (8.3.2-1)$$

This is the estimate of the difference of the true states

$$\Delta^{ij}(k) = x^i(k) - x^j(k) \quad (8.3.2-2)$$

The **same target hypothesis** is that the true states are equal,

$$H_0 : \quad \Delta^{ij}(k) = 0 \quad (8.3.2-3)$$

while the **different target alternative** is

$$H_1 : \quad \Delta^{ij}(k) \neq 0 \quad (8.3.2-4)$$

The error in the difference between the state estimates

$$\tilde{\Delta}^{ij}(k) = \Delta^{ij}(k) - \hat{\Delta}^{ij}(k) \quad (8.3.2-5)$$

is zero-mean and has covariance

$$T^{ij}(k) \triangleq E\{\tilde{\Delta}^{ij}(k)\tilde{\Delta}^{ij}(k)'\} = E\{[\tilde{x}^i(k) - \tilde{x}^j(k)][\tilde{x}^i(k) - \tilde{x}^j(k)]'\} \quad (8.3.2-6)$$

given, under the **error independence assumption**, by

$$\boxed{T^{ij}(k) = P^i(k) + P^j(k)} \quad (8.3.2-7)$$

Assuming the estimation errors to be Gaussian, the test of H_0 vs. H_1 — the **track to track association** test — is

$$\boxed{\text{Accept } H_0 \text{ if } D \triangleq \hat{\Delta}^{ij}(k)'[T^{ij}(k)]^{-1}\hat{\Delta}^{ij}(k) \leq D_\alpha} \quad (8.3.2-8)$$

The threshold D_α is such that

$$P\{D > D_\alpha | H_0\} = \alpha \quad (8.3.2-9)$$

where, e.g., $\alpha = 0.01$. From the Gaussian assumption, the threshold is

$$D_\alpha = \chi_{n_z}^2(1 - \alpha) \quad (8.3.2-10)$$

8.3.3 The Fusion of the Estimates

8.3.3 The Fusion of the Estimates

If H_0 is accepted, then one can carry out the **fusion** (combination) of the two estimates $\hat{x}^i(k)$ and $\hat{x}^j(k)$ of the common true state

$$x^i(k) = x^j(k) \triangleq x(k) \quad (8.3.3-1)$$

The static linear estimation equation that yields the posterior mean \hat{x} in terms of the prior mean \bar{x} and the measurement z is

$$\hat{x} = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z}) \quad (8.3.3-2)$$

Denoting the information from sensor i as the “prior data” D^i one has

$$p(x|D^i) = \mathcal{N}(x; \hat{x}^i, P^i) \quad (8.3.3-3)$$

where the time argument has been omitted for simplicity.

Then a “measurement”

$$\hat{x}^j = x - \tilde{x}^j \quad (8.3.3-4)$$

is made, which represents the data D^j . The error \tilde{x}^j is zero-mean, with covariance P^j and independent of \hat{x}^i .

The equivalent terms in (8.3.3-2) for the present problem are

$$\hat{x} \rightarrow E[x|D^i, D^j] \quad (8.3.3-5)$$

$$\bar{x} \rightarrow \hat{x}^i = E[x|D^i] \quad (8.3.3-6)$$

$$z \rightarrow \hat{x}^j \quad (8.3.3-7)$$

$$\bar{z} \rightarrow E[\hat{x}^j|D^i] = \hat{x}^i \quad (8.3.3-8)$$

The covariance terms in (8.3.3-2) are

$$P_{xz} \rightarrow E\{[x - E(x|D^i)][\hat{x}^j - E(\hat{x}^j|D^i)]'\} = E\{\tilde{x}^i[\hat{x}^j - \hat{x}^i]'\} = E\{\tilde{x}^i[\tilde{x}^i - \tilde{x}^j]'\} = P^i \quad (8.3.3-9)$$

$$\begin{aligned} P_{zz} &\rightarrow E\{[\hat{x}^j - E(\hat{x}^j|D^i)][\hat{x}^j - E(\hat{x}^j|D^i)]'\} = E\{[\hat{x}^j - \hat{x}^i][\hat{x}^j - \hat{x}^i]'\} \\ &= E\{[\tilde{x}^i - \tilde{x}^j][\tilde{x}^i - \tilde{x}^j]'\} = P^i + P^j \end{aligned} \quad (8.3.3-10)$$

Using these, (8.3.3-2) becomes

$$\hat{x} = \hat{x}^i + P^i(P^i + P^j)^{-1}(\hat{x}^j - \hat{x}^i) \quad (8.3.3-11)$$

8.3.4 A Fusion Example

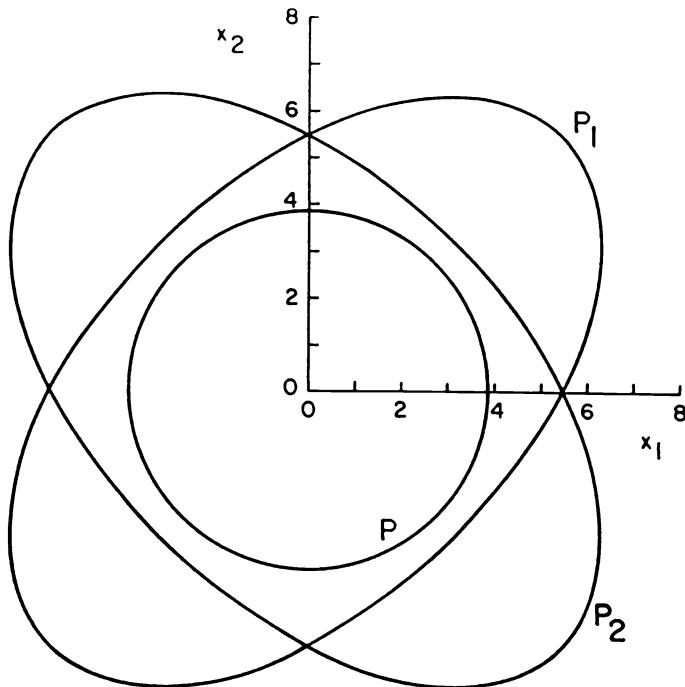


Figure 8.3.4-1: Fusion of estimates with independent errors.

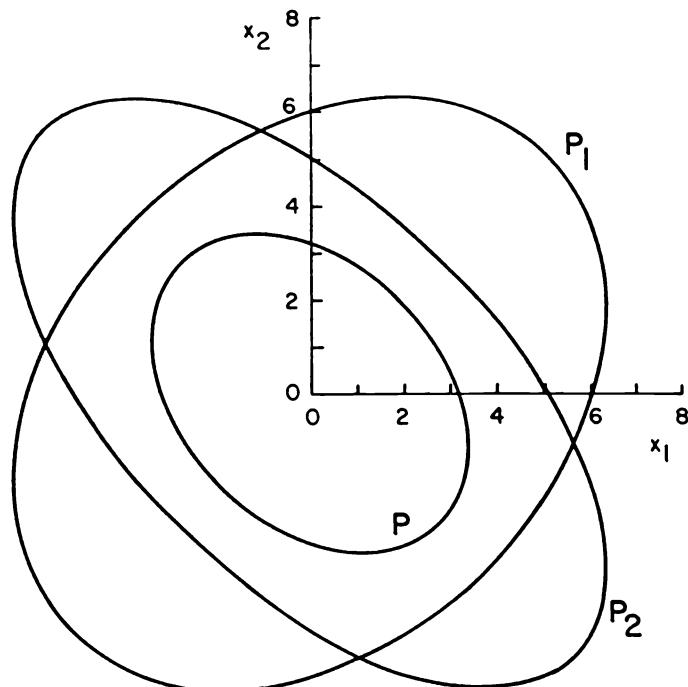


Figure 8.3.4-2: Fusion of estimates with independent errors.

8.3.4 A Fusion Example

8.3.4 A Fusion Example

Consider two estimates, each a 2-dimensional vector, with the following covariance matrices

$$P_1 = \begin{bmatrix} 10 & 5 \\ 5 & 10 \end{bmatrix} \quad (8.3.4-1)$$

$$P_2 = \begin{bmatrix} 10 & -5 \\ -5 & 10 \end{bmatrix} \quad (8.3.4-2)$$

Figure 8.3.4-1 presents the “ 1σ ” **probability region** ellipses corresponding to these matrices, namely

$$x' P_i^{-1} x = 1 \quad i = 1, 2 \quad (8.3.4-3)$$

and the “ 1σ ” ellipse corresponding to the fused estimate whose covariance is

$$P = P_1(P_1 + P_2)^{-1}P_2 \quad (8.3.4-4)$$

The **ellipse semiaxes** are the square roots of the eigenvalues of the corresponding covariance matrices.

Note the **reduction of the uncertainty in fusion**: the ellipse corresponding to the fused estimate is strictly contained in the intersection of the two ellipses prior to fusion.

Figure 8.3.4-2 shows the ellipses of uncertainty corresponding to the covariance matrices

$$P_1 = \begin{bmatrix} 10 & 3 \\ 3 & 10 \end{bmatrix} \quad (8.3.4-5)$$

$$P_2 = \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix} \quad (8.3.4-6)$$

and the resulting fused estimate according to (8.3.4-4).

8.3.4 A Fusion Example

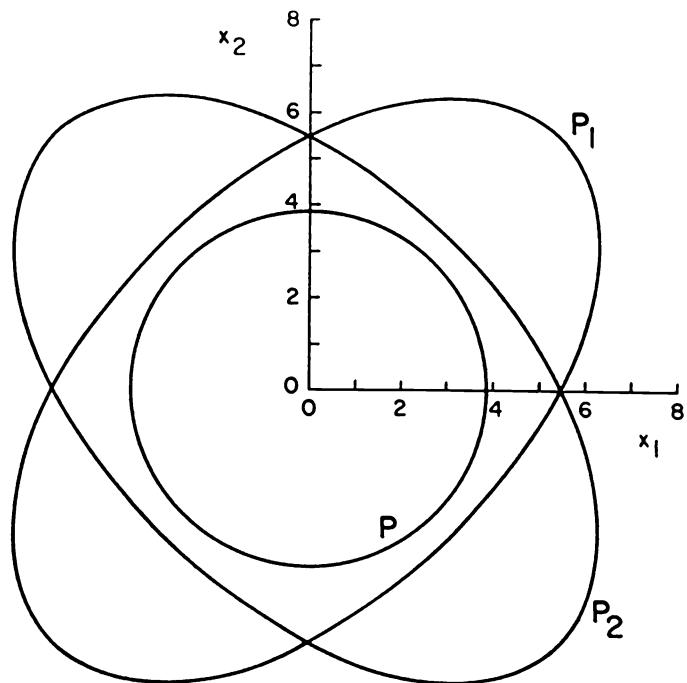


Figure 8.3.4-1: Fusion of estimates with independent errors.

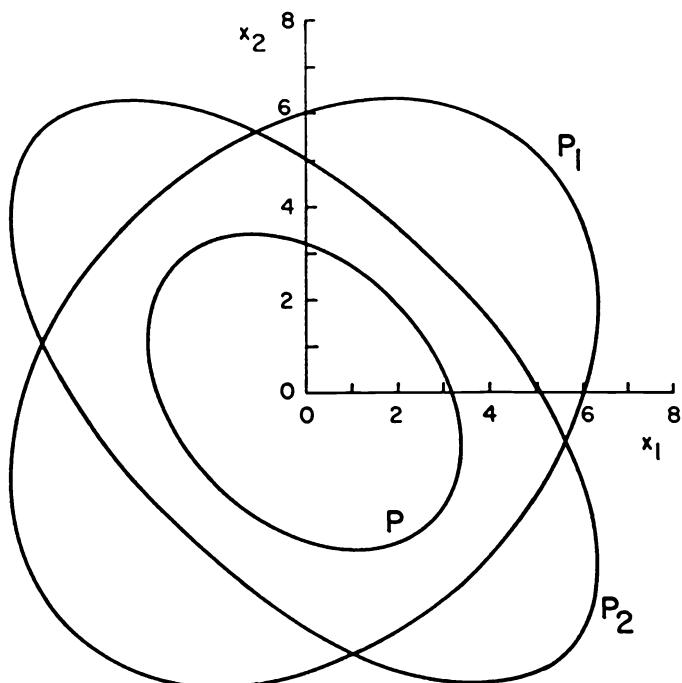


Figure 8.3.4-2: Fusion of estimates with independent errors.

8.3.5 Fusion Under the Independence Assumption — Summary

8.3.5 Fusion Under the Independence Assumption — Summary

Given

- two state estimates pertaining to the same time and
- the associated covariances

from two different sites, each consisting of a sensor and its signal and information processor, it is desired to test whether the corresponding true states can be accepted as being the same, i.e., they describe the same object.

The test can be done using standard hypothesis testing techniques assuming Gaussian distribution for the estimation errors.

If the estimation errors are assumed to be *independent*, the estimates and covariances provided by the two information processors fully specify the required statistical test.

If the test indicates that one can accept the hypothesis that the two objects are the same then one can *combine (fuse) the estimates*.

The accuracy of the resulting *fused information* is higher than that of the individual estimates.

A simple equation yields the fused estimate and associated covariance.

8.4 TESTING AND FUSION OF DEPENDENT TRACKS

8.4.1 Introduction

This section presents another implementation of the **track to track association and fusion** from the **Type IIa configuration**. In the previous section the association testing was done under the assumption that the estimation errors in these tracks are independent. However, as shown in the sequel, the estimation errors are *dependent*.

The initial state estimates and covariances for the two “track files,” for the same target, are $\hat{x}^m(0|0)$ and $P^m(0|0)$, $m = i, j$.

The dynamics of the target are

$$x(k+1) = F(k)x(k) + v(k) \quad (8.4.1-1)$$

where the process noise is zero-mean, white and with covariance $Q(k)$.

The measurements obtained from the two sensors (assumed synchronized) are

$$z^m(k) = H^m(k)x(k) + w^m(k) \quad m = i, j \quad (8.4.1-2)$$

with the measurement noise sequences zero-mean, white, with covariance $R^m(k)$ and *mutually independent*.

The *dependence* between the estimation errors $\tilde{x}^i(k|k)$ and $\tilde{x}^j(k|k)$ from the two track files arises from the *common process noise*.

The fact that the two measurement noise sequences are independent is not sufficient for independence of the estimation errors in the two track files.

The **dependence of two tracks** is quantified by the **cross-covariance** of the corresponding estimation errors and evaluated next for **synchronized sensors**.

The same methodology can be used when this synchronicity assumption does not hold; however, since the calculations become unwieldy in this case, an approximate technique is suggested later.

8.4.2 The Cross-Covariance of the Estimation Errors

8.4.2 The Cross-Covariance of the Estimation Errors

The state estimate using the measurements from sensor m is

$$\hat{x}^m(k|k) = F(k-1)\hat{x}^m(k-1|k-1) + W^m(k)[z^m(k) - H^m(k)F(k-1)\hat{x}^m(k-1|k-1)] \quad (8.4.2-1)$$

where W^m is the filter gain in information processor, $m = i, j$.

Then the corresponding estimation error is

$$\begin{aligned} \tilde{x}^m(k|k) &= x(k) - \hat{x}^m(k|k) \\ &= F(k-1)x(k-1) + v(k-1) - F(k-1)\hat{x}^m(k-1|k-1) \\ &\quad - W^m(k)\{H^m(k)[F(k-1)x(k-1) + v(k-1)] \\ &\quad + w^m(k) - H^m(k)F(k-1)\hat{x}^m(k-1|k-1)\} \\ &= [I - W^m(k)H^m(k)]F(k-1)\tilde{x}^m(k-1|k-1) \\ &\quad + [I - W^m(k)H^m(k)]v(k-1) - W^m(k)w^m(k) \end{aligned} \quad (8.4.2-2)$$

Multiplying the above with $m = i$ by its transpose with $m = j$ yields the **cross-covariance recursion**

$$\begin{aligned} P^{ij}(k|k) &\triangleq E[\tilde{x}^i(k|k)\tilde{x}^j(k|k)'] \\ &= [I - W^i(k)H^i(k)][F(k-1)P^{ij}(k-1|k-1)F(k-1)' \\ &\quad + Q(k-1)][I - W^j(k)H^j(k)]' \end{aligned} \quad (8.4.2-3)$$

This is a *linear recursion* and its initial condition is, assuming the initial errors to be uncorrelated,

$$P^{ij}(0|0) = 0 \quad (8.4.2-4)$$

This is a reasonable assumption in view of the fact that the initial estimates are usually based on the initial measurements, which were assumed to have independent errors.

8.4.3 The Testing for Track Association

8.4.3 The Testing for Track Association

The testing of the hypothesis that the two tracks under consideration originated from the same target is done in the same manner as before except for the following modification to account for the *dependence*.

The covariance associated with the difference of the estimates

$$\hat{\Delta}^{ij}(k) = \hat{x}^i(k|k) - \hat{x}^j(k|k) \quad (8.4.3-1)$$

is

$$T^{ij}(k) \triangleq E\{\tilde{\Delta}^{ij}(k)\tilde{\Delta}^{ij}(k)'\} = E\{[\tilde{x}^i(k|k) - \tilde{x}^j(k|k)][\tilde{x}^i(k|k) - \tilde{x}^j(k|k)]'\} \quad (8.4.3-2)$$

and, with the known cross-covariance P^{ij} , is given by the expression

$$T^{ij}(k) = P^i(k|k) + P^j(k|k) - P^{ij}(k|k) - P^{ji}(k|k) \quad (8.4.3-3)$$

Note the difference between the above and (8.3.2-7).

Effect of the Dependence

The effect of the dependence between the estimation errors is to *reduce* the covariance of the difference (8.4.3-1) of the estimates. This is due to the fact that the cross-covariance term reflects a positive correlation between the estimation errors.

The Test

The hypothesis testing for the **track to track association with the dependence accounted for** is done in the same manner as before in (8.3.2-8), except that the “smaller” covariance from (8.4.3-3) is used in the test statistic, which is, as before, the normalized distance squared between the estimates,

$$D = \hat{\Delta}^{ij}(k)'[T^{ij}(k)]^{-1}\hat{\Delta}^{ij}(k) \quad (8.4.3-4)$$

8.4.4 The Fusion of the Estimates

8.4.4 The Fusion of the Estimates

The fusion is done, as before, according to the linear equation

$$\hat{x} = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z}) \quad (8.4.4-1)$$

where the estimate \hat{x}^i plays the role of the prior mean \bar{x} and the other estimate \hat{x}^j is the “measurement” z .

The covariance between the variable to be estimated and the measurement is

$$P_{xz} \rightarrow E\{\tilde{x}^i(k|k)[\tilde{x}^i(k|k) - \tilde{x}^j(k|k)]'|D^i\} = P^i - P^{ij} \quad (8.4.4-2)$$

The covariance of the measurement is

$$P_{zz} \rightarrow E\{[\tilde{x}^i(k|k) - \tilde{x}^j(k|k)][\tilde{x}^i(k|k) - \tilde{x}^j(k|k)]'\} = P^i(k|k) + P^j(k|k) - P^{ij}(k|k) - P^{ji}(k|k) \quad (8.4.4-3)$$

The ***fusion equation with cross-covariance*** is

$$\boxed{\hat{x} = \hat{x}^i + [P^i - P^{ij}][P^i + P^j - P^{ij} - P^{ji}]^{-1}[\hat{x}^j - \hat{x}^i]} \quad (8.4.4-4)$$

The covariance associated with this fused estimate is

$$\boxed{P = P^i - [P^i - P^{ij}][P^i + P^j - P^{ij} - P^{ji}]^{-1}[P^i - P^{ji}]} \quad (8.4.4-5)$$

Remark

Eq. (8.4.4-4) carries out the fusion of the estimates optimally in the ***minimum mean square error (MMSE)*** or ***maximum likelihood (ML)*** sense, as long as the errors are Gaussian. Nevertheless, the result is *not the optimal global estimate* because the global conditional mean depends on more than the local conditional means, i.e.,

$$E[x|Z^i, Z^j] = \hat{x}(Z^i, Z^j) \neq f[\hat{x}(Z^i), \hat{x}(Z^j)] \quad (8.4.4-6)$$

where Z^m denotes the data at sensor m . For more details, see problem 8-1.

8.4.5 Evaluation of the Effect of the Cross-Covariance on the Intersensor Fusion

Two sensors tracking the same target, each with its local processor, are considered.

The target is modeled by the kinematic model

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} v(k) \quad (8.4.5-1)$$

This is a first order polynomial model (constant velocity in the absence of process noise) with unity sampling time with “piecewise constant” process noise with variable variance q entering into the system.

The measurements at the two sensors are

$$z^m(k) = [1 \ 0] x(k) + w^m(k) \quad m = 1, 2 \quad (8.4.5-2)$$

with the two measurement noises mutually independent and with unity variance.

The maneuvering index for this target, introduced in (6.5.3-14) of [BL93], is thus

$$\lambda = \sqrt{q} \quad (8.4.5-3)$$

Figure 8.4.5-1 presents in the first column the steady-state estimation error ellipses (corresponding to 85% probability mass under Gaussian assumption) in the state space x_2-x_1 (velocity-position) for each sensor. The values of q considered are 0.01, 0.1, 0.5, 1, 2 and 5.

The second column presents the error ellipses one obtains by combining these estimates assuming they are independent. These ellipses are exactly half (in area) compared to the single-sensor error ellipses.

The third column presents the exact ellipses corresponding to the combined estimates obtained by taking into account the dependence between the two tracks due to the common process noise.

8.4.5 Evaluation of the Effect of the Cross-Covariance on the Intersensor Fusion

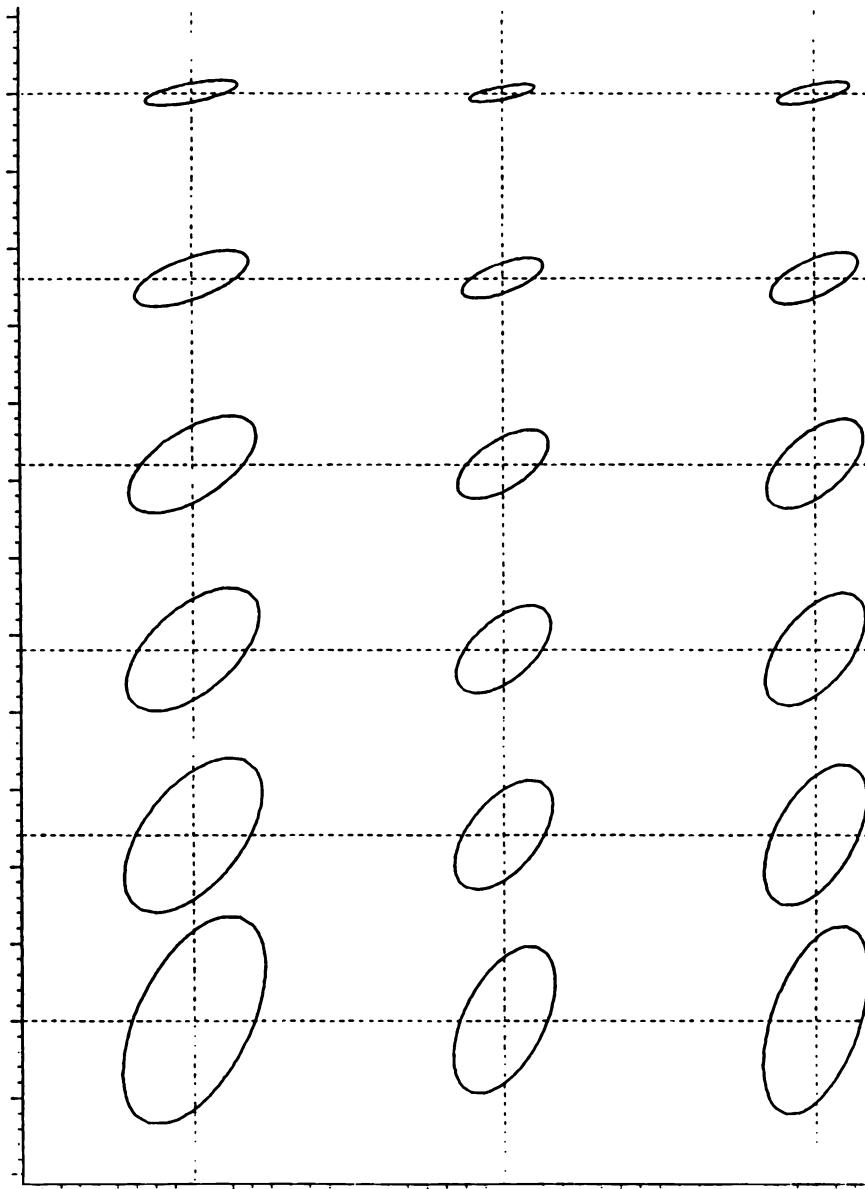


Figure 8.4.5-1: Error ellipses for various levels of maneuvering index (Column 1: for each sensor alone; Column 2: for two sensors combined assuming independence; Column 3: for two sensors combined correctly).

8.4.5 Evaluation of the Effect of the Cross-Covariance on the Intersensor Fusion

Figure 8.4.5-2 shows the ratio of the areas of the ellipses of uncertainty of

- the fused estimates from the two sensors *accounting for their dependence* and
- the single sensor estimates

for various values of the target maneuvering index.

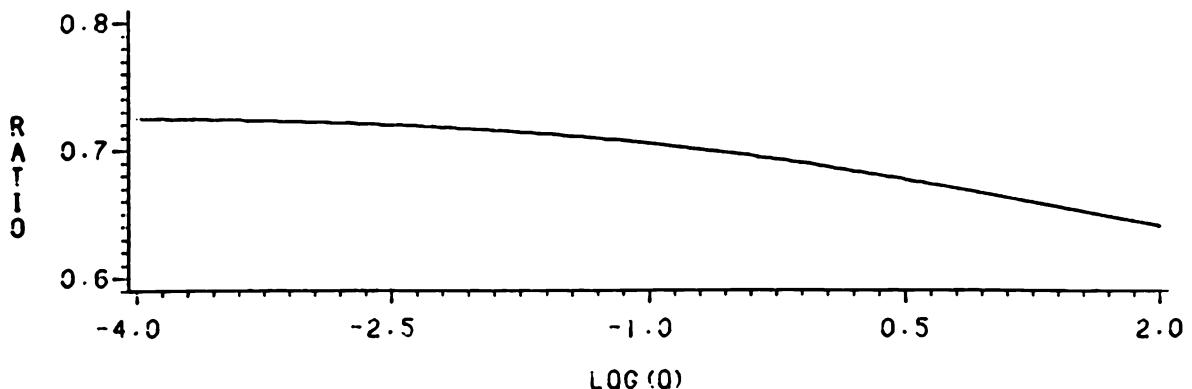


Figure 8.4.5-2: Decrease in the error ellipse area in view of the dependence of the errors in the fusion process.

This ratio would be equal to $1/2$ in the absence of the dependence.

Remark

Numerical results from [RM88] indicate that the **optimal global estimate** has an uncertainty ellipse area only up to 7% below that of the area corresponding to the fused estimate with the cross-covariance. This simple fusion procedure, if done correctly, is quite close to the optimum.

Figure 8.4.5-3 shows the decreases in the elements of the covariance matrix for the same situation — as before, in the absence of dependence, each element would be halved after fusion.

8.4.5 Evaluation of the Effect of the Cross-Covariance on the Intersensor Fusion

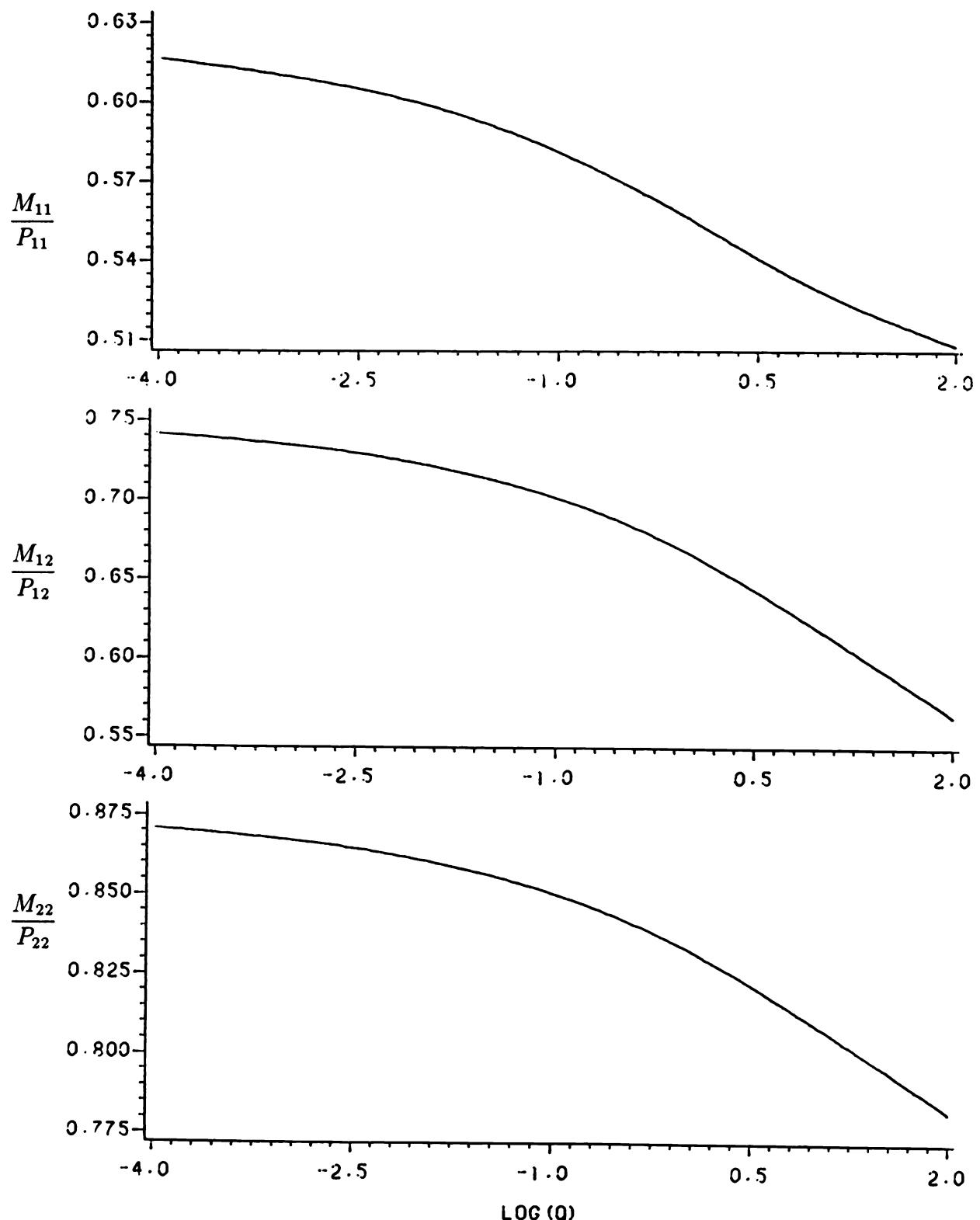


Figure 8.4.5-3: Decrease of the state covariance matrix elements after fusion in view of the dependence of the errors.

8.4.6 An Approximation of the Cross-Covariance

8.4.6 An Approximation of the Cross-Covariance

The results from the previous subsection indicate an almost constant ratio of the error ellipse areas around 0.7. Since the calculation of the cross-covariance in real-time applications is not practical, the following approximation can be considered.

Assume, as in the example, $P^i = P^j$, and, for simplicity,

$$P^{ij} = \rho P^i = \rho P^j \quad (8.4.6-1)$$

Using (8.4.4-5), the fused covariance is

$$P = \frac{1 + \rho}{2} P^i \quad (8.4.6-2)$$

The volume of the corresponding ellipsoid is

$$V^{ij} = c|P|^{1/2} = \left(\frac{1 + \rho}{2}\right)^{\frac{n_x}{2}} V^i \quad (8.4.6-3)$$

where n_x is the dimension of the vector x .

For $n_x = 2$ one has

$$V^{ij} = \frac{1 + \rho}{2} V^i \quad (8.4.6-4)$$

which suggests, based on the above ratio of 0.7, that $\rho \approx 0.4$.

Similar approximations can be found for higher dimension cases.

For the case of unequal covariances, (8.4.6-1) can be replaced by a similar relationship for each element of the covariance matrix:

$$P_{lm}^{ij} = \rho (P_{lm}^i P_{lm}^j)^{\frac{1}{2}} \quad l, m = 1, \dots, n_x \quad (8.4.6-5)$$

The value of ρ can be found in this situation by numerical evaluation of the ellipsoid volume, which does not have an explicit expression as in (8.4.6-3).

8.4.7 Fusion Accounting for Dependence — Summary

In reality the estimation errors for the same target from two sites are *not independent*.

While the measurement noises in different sensors can be assumed independent, the *common process noise* entering into the state equation in the two tracking filters leads to correlated errors.

The resulting *cross-covariance* can be calculated.

This term leads to a modified version of the statistical test that checks whether the two targets are the same.

If the two estimates are accepted as belonging to the same target, the fusion of the estimates is done with a *modified equation that accounts for the dependence of the errors*.

The covariance of the resulting fused estimate is larger than in the case of independent errors but still smaller than that of the individual site estimates.

The fusion of the estimates can be done asynchronously (“on demand” at arbitrary times), however, the calculation of the cross-covariance has to be done synchronously. An approximation that avoids the need to calculate the cross-covariance in real time has been presented.

The optimal fusion of the estimates is not equivalent to the fusion of the entire data, i.e., it is only suboptimal. The optimal fusion of the entire data yields only slightly better results at a much larger cost.

8.5 MULTISENSOR FUSION OF FEATURE ESTIMATES

8.5.1 The Single Sensor Feature Estimation

Consider a target which has a **feature** or **classification** that is time-invariant and can take values in a discrete set

$$\phi \in \{\phi_i : i = 1, \dots, n\} \quad (8.5.1-1)$$

i.e., the feature can be one of a finite number of features.

A Bayesian approach will be taken for the feature estimation.

Assumptions

A prior pmf over this set is assumed to be available

$$p_i(0) = P\{\phi = \phi_i\} \quad i = 1, \dots, n \quad (8.5.1-2)$$

The measurement is characterized by the conditional pdf

$$p(z|\phi = \phi_i) \triangleq g_i(z) \quad (8.5.1-3)$$

This is the **likelihood function of the feature** ϕ_i .

The **feature estimation** consists of evaluation of its pmf over the set given in (8.5.1-1), conditioned on the measurements $z(k)$, $k = 1, \dots$, related to the target's feature according to (8.5.1-3).

The conditional probability of the target having feature i is, using Bayes' formula, given by

$$p_i(1) \triangleq P\{\phi = \phi_i | z(1)\} = \frac{1}{c_1} g_i[z(1)] p_i(0) \quad (8.5.1-4)$$

where c_1 is the normalizing constant

$$c_1 \triangleq \sum_{j=1}^n g_j[z(1)] p_j(0) \quad (8.5.1-5)$$

For subsequent measurements, the recursive updating of the feature probability is

$$p_i(k) = \frac{1}{c_k} g_i[z(k)] p_i(k-1)$$

(8.5.1-6)

where the normalizing constant is

$$c_k \triangleq \sum_{j=1}^n g_j[z(k)] p_j(k-1) \quad (8.5.1-7)$$

8.5.1 The Single Sensor Feature Estimation

The condition for (8.5.1-6) to hold is that

$$p[z(1), \dots, z(k)|\phi] = \prod_{j=1}^k p[z(j)|\phi] \quad (8.5.1-8)$$

i.e., *conditioned on the feature, the measurements are independent.*

Example

Let the feature be a frequency with two possible values $\phi_1 = 10$, $\phi_2 = 12$, with the prior (8.5.1-2) uniform, i.e., $p_1(0) = p_2(0) = 0.5$.

Assume there is a single measurement that is continuous-valued with additive Gaussian noise

$$w \sim \mathcal{N}(0, 1) \quad (8.5.1-9)$$

Then

$$z = \phi + w \quad (8.5.1-10)$$

and the likelihood function of feature ϕ_i is

$$g_i(z) = \mathcal{N}(z; \phi_i, 1) \quad i = 1, 2 \quad (8.5.1-11)$$

Eq. (8.5.1-4) yields, after cancellations

$$p_1(1) = \frac{e^{-\frac{(z-\phi_1)^2}{2}}}{e^{-\frac{(z-\phi_1)^2}{2}} + e^{-\frac{(z-\phi_2)^2}{2}}} \quad (8.5.1-12)$$

If, say, $z = 10$, then the above yields

$$p_1(1) = \frac{1}{1 + e^{-2}} = 0.8808 \quad (8.5.1-13)$$

8.5.2 Multisensor Feature Estimation

Denote by Z^j , $j = 1, 2$, the data from the two sensors, each consisting of several measurements. The **local estimates**

$$p_i^j = P\{\phi = \phi_i | Z^j\} \quad j = 1, 2 \quad (8.5.2-1)$$

are calculated (separately) according to (8.5.1-6).

Unlike the state estimation in dynamic systems with data from multiple sensors, which requires in order to be optimal to process the entire data at a central location (cf. Subsection 8.3.5), an exact **global estimate**

$$p_i^G = P\{\phi = \phi_i | Z\} = P\{\phi = \phi_i | Z^1, Z^2\} \quad (8.5.2-2)$$

can be obtained in the present static case by fusing the local estimates (8.5.2-1).

A **fused** global estimate of the following form will be obtained:

$$p_i^G = f(p_i^1, p_i^2) \quad (8.5.2-3)$$

The above fusion, yet to be determined, is practical since

1. it requires significantly less data transmission,
2. it can be performed occasionally (when needed).

The Fusion Equation

Given $p_i^1, p_i^2, i = 1, \dots, n$, the “local” pmf of the target’s feature, the pmf conditioned on both data sets can be computed assuming the two data sets are conditionally independent, i.e.,

$$P(Z^1, Z^2 | \phi = \phi_i) = p(Z^1 | \phi = \phi_i)p(Z^2 | \phi = \phi_i) \quad \forall i \quad (8.5.2-4)$$

Using Bayes’ formula, one has the fused probability (8.5.2-3) as

$$\begin{aligned} p_i^G &= P\{\phi = \phi_i | Z^1, Z^2\} = \frac{1}{c} p(Z^1, Z^2 | \phi = \phi_i) P\{\phi = \phi_i\} \\ &= \frac{1}{c} p(Z^1 | \phi = \phi_i) P\{Z^2 | \phi = \phi_i\} p_i(0) \\ &= \frac{1}{c} \frac{p(Z^1 | \phi = \phi_i) p_i(0) p(Z^2 | \phi = \phi_i) p_i(0)}{p_i(0)} \end{aligned} \quad (8.5.2-5)$$

The above can be rewritten, similarly to [CB87], in the form (8.5.2-3) as

$$p_i^G = \frac{1}{c} \frac{p_i^1 p_i^2}{p_i(0)}$$

(8.5.2-6)

This is the final fusion equation, which is optimal if the conditional independence (8.5.2-4) — which amounts to requiring independent measurement noises — holds.

The General Fusion Equation

The general formula for two information sets with an arbitrary overlap is

$$p_i^G = P\{\phi = \phi_i | Z^1, Z^2\} = \frac{1}{c} \frac{P\{\phi = \phi_i | Z^1\} P\{\phi = \phi_i | Z^2\}}{P\{\phi = \phi_i | Z^1 \cap Z^2\}} \quad (8.5.2-7)$$

For more than two information sets one can use the above recursively, or the general formula in Ch. 8 of [Bar90].

Example

Assume each sensor makes one measurement for the optimal (centralized) processing as in (8.5.1-10). The “local” data are $Z^1 = z(1)$ and $Z^2 = z(2)$.

Taking for the sake of the example, $z(1) = z(2) = 10$, the centralized processing is done as follows. Using (8.5.1-12), one has the conditional probability of ϕ_1 given Z^1 as

$$p_1(1) = \frac{1}{1 + e^{-2}} = 0.8808 \quad (8.5.2-8)$$

Using recursion (8.5.1-6) to incorporate Z^2 into the conditioning one obtains

$$p_1(2) = \frac{0.8808}{0.8808 + e^{-2}(1 - 0.8808)} = 0.983 \quad (8.5.2-9)$$

as the optimal centralized probability of ϕ_1 .

The “local” probabilities of ϕ_1 at the two sensors are as in (8.5.2-8),

$$p_1^1 = p_1^2 = 0.8808 \quad (8.5.2-10)$$

With (8.5.2-10), the fusion equation (8.5.2-6) yields

$$p_1^G = \frac{(0.8808)^2}{(0.8808)^2 + (1 - 0.8808)^2} = 0.983 \quad (8.5.2-11)$$

i.e., the same value as (8.5.2-9).

8.5.3 Feature Estimate Fusion — Summary

8.5.3 Feature Estimate Fusion — Summary

The procedure to evaluate the *feature estimate* or *classification probability* of an object based on noisy measurements of features has been derived using:

- the prior probability of the object belonging to each class (having one of the finite number of possible features)
- the likelihood function of each feature (pdf of the measurement given the feature).

The following decentralized implementation has been considered:

- Each sensor carries out its classification, and
- The results are fused.

The fusion equation of the classification probabilities has been presented under the following assumption:

- the measurements are, conditioned on the feature, independent which is, in general, satisfied.

8.6 DISTRIBUTED APPROACHES TO MULTISENSOR TRACKING WITH OPTIMAL INFORMATION FUSION

8.6.1 Optimal Distributed Estimation with a Central Estimator

An implementation of the **Type IIa configuration**, which consists of **distributed estimation without feedback** — no global information is fed back to the local estimators — *with a central estimator* is discussed next.

In this situation there are, say, two sites at which the same target is tracked with **synchronized sensors**. At each site **local estimates** $\hat{x}^i(k|k)$, $i = 1, 2$, with the corresponding covariances $P^i(k|k)$, $i = 1, 2$, are computed.

In Section 8.4 it was shown that a central estimator, in order to combine these local estimates, has to account for the “cross-covariance” between the state estimation errors $\tilde{x}^1(k|k)$ and $\tilde{x}^2(k|k)$. The calculation of this “cross-covariance” is quite involved and not practical. Furthermore, the result is still not optimal, although it is quite close to the optimum.

The algorithm to be presented in the sequel shows how a central estimator can combine *optimally* the local estimates without having to calculate the above “cross-covariance” if it has access to

$$\{\hat{x}^i(k|k), P^i(k|k), \hat{x}^i(k|k-1), P^i(k|k-1), \quad i = 1, 2\} \quad (8.6.1-1)$$

In other words, full estimation is carried out locally and the results (8.6.1-1), consisting of the latest predicted *and* updated states, with their associated covariances, are needed to obtain the **optimal global estimate**.

8.6.1 Optimal Distributed Estimation with a Central Estimator

Derivation of the Central Estimate

The local measurements are *synchronized*

$$z^i(k) = H^i(k)x(k) + w^i(k) \quad i = 1, 2 \quad (8.6.1-2)$$

with $R^i(k)$ the corresponding measurement noise covariance.

Denote

$$z(k) = \begin{bmatrix} z^1(k) \\ z^2(k) \end{bmatrix} \quad (8.6.1-3)$$

$$H(k) = \begin{bmatrix} H^1(k) \\ H^2(k) \end{bmatrix} \quad (8.6.1-4)$$

$$R(k) = \begin{bmatrix} R^1(k) & 0 \\ 0 & R^2(k) \end{bmatrix} \quad (8.6.1-5)$$

The estimate at k is, using expression (5.2.3-17) of [BL93] for the filter gain,

$$\hat{x}^i(k|k) = \hat{x}^i(k|k-1) + P^i(k|k)H^i(k)'R^i(k)^{-1}[z^i(k) - H^i(k)\hat{x}^i(k|k-1)] \quad (8.6.1-6)$$

for the local estimators $i = 1, 2$, and without the superscript i for the central estimator.

Similarly, the covariance update equation is, from (5.2.3-16) of [BL93],

$$P^i(k|k)^{-1} = P^i(k|k-1)^{-1} + H^i(k)'R^i(k)^{-1}H^i(k) \quad (8.6.1-7)$$

The estimate (8.6.1-6) will be manipulated such that it can be expressed in terms of the local information (8.6.1-1) and the central prediction $\hat{x}(k|k-1)$, i.e.,

The central estimator will obtain the global estimate without the measurements, just by combining the local estimates.

Multiplying (8.6.1-6) by (8.6.1-7) yields for $i = 1, 2$

$$\begin{aligned} P^i(k|k)^{-1}\hat{x}^i(k|k) &= [P^i(k|k-1)^{-1} + H^i(k)'R^i(k)^{-1}H^i(k)]\hat{x}^i(k|k-1) \\ &\quad + P^i(k|k)^{-1}P^i(k|k)H^i(k)'R^i(k)^{-1}[z^i(k) - H^i(k)\hat{x}(k|k-1)] \\ &= P^i(k|k-1)^{-1}\hat{x}^i(k|k-1) + H^i(k)'R^i(k)^{-1}z^i(k) \end{aligned} \quad (8.6.1-8)$$

8.6.1 Optimal Distributed Estimation with a Central Estimator

Thus

$$H^i(k)'R^i(k)^{-1}z^i(k) = P^i(k|k)^{-1}\hat{x}^i(k|k) - P^i(k|k-1)^{-1}\hat{x}^i(k|k-1) \quad (8.6.1-9)$$

and this will be used to eliminate the measurements from the central estimator's update equation.

Using (8.6.1-3), (8.6.1-4) and taking advantage of the block-diagonal form of (8.6.1-5), the central state updating, given by (8.6.1-6) without superscript, can be rewritten as

$$\hat{x}(k|k) = \hat{x}(k|k-1) + P(k|k) \sum_{i=1}^2 H^i(k)'R^i(k)^{-1}[z^i(k) - H^i(k)\hat{x}(k|k-1)] \quad (8.6.1-10)$$

Similarly, the central covariance update, given by (8.6.1-7) without superscript, can be rewritten as

$$P(k|k)^{-1} = P(k|k-1)^{-1} + \sum_{i=1}^2 H^i(k)'R^i(k)^{-1}H^i(k) \quad (8.6.1-11)$$

Multiplying (8.6.1-10) with (8.6.1-11) yields, after cancellations

$$P(k|k)^{-1}\hat{x}(k|k) = P(k|k-1)^{-1}\hat{x}(k|k-1) + \sum_{i=1}^2 H^i(k)'R^i(k)^{-1}z^i(k) \quad (8.6.1-12)$$

The Optimal Fusion

Substituting (8.6.1-9) into (8.6.1-12), one obtains the ***fusion equation***

$$P(k|k)^{-1}\hat{x}(k|k) = P(k|k-1)^{-1}\hat{x}(k|k-1) + \sum_{i=1}^2 [P^i(k|k)^{-1}\hat{x}^i(k|k) - P^i(k|k-1)^{-1}\hat{x}^i(k|k-1)] \quad (8.6.1-13)$$

which is the central estimate in terms of only the local estimates.

The central covariance update (8.6.1-11) can be written in terms of the local covariances as

$$P(k|k)^{-1} = P(k|k-1)^{-1} + \sum_{i=1}^2 [P^i(k|k)^{-1} - P^i(k|k-1)^{-1}] \quad (8.6.1-14)$$

Eqs. (8.6.1-13) and (8.6.1-14) summarize the ***optimal fusion algorithm***.

8.6.1 Optimal Distributed Estimation with a Central Estimator

Figure 8.6.1-1 illustrates the flow of information in this configuration.

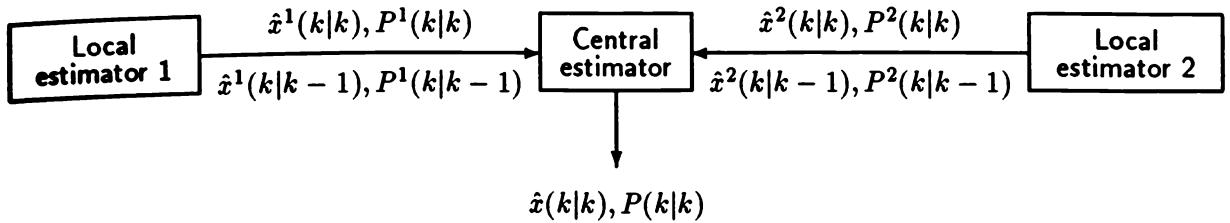


Figure 8.6.1-1: Information flow in a distributed estimation system with a central estimator.

Remark

This algorithm assumes that the central estimator runs at the same rate as the local ones.

A suboptimal version of the above would be to run the central estimator at a lower rate, e.g., every N sampling times. The corresponding counterparts of (8.6.1-13) and (8.6.1-14) are, in this case, obtained by replacing $k - 1$ with $k - N$.

8.6.2 Distributed Estimation without a Central Estimator

8.6.2.2 Distributed Estimation without a Central Estimator

The implementation of the **Type IIb configuration** — which is **distributed estimation with feedback** can be done with or without a central estimator.

The distributed estimation with feedback but *without a central estimator* is as follows: each node (local estimator) will perform, in addition to its local estimation function, also the function of the central estimator.

The resulting redundancy can enhance the reliability/survivability of the network.

Synchronized Sensors

In this configuration, in the case of **synchronized sensors** each node broadcasts its latest local state estimate for the target of interest, and, upon reception of its neighbors' similar broadcasts, calculates the “global” estimate.

In view of the past information exchange between the nodes (assuming perfect communication reliability), the previous local and global estimates coincide, i.e.,

$$\hat{x}^i(k|k-1) = \hat{x}(k|k-1) \quad (8.6.2-1)$$

$$P^i(k|k-1) = P(k|k-1) \quad (8.6.2-2)$$

The **fusion equation** for the estimates (8.6.1-13) becomes

$$P(k|k)^{-1}\hat{x}(k|k) = \sum_{i=1}^2 P^i(k|k)^{-1}\hat{x}^i(k|k) - P(k|k-1)^{-1}\hat{x}(k|k-1) \quad (8.6.2-3)$$

and for the covariances, (8.6.1-14) becomes

$$P(k|k)^{-1} = \sum_{i=1}^2 P^i(k|k)^{-1} - P(k|k-1)^{-1} \quad (8.6.2-4)$$

Remark

As in Subsection 8.6.1, the above equations yield the **optimal global estimate** and its covariance.

8.6.2 Distributed Estimation without a Central Estimator

Figure 8.6.2-1 illustrates the flow of information for this distributed system configuration for synchronized sensors.

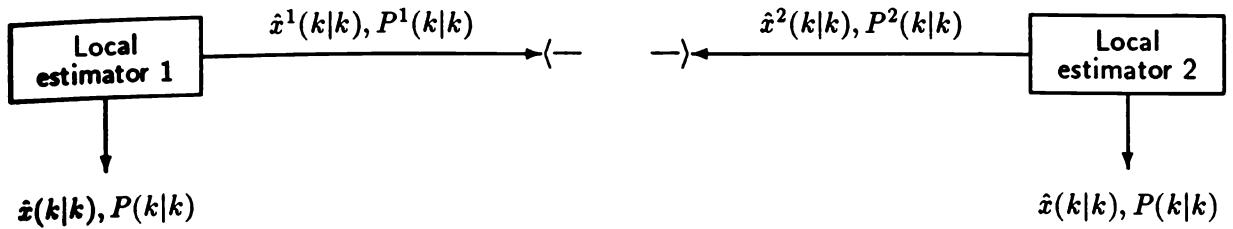


Figure 8.6.2-1: Information flow in a distributed estimation system without a central estimator (synchronized sensors).

Asynchronous Sensors

In the case of **asynchronous sensors** the following policy can be used:

- Each node broadcasts its latest estimate and each node replaces its estimate by the received estimate.

Assuming that

- the update/broadcast/reception are instantaneous
- no two updates occur simultaneously,

then each node will always have the latest central estimate.

The updates are done with time-varying sampling periods and the discrete-time state equations will have to take this into account accordingly.

Remark

In practice the above assumptions are not satisfied exactly and this scheme will yield only approximate optimal global estimates.

8.6.3 Information Fusion — Summary

The estimation of the state of a target using measurements from more than one sensor can be done in several ways:

- (1) Incorporate them sequentially, using a suitable association scheme, into the estimator — (centralized) Type IV configuration (Subsection 8.2.5, Section 4.5) — optimal
- (2) Obtain state estimates and covariances from local filters and fuse them *whenever needed* — (decentralized) Type IIa configuration — assuming the corresponding estimation errors
 - (i) independent (Subsection 8.2.3, Section 8.3) — grossly suboptimal
 - (ii) dependent due to the common process noise (Subsection 8.2.3, Section 8.4) — in this case the cross-covariance has to be calculated (or approximated) — slightly suboptimal
- (3) Obtain state estimates, one-step predicted states and their covariances and reconstruct indirectly algorithm (1) — also Type IIa configuration (Subsection 8.2.3, Section 8.1), but optimal (requires synchronized sensors)
- (4) Use feedback to the local estimators or broadcasting — Type IIb configuration (Subsection 8.2.3, Section 8.2) — optimal (for synchronized sensors); for asynchronous sensors a broadcasting/replacement policy yields nearly optimal estimates
- (5) For synchronized sensors one can first fuse the sensor data from the same time (static association) and then the central estimator carries out dynamic association (across time) and tracking — Type III configuration (Subsection 8.2.4, Section 8.8) — optimality hinges on the static association.

8.7 MULTISENSOR AIR TRAFFIC SURVEILLANCE — A CASE STUDY

8.7.1 Introduction

The objective of a surveillance and tracking system is to detect an *unknown* number of targets in its field of view and estimate the states (target position, velocity, acceleration, etc.) using sensor measurements contaminated by noise. This must be accomplished in the presence of spurious observations (due to background noise/clutter or sensor noise) and occasional missed detections by sensors.

The design of a tracking system consists of the following interrelated steps:

1. Selection of the state variable models used to represent the target motion and sensor measurements, including models of clutter and measurements uncertainties
2. Selection of the information processing configuration
3. Formulation of an index of desirability for each candidate measurement-target association
4. Determination of a consistent set of measurement-target associations
5. Estimation of target states.

Association is the decision process of *linking* measurements of a common origin (i.e., a target or false alarms) such that each measurement is associated with only one origin. A set of linked measurements can then be *statistically filtered* to *estimate* the states of targets.

In this section the maximum likelihood formulation of the data association problem is presented and solved using a sliding window 2-dimensional assignment algorithm¹: the established tracks are associated with the new measurements from the latest scan.

The results obtained with the interactive software MATSURV™ — Multisensor Air Traffic SURVeillance system for tracking multiple targets using measurements from **asynchronous sensors** with a **centralized configuration (Type IV)** are presented.

¹The nearly optimal association obtained using the 2-D assignment [PDBW92] for the measurement data base considered made it unnecessary to use the more general S-D algorithm.

8.7.2 Description of the Multisensor Data

8.7.2 Description of the Multisensor Data

In this Subsection a description of the *raw scan data* to be used by the *sensor data fusion processor*, which gathers and organizes the data from several sensors, is presented.

- The data from the fusion processor consists of scans from two L-band FAA radars located at Remsen (designated as “R”), and Dansville (“D”), NY.
- The scan periods are approximately 10 seconds. Each of these scans contains a number of primary radar or **skin returns**. The skin returns consists of a time stamp, a slant range and an azimuth angle measurement. For **cooperative targets** a secondary or **beacon return** is also obtained, which provides, in addition to the above, a **target identification number (ID or squawk)** and a target altitude measurement.
- The observability of the target state requires a full measurement of its position. The skin returns, which provide only a partial measurement of the target state, were not used in the tracking filter described in the sequel.

The target ID information from the beacon returns was used as follows:

- In the first phase all the measurements were associated with the appropriate targets based on the target ID, and the performance of the tracking filter was evaluated to validate and fine tune the filter.
- The availability of the ID of the measurements — the **ground truth** — provided a means for evaluating the association performance of the algorithm in the second phase.

Figure 8.7.2-1 shows the entire data set of measurements with ID, available from the two FAA radars.

8.7.2 Description of the Multisensor Data

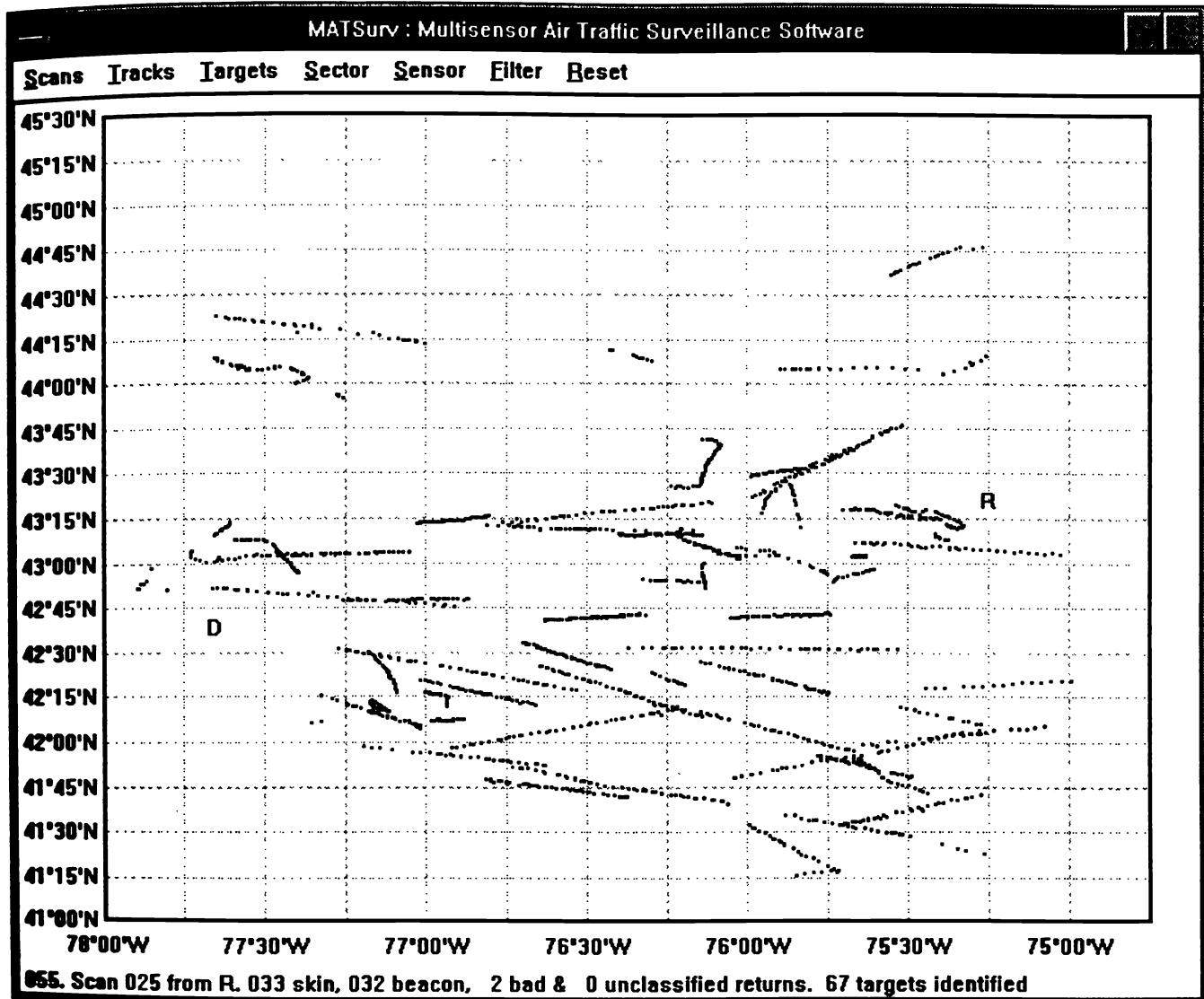


Figure 8.7.2-1: Raw measurements from sites D and R.

8.7.3 The Sensor Measurement Statistics

8.7.3 The Sensor Measurement Statistics

The measurement noises of the sensors are assumed to be zero-mean white with variances as follows.

For the range, the **resolution cells** are

$$\Delta r = \frac{c\tau}{2}, \quad (8.7.3-1)$$

where $c = 3 \cdot 10^8 \text{ m/s}$ and τ is the pulse width ($6\mu\text{s}$ for Remsen and $1.8\mu\text{s}$ for Dansville). It is assumed that the range measurement is uniformly distributed in each resolution cell, hence the standard deviation of the range measurement noise is $\sigma_r = \frac{\Delta r}{\sqrt{12}}$, which yields $0.260 \text{ km} = 0.140 \text{ nmi}$ for Remsen and $0.078 \text{ km} = 0.042 \text{ nmi}$ for Dansville.

The azimuth resolution cell is the horizontal beamwidth, which is 1.3° for Remsen and 1.2° for Dansville. Assuming the azimuth measurements uniformly distributed in a cell, the corresponding noise standard deviations were taken as $\sigma_\theta = 0.375^\circ = 6.55 \text{ mrad}$ for Remsen and $0.346^\circ = 6.05 \text{ mrad}$ for Dansville.

The altitude measurement noise was taken, based on FAA data, as

$$\sigma_h = \sqrt{50^2 + \frac{100^2}{12}} = 57.7350 \text{ ft} = 17.5976 \text{ m} \quad (8.7.3-2)$$

The above assumes 50ft RMS measurement noise and 100ft quantization. The latter is assumed to translate into a quantization noise uniformly distributed in an interval of 100, i.e., with variance $100^2/12$; while this noise is not white, it has a relatively small effect.

The Target State

After the k^{th} sampling interval (the intervals are not constant — this is discussed later) the estimated target state in geographical coordinates, denoted as $x_g(k|k)$, is a vector with the following components:

$$x_g(k|k) = \begin{bmatrix} x_\psi(k|k) & -\text{latitude } \in [-\pi/2, \pi/2] \\ x_\lambda(k|k) & -\text{longitude } \in [-\pi, \pi] \\ x_a(k|k) & -\text{altitude in km} \\ \dot{x}_\psi(k|k) & -\text{velocity due north in km/s} \\ \dot{x}_\lambda(k|k) & -\text{velocity due east in km/s} \\ \dot{x}_a(k|k) & -\text{vertical velocity up in km/s} \\ \ddot{x}_\psi(k|k) & -\text{acceleration due north in km/s}^2 \\ \ddot{x}_\lambda(k|k) & -\text{acceleration due east in km/s}^2 \end{bmatrix} \quad (8.7.4-1)$$

where the altitude is above MSL².

The state estimate $x_g(k|k)$ is in mixed dimensions (i.e., in angles and distances). Nevertheless, the covariance matrix $P_g(k|k)$ will be defined completely in terms of distance and distance rates only. The compatibility between the state in mixed dimensions and the covariance in uniform dimensions follows by interpreting latitude as *distance due north of equator* and longitude as *distance due east of Greenwich meridian*. Thus $P_g(k|k)$ is the covariance matrix associated with the distance errors due north, east and up and errors in their corresponding time derivatives.

A **local Cartesian coordinate frame** is defined based on the target state estimate $x_g(k|k)$. The origin of this reference frame is $[x_\psi(k|k) \ x_\lambda(k|k) \ 0]$ and the axes are oriented along the local north \mathbf{n}_k , east \mathbf{e}_k , and altitude \mathbf{u}_k directions. The symbol L_k is used to denote this local reference frame defined by the origin and the three coordinate axes.

²Mean sea level.

8.7.4 Design of the Tracking Filter

Measurement Conversions

The raw measurement vector $z(k+1)$ (consisting of slant range, azimuth and altitude) is first transformed into the geographical coordinates to become $z_g(k+1)$. The three components of $z_g(k+1)$ are the latitude, longitude and altitude.

The raw measurement covariance $R(k+1)$ in sensor coordinates is transformed to $R_g(k+1)$ in geographical coordinates via the first order approximation:

$$R_g = [\nabla_z z'_g] R [\nabla_z z'_g]' \quad (8.7.4-2)$$

The Jacobian $\nabla_z z'_g$ is evaluated at the raw measurement $z(k+1)$.

The measurement $z_g(k+1)$ and covariance $R_g(k+1)$ are then transformed into the local Cartesian axes L_k . This transformation yields $z_{L_k}(k+1)$ and $R_{L_k}(k+1)$. Note that the measurement $z_g(k+1)$ is in mixed dimensions (two angles and one distance) whereas $z_{L_k}(k+1)$ consists of all distance components. A similar mapping occurs in the transformation of the measurement covariance matrix from $R_g(k+1)$ to $R_{L_k}(k+1)$.

State Estimate Conversion

The *geographical state estimate* $x_g(k|k)$ and covariance $P_g(k|k)$ are mapped into the *local coordinate axes* L_k . This yields $x_{L_k}(k|k)$ and $P_{L_k}(k|k)$ as:

$$x_{L_k}(k|k) = [0 \ 0 \ x_a(k|k) \ \dot{x}_\psi(k|k) \ \dot{x}_\lambda(k|k) \ \dot{x}_a(k|k) \ \ddot{x}_\psi(k|k) \ \ddot{x}_\lambda(k|k)]^T \quad (8.7.4-3)$$

$$P_{L_k}(k|k) = P_g(k|k) \quad (8.7.4-4)$$

The equality of the covariances follows from the fact that the two coordinate systems differ by a translation only.

8.7.4 Design of the Tracking Filter

State Prediction

The predicted state $x_{L_k}(k+1|k)$ and covariance $P_{L_k}(k+1|k)$ are determined by assuming that the instantaneous target motion is linear along the local coordinate axes L_k , i.e.,

$$x_{L_k}(k+1|k) = \Phi_k \bar{x}_{L_k}(k|k) \quad (8.7.4-5)$$

$$P_{L_k}(k+1|k) = \Phi_k P_{L_k}(k|k) \Phi_k^T + \Gamma_k Q_k \Gamma_k^T \quad (8.7.4-6)$$

where the system matrices are given by

$$\Phi_k = \begin{bmatrix} 1 & 0 & 0 & \delta_k & 0 & 0 & \frac{1}{2}\delta_k^2 & 0 \\ 0 & 1 & 0 & 0 & \delta_k & 0 & 0 & \frac{1}{2}\delta_k^2 \\ 0 & 0 & 1 & 0 & 0 & \delta_k & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \delta_k & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \delta_k \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Gamma_k = \begin{bmatrix} \frac{1}{2}\delta_k^2 & 0 & 0 \\ 0 & \frac{1}{2}\delta_k^2 & 0 \\ 0 & 0 & \frac{1}{2}\delta_k^2 \\ \delta_k & 0 & 0 \\ 0 & \delta_k & 0 \\ 0 & 0 & \delta_k \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (8.7.4-7)$$

and $\delta_k = t_{k+1} - t_k$ is the sampling time interval³.

The process noise covariance matrix Q_k is of the form

$$Q_k = \begin{bmatrix} \sigma_n^2(k) & 0 & 0 \\ 0 & \sigma_e^2(k) & 0 \\ 0 & 0 & \sigma_a^2(k) \end{bmatrix} \quad (8.7.4-8)$$

The choice of $\sigma_n(k)$, $\sigma_e(k)$ and $\sigma_a(k)$ is a critical design issue. Since $\sigma_n(k)$ and $\sigma_e(k)$ are process noise levels along the north and east axes, they are set to the same value, i.e., $\sigma_n(k) = \sigma_e(k) = \sigma_h(k)$ where h denotes horizontal. The vertical motion of the target is *more predictable* than along the horizontal axes. Hence, the process noise along the altitude axis, $\sigma_a(k)$, is different from $\sigma_h(k)$.

³The time interval δ_k can in general be negative, particularly when the detection at t_{k+1} and the detection at t_k originate from different radars. The implementation of the filter for negative update time intervals is discussed in Subsection 2.2.4.

8.7.4 Design of the Tracking Filter

Both the horizontal, $\sigma_h(k)$, and the altitude, $\sigma_a(k)$, process noise levels are chosen according to the function shown in Figure 8.7.4-2.

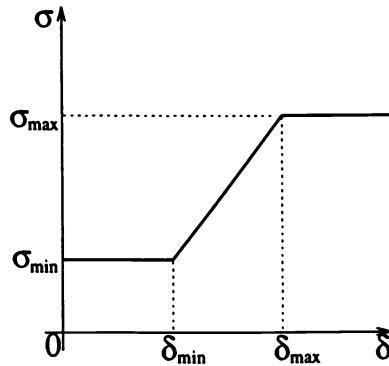


Figure 8.7.4-2: Process noise level as a function of sampling interval.

For $\sigma_h(k)$ the following values were chosen

$$\sigma_{h_{\max}} = 7.5 \text{ m/s}^2 \quad \sigma_{h_{\min}} = 0.0 \text{ m/s}^2 \quad (8.7.4-9)$$

$$\delta_{h_{\min}} = 0.0 \text{ s} \quad \delta_{h_{\max}} = 40.0 \text{ s} \quad (8.7.4-10)$$

and for $\sigma_a(k)$ the following values were chosen

$$\sigma_{a_{\max}} = 0.5 \text{ m/s}^2 \quad \sigma_{a_{\min}} = 0.0 \text{ m/s}^2 \quad (8.7.4-11)$$

$$\delta_{a_{\min}} = 1.0 \text{ s} \quad \delta_{a_{\max}} = 40.0 \text{ s} \quad (8.7.4-12)$$

8.7.4 Design of the Tracking Filter

State Update

The Kalman filter state and covariance update equations are implemented in the local Cartesian coordinates. These update equations yield $x_{L_k}(k+1|k+1)$ and $P_{L_k}(k+1|k+1)$.

In addition one gets the innovations $\nu_{L_k}(k+1)$ and the innovations covariance $S_{L_k}(k+1)$ which are used to gauge the performance of the filter.

$$\nu_{L_k}(k+1) = z_{L_k}(k+1) - H x_{L_k}(k+1|k) \quad (8.7.4-13)$$

$$S_{L_k}(k+1) = H P_{L_k}(k+1|k) H' + R_{L_k}(k+1) \quad (8.7.4-14)$$

$$W_{L_k}(k+1) = P_{L_k}(k+1|k) H' S_{L_k}(k+1)^{-1} \quad (8.7.4-15)$$

$$x_{L_k}(k+1|k+1) = x_{L_k}(k+1|k) + W_{L_k}(k+1) \nu_{L_k}(k+1) \quad (8.7.4-16)$$

$$P_{L_k}(k+1|k+1) = P_{L_k}(k+1|k) - W_{L_k}(k+1) S_{L_k}(k+1) W_{L_k}(k+1)' \quad (8.7.4-17)$$

where $H = [I \ 0 \ 0]$ is a 3×8 matrix and the 8×3 matrix $W_{L_k}(k+1)$ is the filter gain.

Updated State Conversion

The updated state $x_{L_k}(k+1|k+1)$ and covariance $P_{L_k}(k+1|k+1)$ are obtained in the L_k reference frame. To complete the cycle of transformations, the updated state $x_{L_k}(k+1|k+1)$ and covariance $P_{L_k}(k+1|k+1)$, which are in L_k , are transformed into geographical coordinates, yielding $x_g(k+1|k+1)$ (which determines L_{k+1}) and $P_g(k+1|k+1)$, respectively. The two equations below illustrate the sequence of operations that are involved in this transformation.

$$x_{L_k}(k+1|k+1) \implies x_g(k+1|k+1) \implies L_{k+1} \quad (8.7.4-18)$$

$$P_{L_k}(k+1|k+1) \implies P_g(k+1|k+1) \quad (8.7.4-19)$$

Initialization

The filter is initialized using the first two measurements, i.e., $x_g(2|2)$ and $P_g(2|2)$ are determined using two point differencing from $z_g(1)$, $R_g(1)$ and $z_g(2)$, $R_g(2)$. Since only two measurements are used the initial acceleration components are both set to zero.

A gate based on the maximum velocity of Mach 5 limited the number of candidate measurements in the initial association. If there was more than one candidate, the subsequent assignment resolved the ambiguity.

8.7.5 Data Association

The formulation of the 2-D assignment problem is presented below. Consider the scenario at scan $k > 1$. There are $n(k)$ tracks from the previous assignments, and corresponding to track i , $i = 1, \dots, n(k)$, one have the latest state estimate $x_g^{(i)}(k|k)$ and covariance $P_g^{(i)}(k|k)$. Let scan $k + 1$ contain $m(k + 1)$ measurements.

The hypotheses upon which the 2-D assignment algorithm is based on are the following:

- Measurement j , $1 \leq j \leq m(k + 1)$, originated from a target corresponding to one of the $n(k)$ validated tracks, say track i , $i = 1, \dots, n(k)$.
- Measurement j , $1 \leq j \leq m(k + 1)$, is a false alarm. Track index $i = 0$ is used to designate a *dummy target* (i.e., a source of false alarms). Such a measurement is also kept as a candidate for a new track, i.e., if within $T_1 = 30s$ it has another measurement in its neighborhood (in a gate based on the maximum expected velocity) then it initiates a new track.

Each track (excluding track 0) is assigned at most one measurement, and each measurement is assigned to at most one track. On the other hand, the number of measurements that may be assigned to track 0 is not limited.

An existing track is dropped if within $T_2 = 60s$ no new measurement is associated with it.

Feasible Assignments

We shall define an assignment ω , as the mapping between the measurement indices j and the track indices i . This assignment can be represented using the set of binary variables $\{\rho_{ij}\}$.

A feasible assignment must satisfy the following requirements

$$\omega = \{\rho_{ij} \in \{0, 1\} : i = 0, \dots, n(k), j = 1, \dots, m(k + 1)\} \quad (8.7.5-1)$$

$$\sum_{i=0}^{n(k)} \rho_{ij} = 1 \quad j = 1, \dots, m(k + 1) \quad (8.7.5-2)$$

$$\sum_{j=1}^{m(k+1)} \rho_{ij} = 1 \quad i = 1, \dots, n(k) \quad (8.7.5-3)$$

The 2-D Assignment Problem and its Complexity

Let Ω be the set of all feasible assignments, then the total number of feasible assignments, i.e., the cardinality (number of elements) of the set Ω , can be shown to be

$$|\Omega| = \sum_{i=0}^p \left(\frac{p!}{i!} \right) \binom{q}{p-i} \quad (8.7.5-4)$$

where $p = \min\{n(k), m(k+1)\}$ and $q = \max\{n(k), m(k+1)\}$; $n(k)$ is the number of tracks at time k (list 1) and $m(k+1)$ is the number of measurements at $k+1$ (list 2).

Let c_{ij} be the cost of assigning measurement j to track i . The 2-D assignment problem can now be stated simply as finding an optimal assignment ω^* that minimizes the overall cost, i.e.,

$$\omega^* = \arg \min_{\omega \in \Omega} \sum_{i=0}^{n(k)} \sum_{j=1}^{m(k+1)} \rho_{ij} c_{ij} \quad (8.7.5-5)$$

The number of feasible assignments, $|\Omega|$, is very large even for moderate values of p and q , hence an efficient assignment algorithm is required for solving this problem. The *modified Auction algorithm* is ideally suited for solving this 2-D assignment problem. A detailed description of this algorithm can be found in [PDBW92].

In the present problem of assigning measurements to tracks, the cost criterion c_{ij} is the negative logarithm of the ratio of the likelihood of measurement j originating from track $i \neq 0$ to the likelihood of measurement j originating from track 0 (i.e., that it is a false alarm).

The Likelihood Ratio Cost

The evaluation of c_{ij} is presented in the following. The likelihood that measurement j , $1 \leq j \leq m(k+1)$, originated from track i , $1 \leq i \leq n(k)$, is given by

$$\Lambda_{ij}(k+1) = P_D \cdot |2\pi S_{ij}(k+1)|^{-\frac{1}{2}} \cdot \exp \left\{ -\frac{1}{2} \eta_{ij}(k+1) \right\} \quad (8.7.5-6)$$

where P_D is the probability of detection, $\eta_{ij}(k+1)$ is the normalized innovation squared given by

$$\eta_{ij}(k+1) = \nu_{ij}(k+1)' [S_{ij}(k+1)]^{-1} \nu_{ij}(k+1) \quad (8.7.5-7)$$

$\nu_{ij}(k+1)$ is the innovation and $S_{ij}(k+1)$ the innovation covariance associated with track i updated with measurement j .

It is assumed that false alarms are uniformly probable in the whole surveillance region. However, since gating is used, a candidate false alarm is uniformly distributed in the gate volume denoted as $\Psi(k+1)$. Since false alarms are assigned to track 0, the likelihood that the measurement j is a false alarm is,

$$\Lambda_{0j}(k+1) = \frac{1}{\Psi(k+1)} \quad (8.7.5-8)$$

The **assignment cost** c_{ij} for measurement j to track i is the negative log-likelihood ratio, given by

$$c_{ij} = -\ln \left[\frac{\Lambda_{ij}(k+1)}{\Lambda_{0j}(k+1)} \right] = \frac{1}{2} \eta_{ij}(k+1) + \ln \left(\frac{|2\pi S_{ij}(k+1)|^{\frac{1}{2}}}{P_D \Psi(k+1)} \right) \quad (8.7.5-9)$$

Note that a *positive cost* c_{ij} indicates that the origin of measurement j is more likely to be *other than track i* — it is an **extraneous measurement**.

The following values have been used for the probability of detection P_D and the surveillance region volume of the region in which a candidate false alarm is uniformly distributed:

$$P_D = 0.95 \quad (8.7.5-10)$$

$$\Psi(k+1) = 1000 \text{ km}^3 \quad (8.7.5-11)$$

8.7.6 Real Data Examples

The performance of the tracking filter and the association algorithm is illustrated for 3 typical target trajectories.

In Case 1, the association algorithm perfectly matches the measurement ID data, and hence the output of the tracking filter with ID and with association are identical. These results are shown in Figures 8.7.6-1 through 8.7.6-8. Note that the tracking filter handles the target maneuver, which is a typical landing pattern, successfully. The negative log-likelihood ratio, shown in Figure 8.7.6-8, if negative, indicates that “target” is more likely than “false” for the measurement under consideration.

In Case 2 the trajectory formed using the association algorithm is *superior* to the one using the IDs. The results for this case are shown in Figures 8.7.6-9 through 8.7.6-20. This case is unusual because it appears that the ID was changed toward the end of the flight (before landing). The ID-based association came up with 19 measurements while the assignment algorithm obtained the same 19 plus another 6 by which time there was a new ID but the measurements gave a good track. This shows the superiority of the assignment algorithm over the ***ground truth*** which has its quirks. Once again note that the tracking filter is quite robust in handling the rather sharp target maneuver.

Case 3, described in Figures 8.7.6-21 through 8.7.6-32, illustrates two additional beneficial aspects of the association algorithm. First the association algorithm drops a measurement that is very noisy — ***outlier rejection*** — thus yielding a better track than obtained using the ID. Secondly, it brings in a measurement that has been (incorrectly) assigned an ID corresponding to a different target. Comparing the tracks shown in Figures 8.7.6-25 and 8.7.6-26 it can be seen that dropping the outlying measurement definitely improves the track. The normalized innovations squared shown in Figures 8.7.6-29 and 8.7.6-30 clearly indicate that the measurement that is dropped is an outlier while the new measurement (with a different ID) definitely belongs to this target.⁴ The reason the association algorithm replaced the outlier can be seen from Figures 8.7.6-31 and 8.7.6-32: the outlier has a positive cost — a log-likelihood ratio indicating that it is more likely to have other origin than the track under consideration.

⁴This is a known phenomenon of transponder interference when two aircraft are close. In this case the “*ground truth*” is not completely true (ID codes have been switched due to the interference).

8.7.6 Real Data Examples

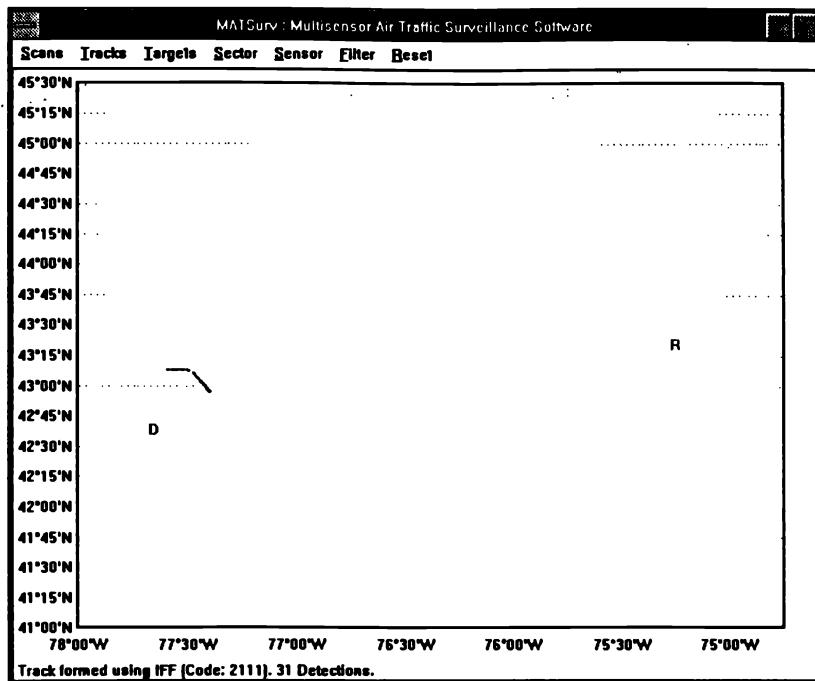


Figure 8.7.6-1: Case 1. Track using ID.

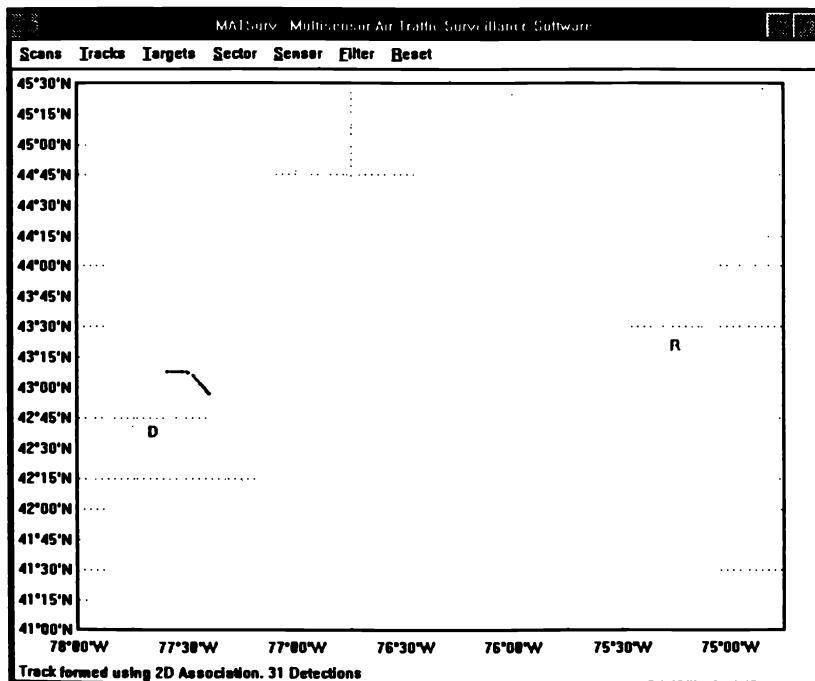


Figure 8.7.6-2: Case 1. Track using association (perfectly matches with the IDs).

8.7.6 Real Data Examples

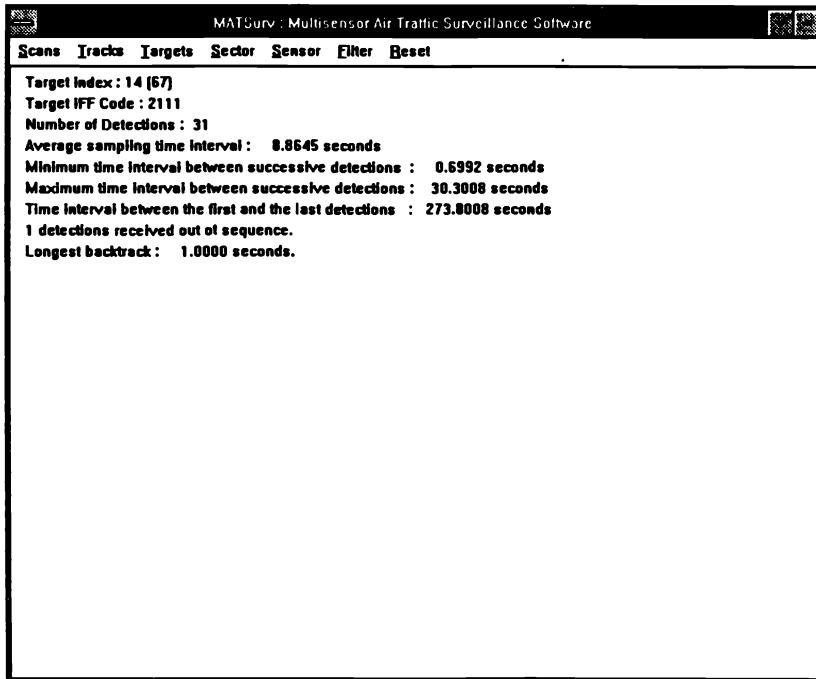


Figure 8.7.6-3: Case 1. Target trajectory information using ID.

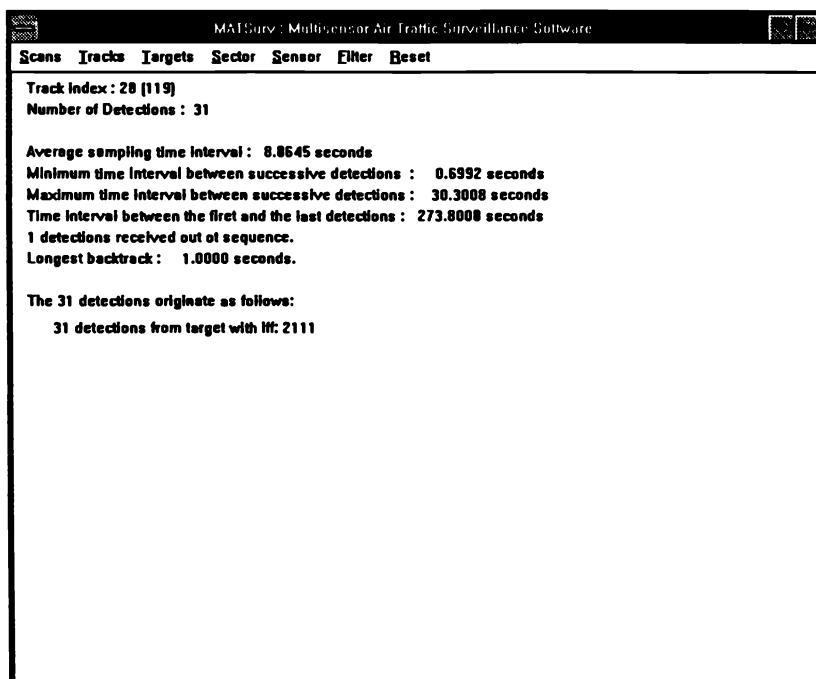


Figure 8.7.6-4: Case 1. Target trajectory information using association.

8.7.6 Real Data Examples

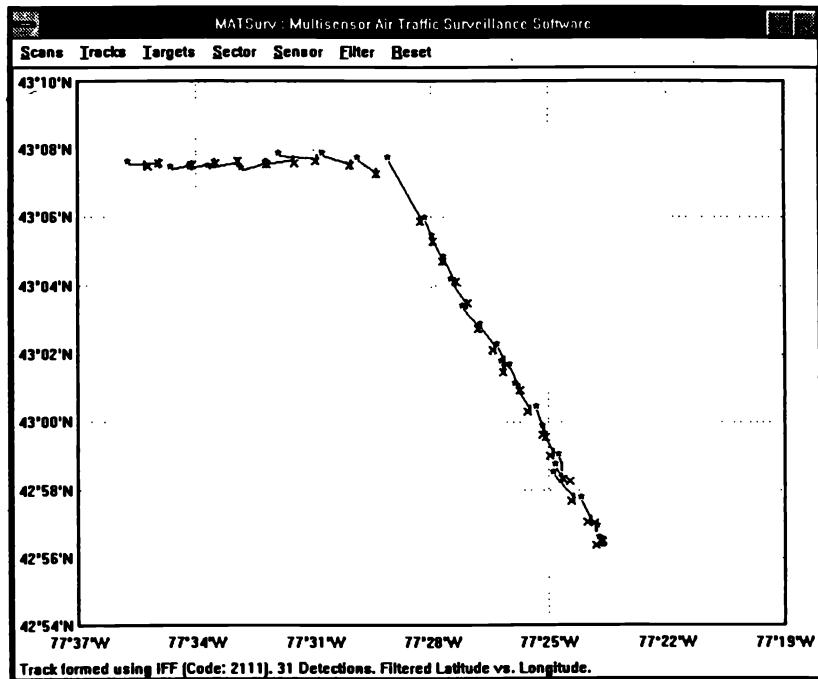


Figure 8.7.6-5: Case 1. Estimated trajectory of target in the horizontal plane.

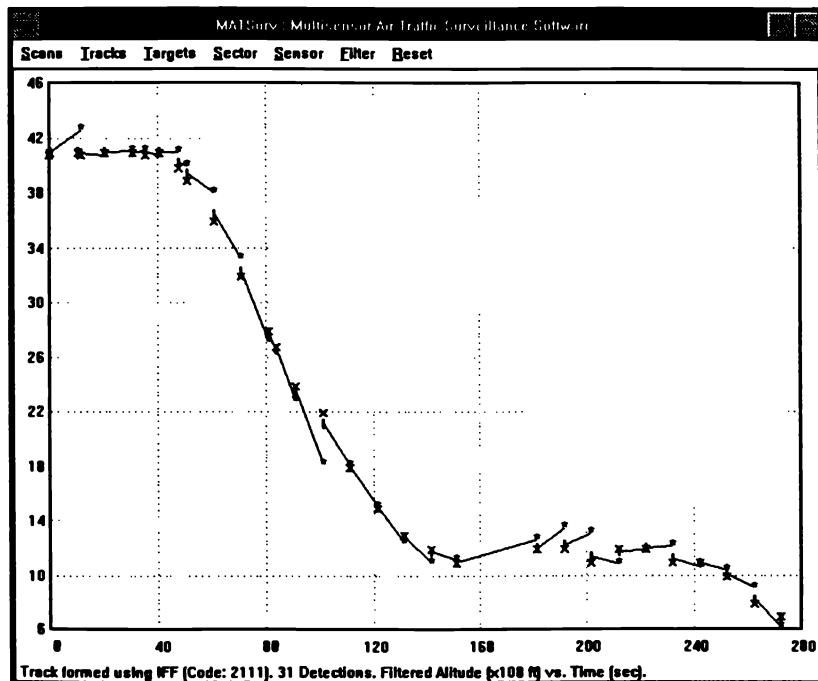


Figure 8.7.6-6: Case 1. Estimated altitude of target.

8.7.6 Real Data Examples

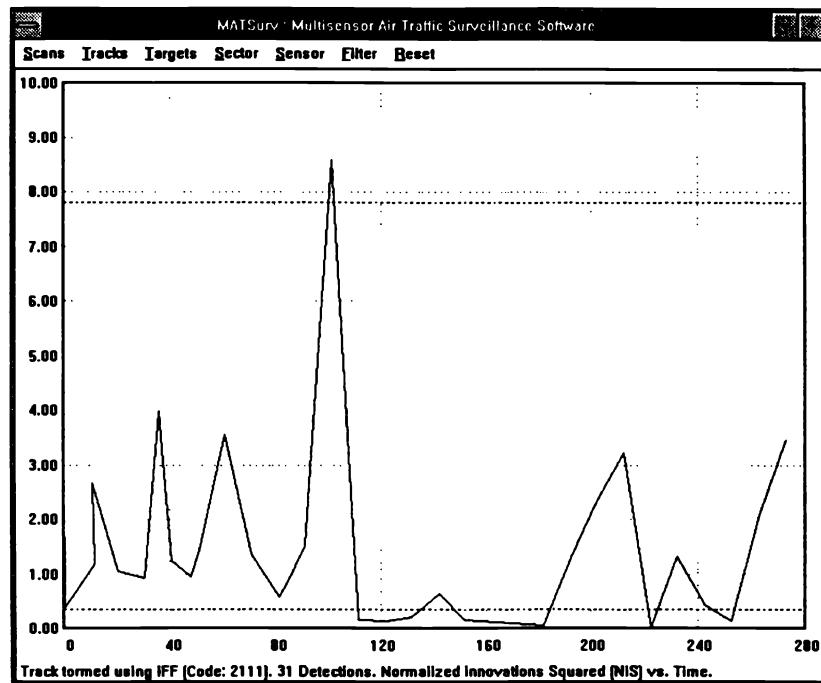


Figure 8.7.6-7: Case 1. Normalized innovation squared (in 3 dimensions).

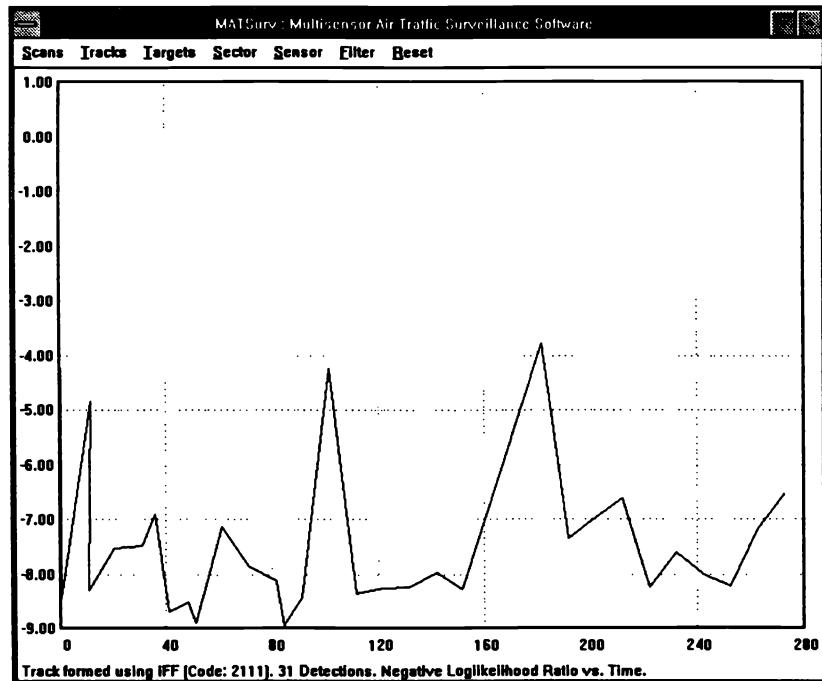


Figure 8.7.6-8: Case 1. Negative log-likelihood ratio.

8.7.6 Real Data Examples

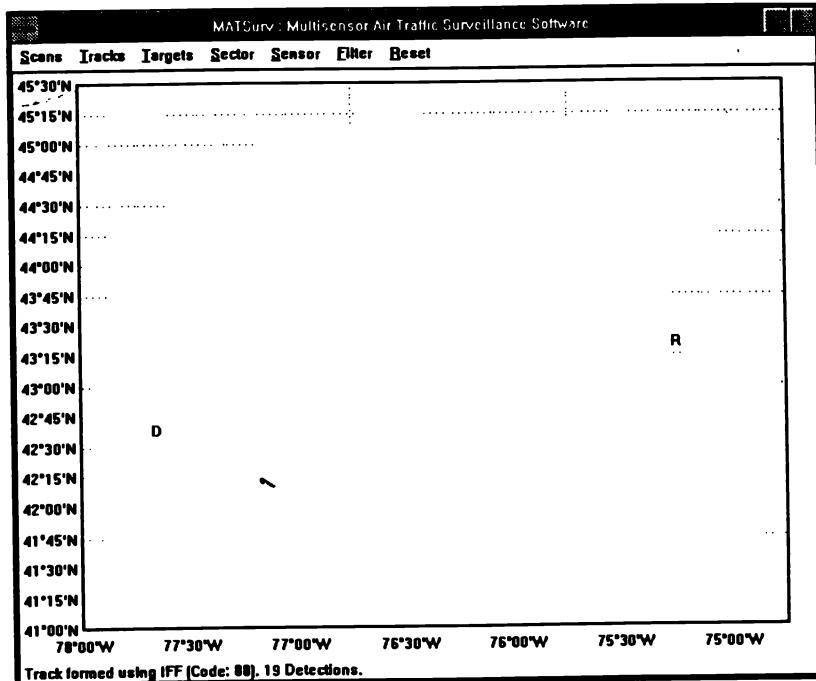


Figure 8.7.6-9: Case 2. Target trajectory using ID.

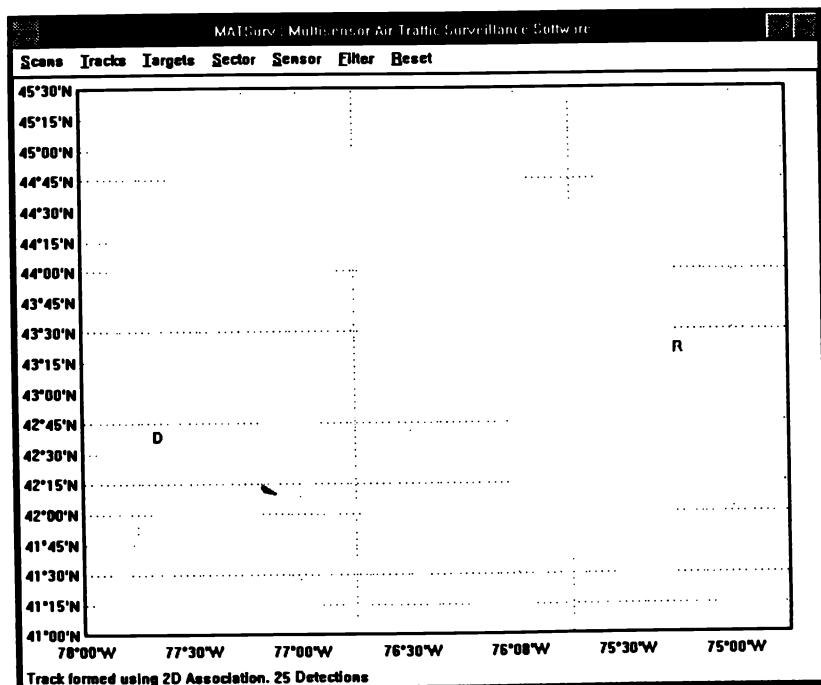


Figure 8.7.6-10: Case 2. Target trajectory using association.

8.7.6 Real Data Examples

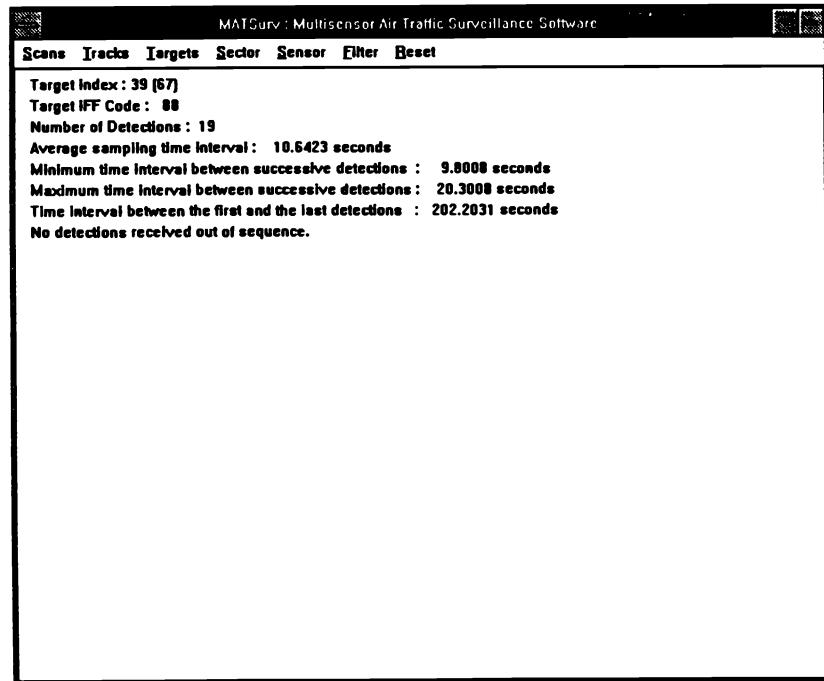


Figure 8.7.6-11: Case 2. Target trajectory information using ID (19 detections: ID changed in flight).

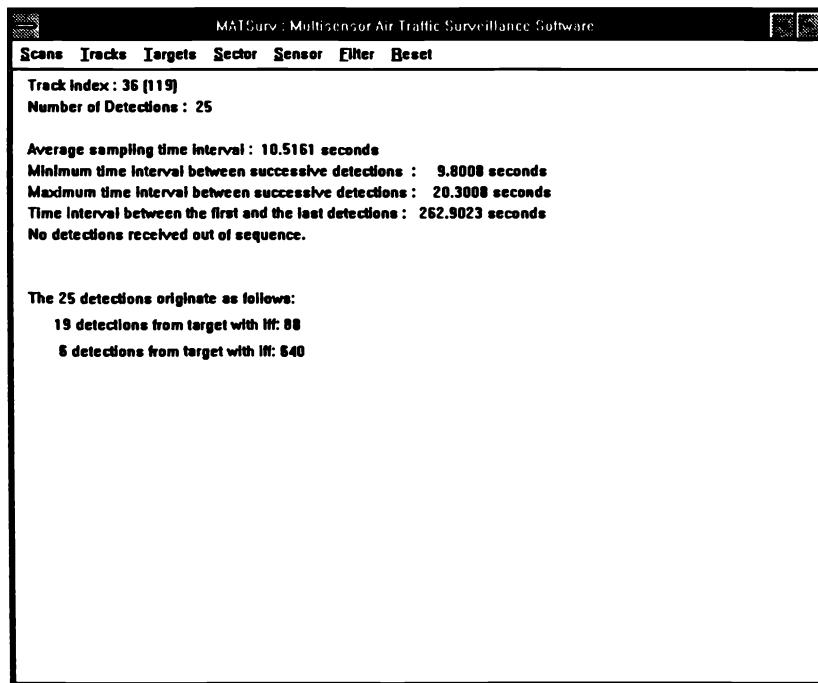


Figure 8.7.6-12: Case 2. Target trajectory information using association (25 detections).

8.7.6 Real Data Examples

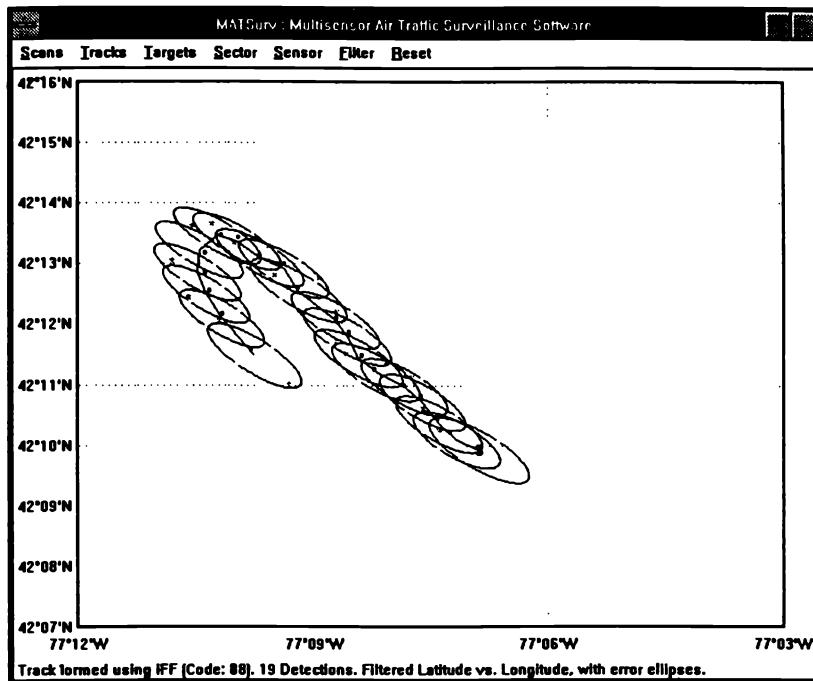


Figure 8.7.6-13: Case 2. Estimated trajectory using ID (19 detections).

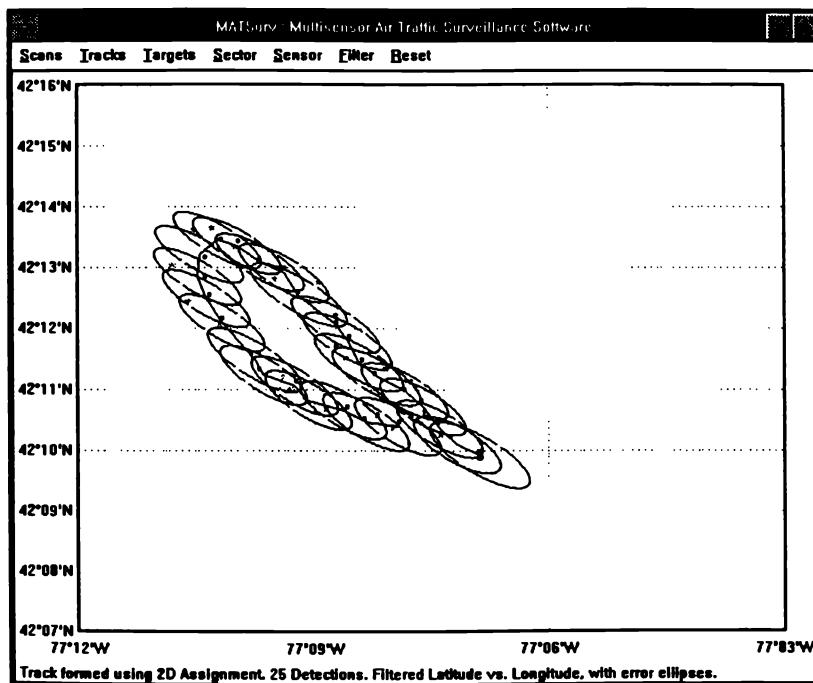


Figure 8.7.6-14: Case 2. Estimated trajectory using association (25 detections).

8.7.6 Real Data Examples

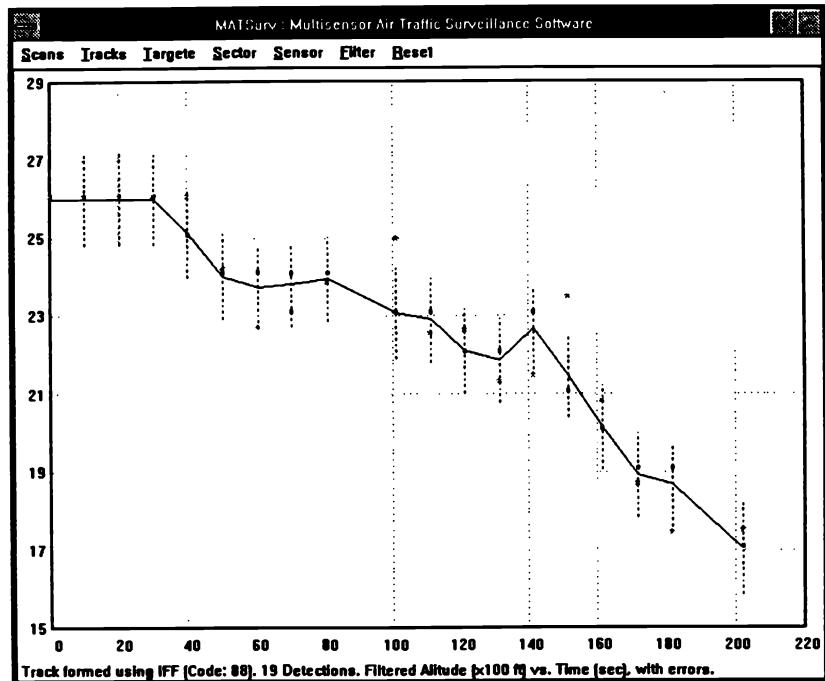


Figure 8.7.6-15: Case 2. Estimated altitude using ID (19 detections).

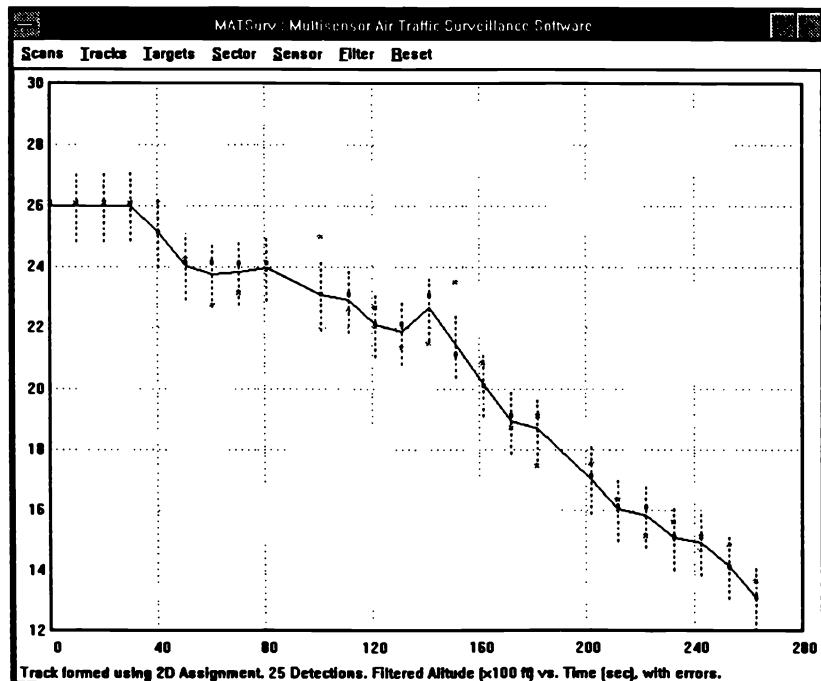


Figure 8.7.6-16: Case 2. Estimated altitude using association (25 detections).

8.7.6 Real Data Examples

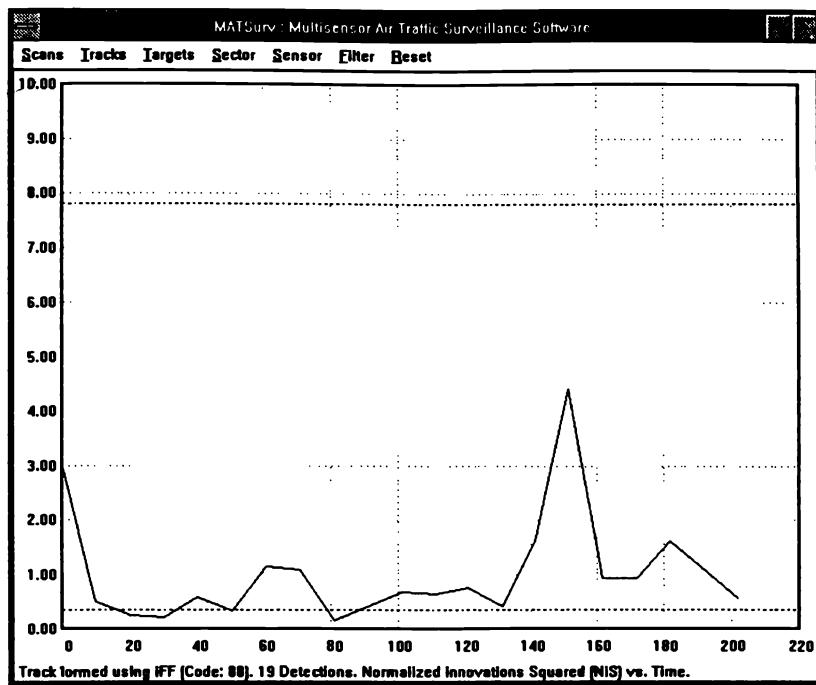


Figure 8.7.6-17: Case 2. Normalized innovation squared using ID (19 detections).

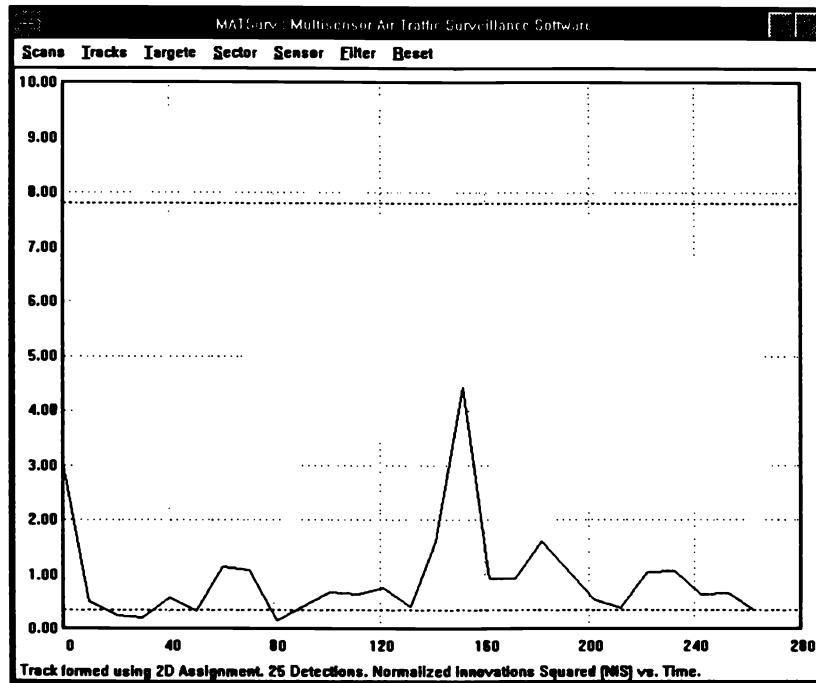


Figure 8.7.6-18: Case 2. Normalized innovation squared using association (25 detections).

8.7.6 Real Data Examples

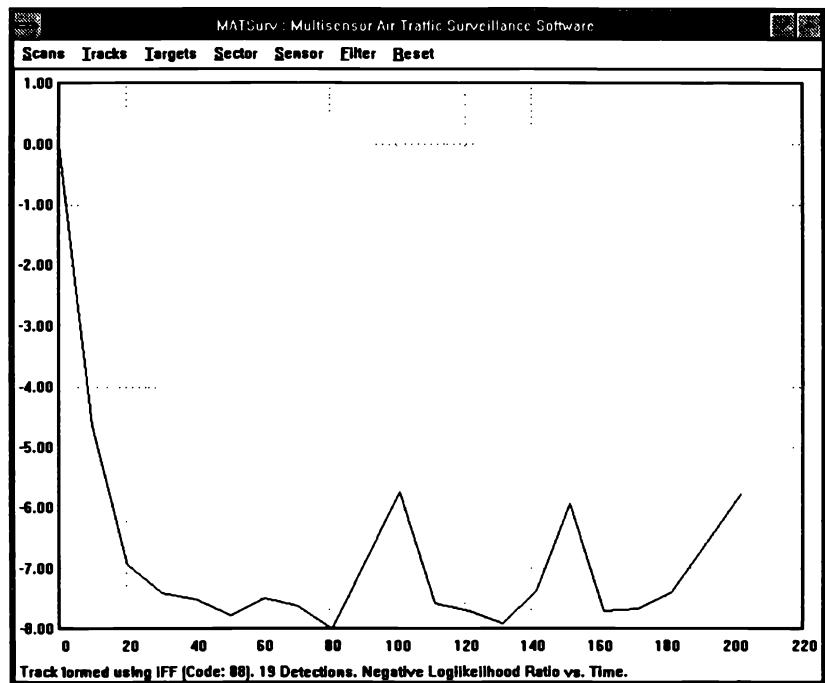


Figure 8.7.6-19: Case 2. Negative log-likelihood ratio using ID (19 detections).

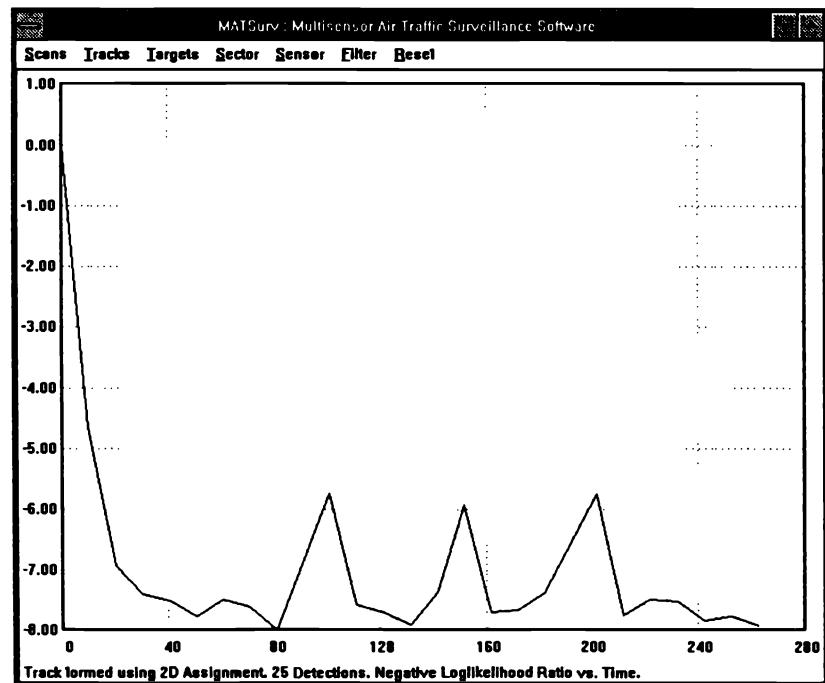


Figure 8.7.6-20: Case 2. Negative log-likelihood ratio using association (25 detections).

8.7.6 Real Data Examples

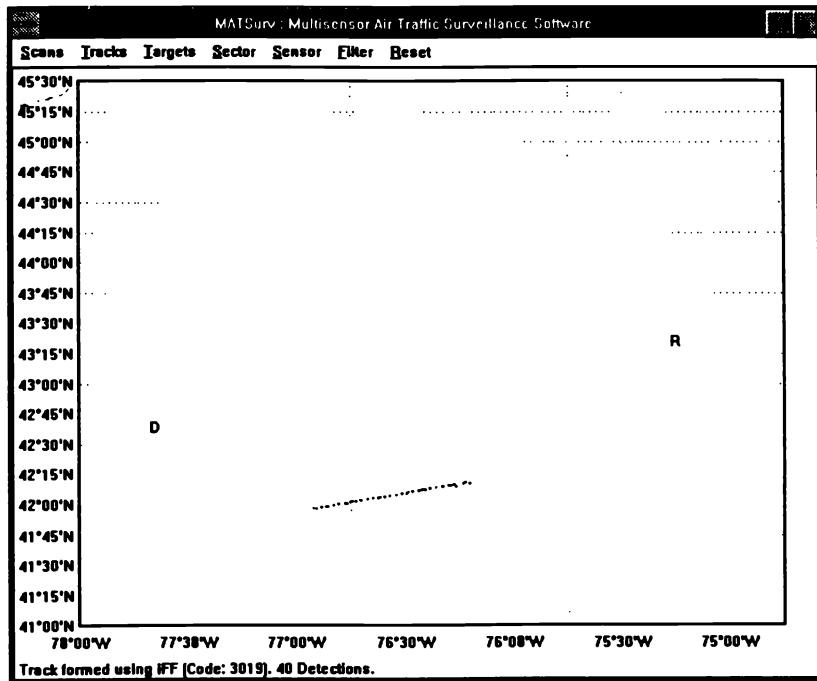


Figure 8.7.6-21: Case 3. Target trajectory using ID.

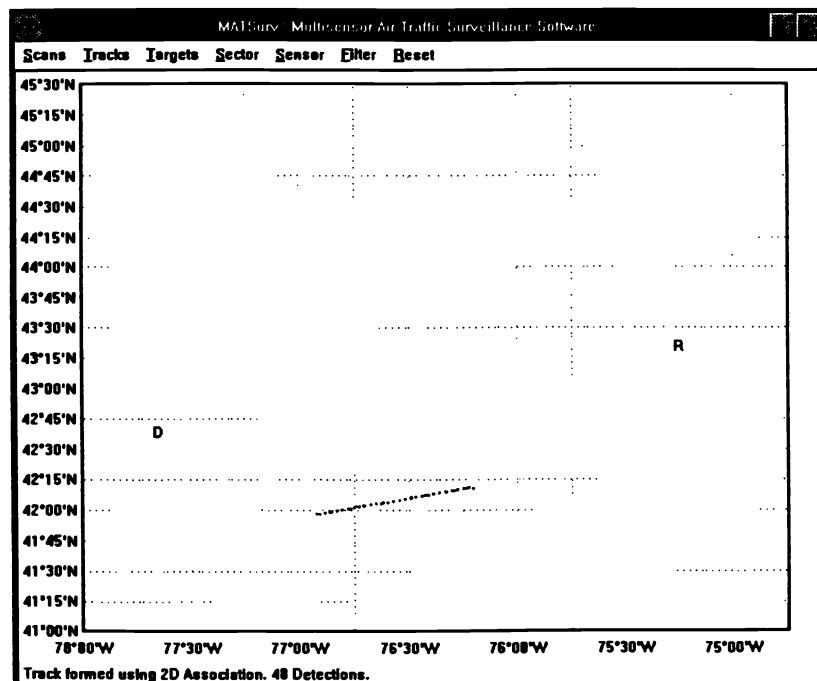


Figure 8.7.6-22: Case 3. Target trajectory using association.

8.7.6 Real Data Examples

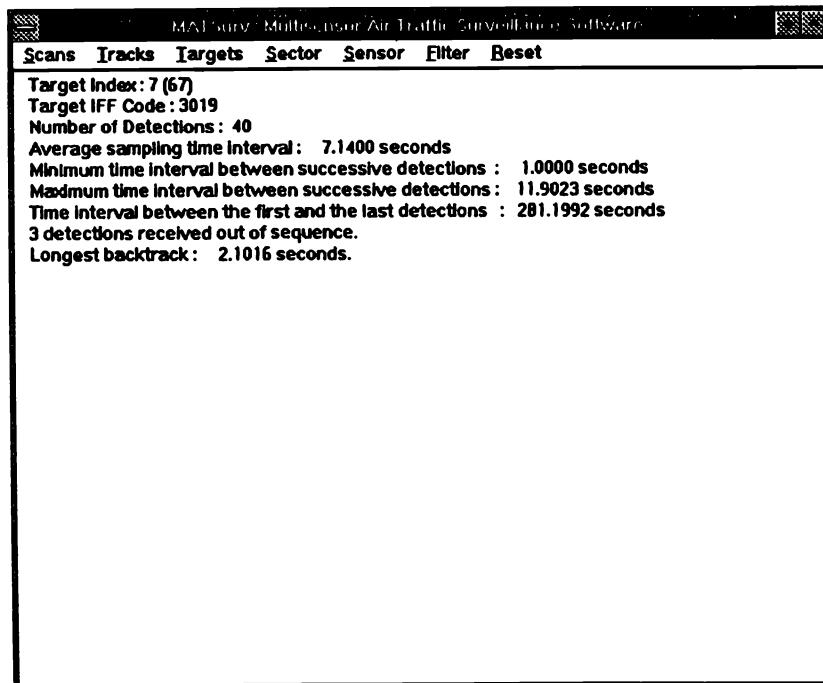


Figure 8.7.6-23: Case 3. Target trajectory information using ID (40 detections).

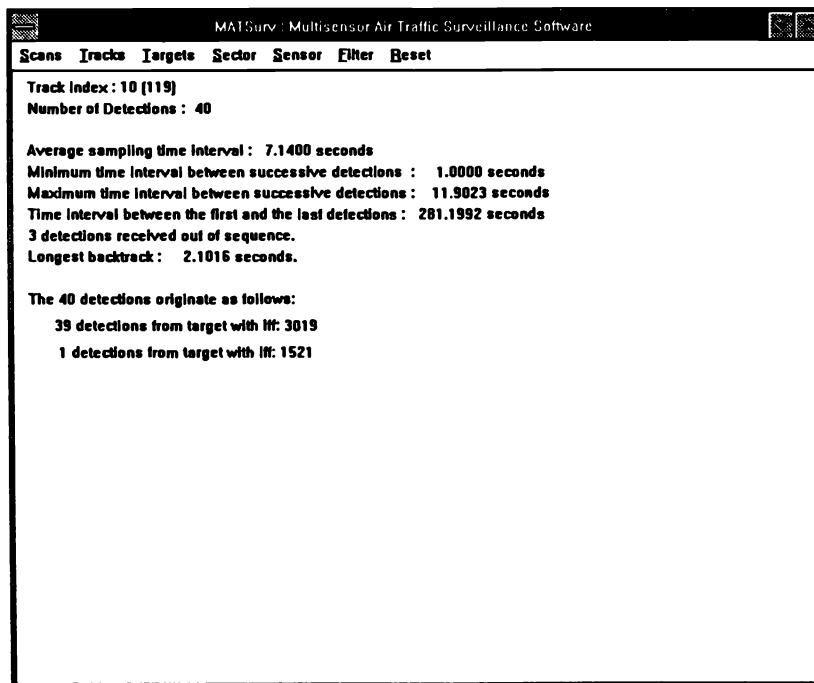


Figure 8.7.6-24: Case 3. Target trajectory information using association (40 detections: one detection with ID 3019 is discarded, and one detection with ID 1521 is added).

8.7.6 Real Data Examples

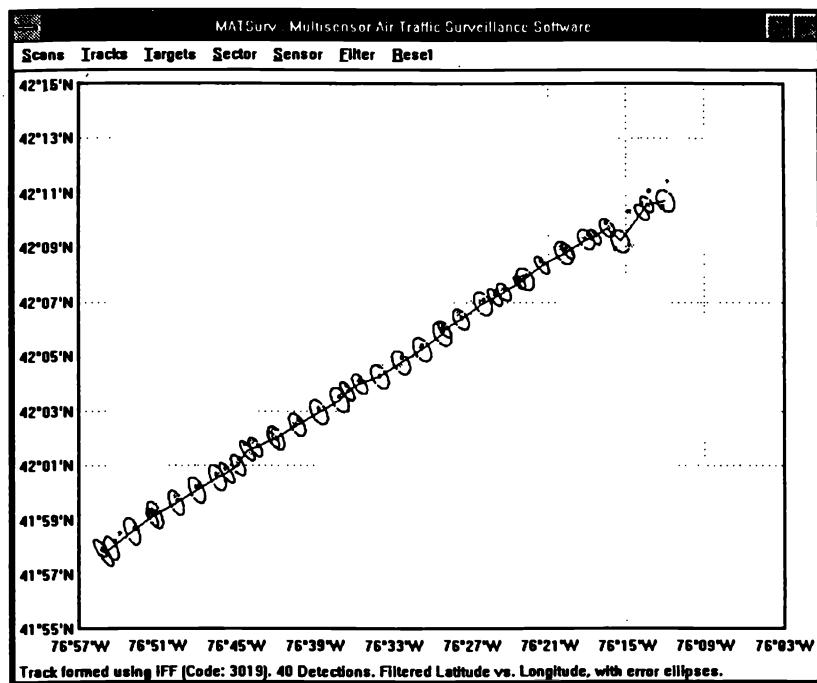


Figure 8.7.6-25: Case 3. Estimated trajectory using ID with outlying detection near top right corner of track.

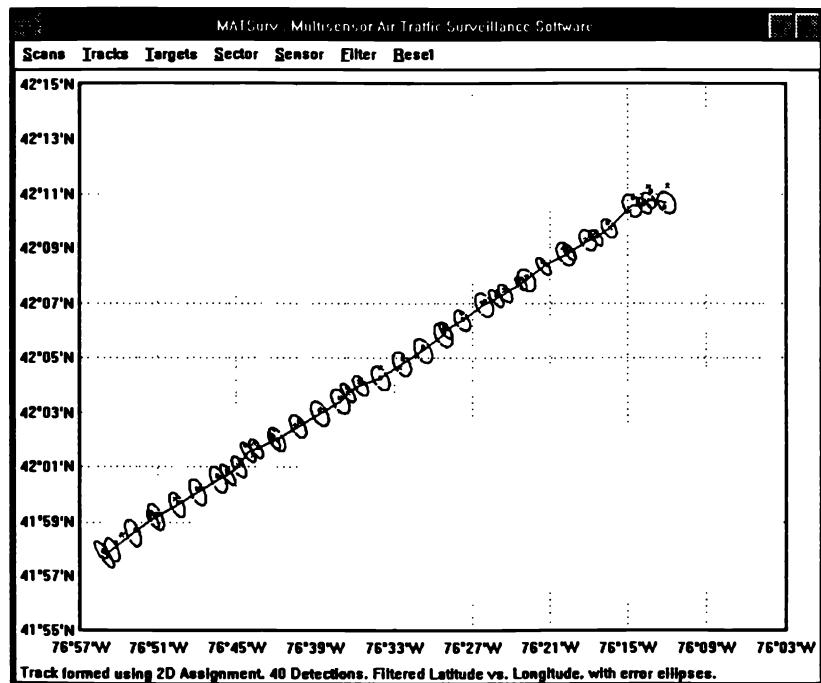


Figure 8.7.6-26: Case 3. Estimated trajectory using association, with outlier replaced.

8.7.6 Real Data Examples

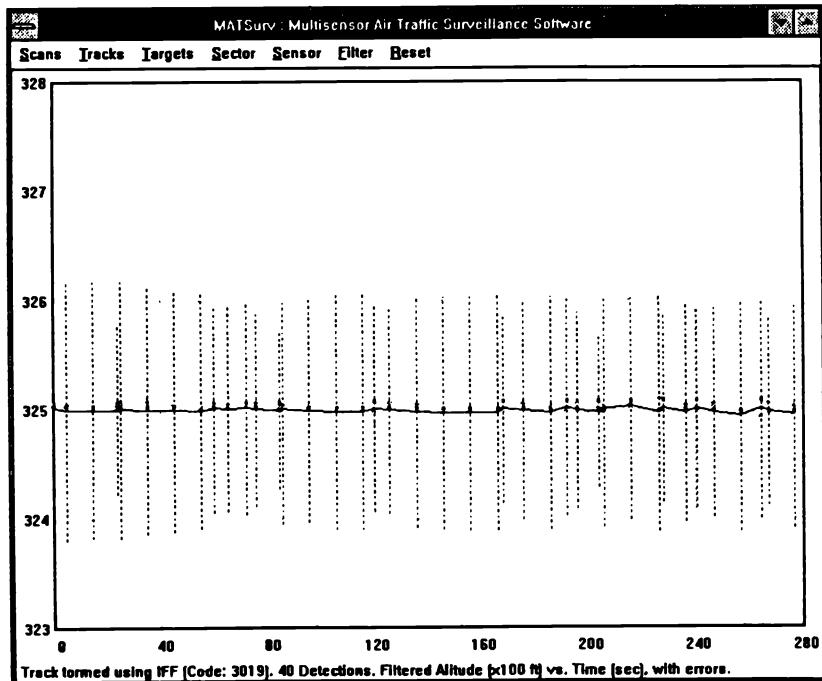


Figure 8.7.6-27: Case 3. Estimated altitude using ID.

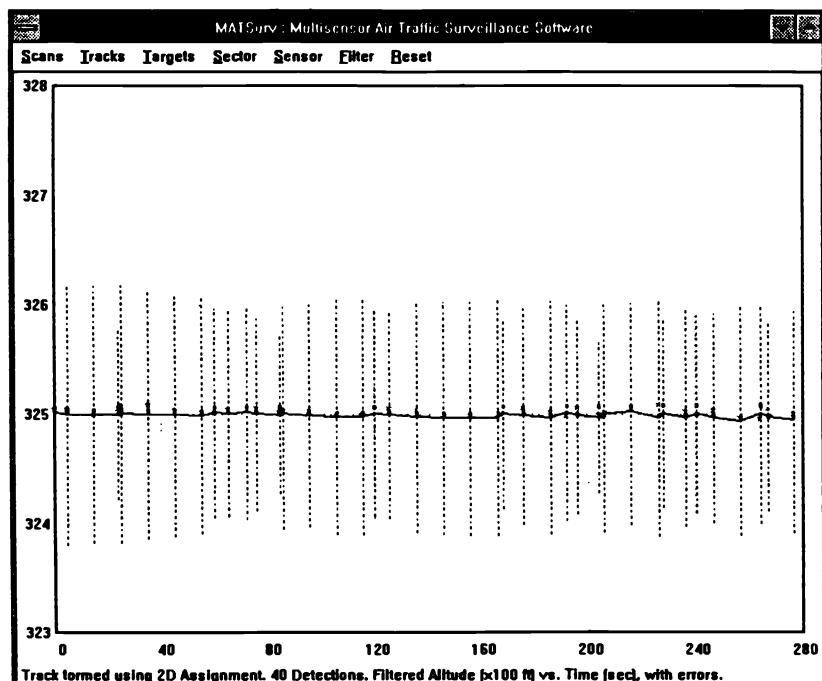


Figure 8.7.6-28: Case 3. Estimated altitude using association.

8.7.6 Real Data Examples

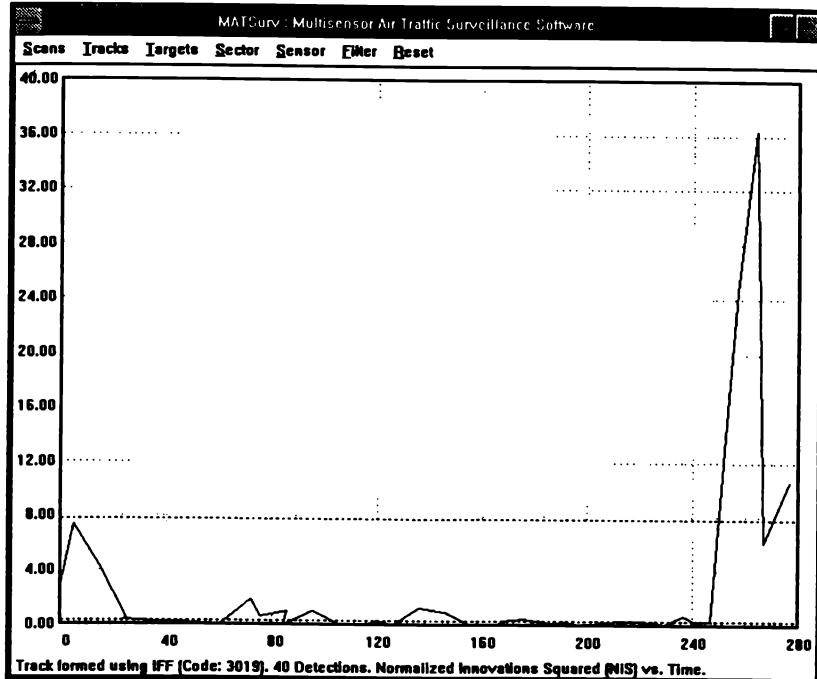


Figure 8.7.6-29: Case 3. Normalized innovation squared using ID (sharp spike due to outlier).

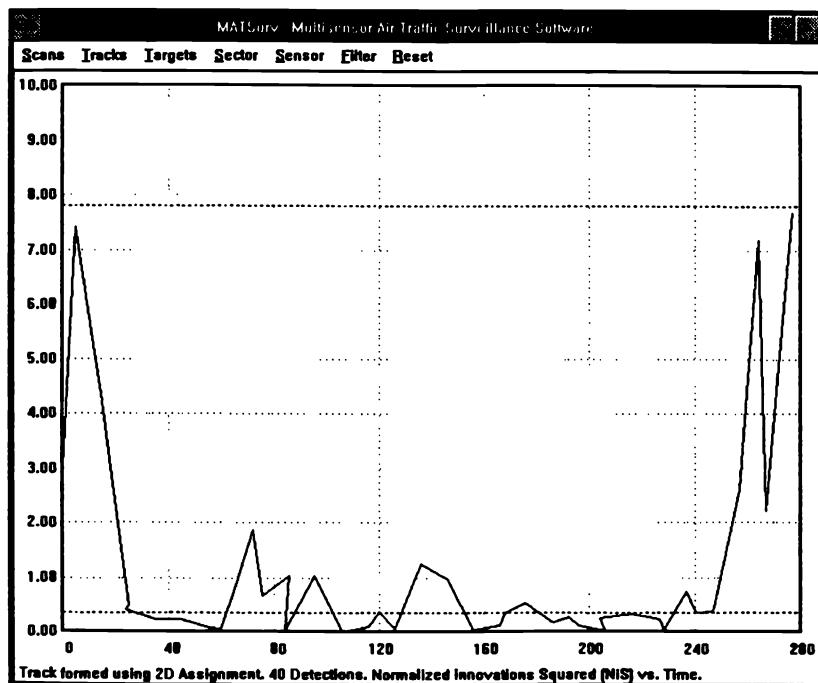


Figure 8.7.6-30: Case 3. Normalized innovation squared using association (within its 95% region).

8.7.6 Real Data Examples

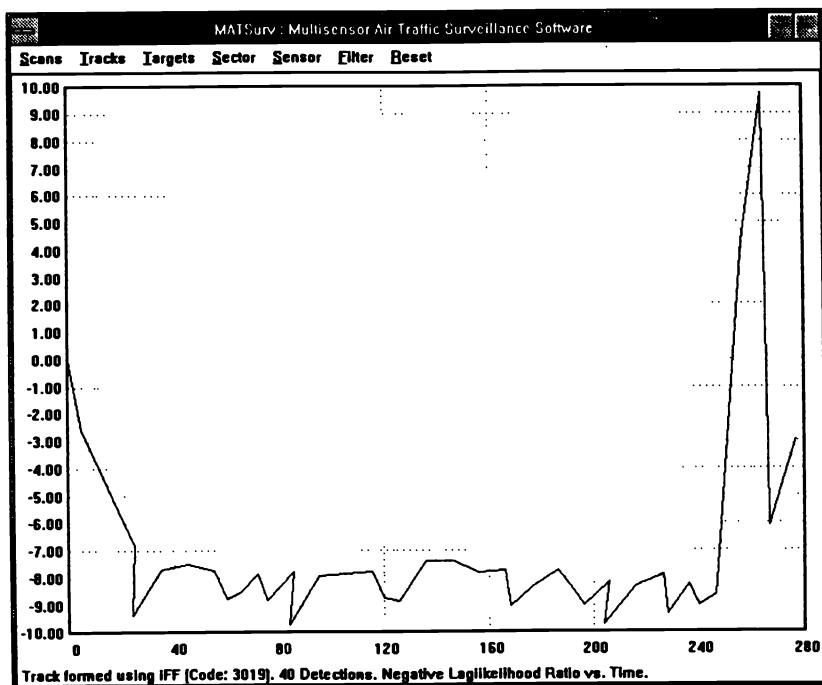


Figure 8.7.6-31: Case 3. Negative log-likelihood ratio using ID (positive for outlier: “extraneous”).

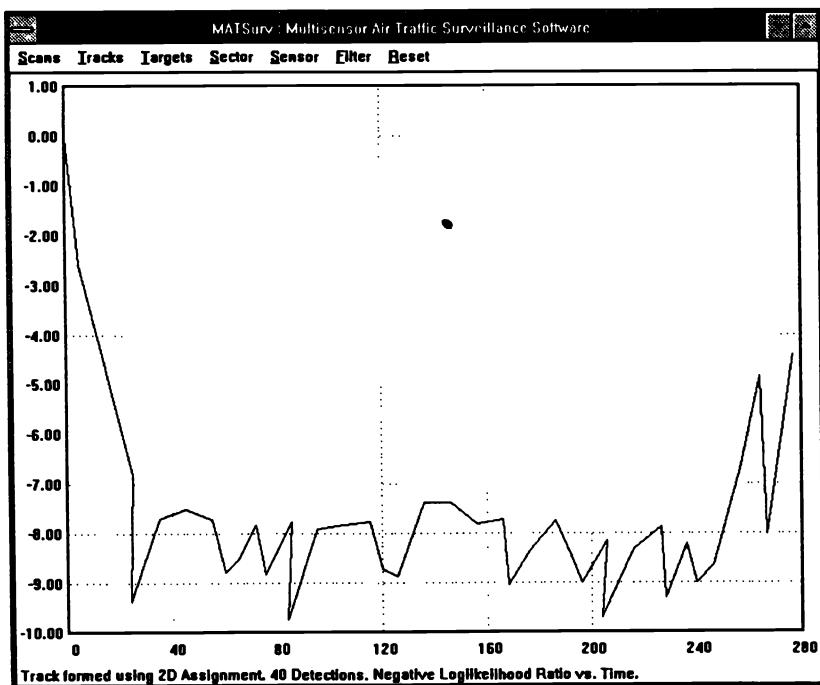


Figure 8.7.6-32: Case 3. Negative log-likelihood ratio using association (consistently below zero).

8.7.7 Multisensor Air Traffic Surveillance — Discussion

The interactive software MATSURV™ — Multisensor Air Traffic SURVeillance — written in C and running under MS-Windows (version 3.0 or higher) carries out association/estimation and displays the results in a graphical format.

First the tracking filter was applied to measurements classified using the ID. These results helped in arriving at the best choice of the design parameters for optimal filter performance. The measurement database from two FAA radars contained detections of targets that were in a variety of trajectories. While many of these target trajectories could be described by a second order linear motion model with a low process noise, there were some maneuvering targets that would require a second order (or third order) motion model with considerable process noise. Since the same tracking filter is required to handle these two extreme cases, the choice of the design parameters has to be necessarily conservative, i.e., tuned to handle the worst case. This required trading off some of the achievable estimation accuracy for an enhanced ability to track targets during maneuver.

The above observation clearly indicates that replacing the Kalman filter with an Interacting Multiple Model estimator would make the above design trade-off unnecessary and will enhance the performance.

In the second phase, the measurements were stripped of their IDs and processed using an assignment algorithm. The results obtained indicate that the association algorithm provides a *superior classification* of the measurements into tracks (i.e., trajectories of the hypothesized targets) than compared to the target trajectories obtained using the measurement IDs.

The multiplicity of targets assigned the same ID prevents the exclusive reliance on the target ID, and its use in evaluating the performance of the association algorithm is clearly inappropriate. A particular track formed by the association algorithm has, in general, a few measurements less than the corresponding target trajectory obtained using the IDs. This is primarily due to the fact that the association algorithm rejects measurements that deviate considerably from the established track, i.e., *outliers*. Discarding these measurements yields a *better estimate* of the trajectory than the one obtained by including these outlying measurements.

8.8 A MULTISENSOR-MULTITARGET DATA ASSOCIATION ALGORITHM FOR HETEROGENEOUS SENSORS

8.8.1 Introduction

An implementation of the ***static intersensor association*** — the first step in the **Type III configuration** (Subsection 8.2.4) — is discussed in this section.

The measurements from the various sensors, assumed to be from the same time — the sensors are assumed *synchronized*) — are to be associated. This yields ***supermeasurements*** that can then be used in (central) *dynamic association and tracking algorithm*.

This will be illustrated on a configuration of 3 sensors which, if passive, e.g., ***electro-optical (EO) sensors***, yield only ***line of sight (LOS)*** measurements — incomplete position observations — that are associated and *full position estimates* (supermeasurements) are obtained. Also the case of heterogeneous — passive and active — sensors is considered.

In the ***multisensor-multitarget position estimation problem*** the key issue is: which measurement originated from which target (data association).

The data association problem is a ***generalized matching problem or assignment problem***, which consists of associating measurements from a number of lists. The number of lists is the number of sensors, N_s .

Such a problem is, for $N_s > 3$, ***NP-hard***: its numerical complexity is not polynomial but exponential/factorial in the size of the problem — the number of elements (measurements) in each list.

8.8.2 Problem Description

8.8.2 Problem Description

The static (synchronized frame) multisensor-multitarget problem — with all observations from the same time — consists of the following:

Find

T = the number of targets

$\hat{\omega}_t$ = position estimate of target t , $t = 1, \dots, T$.

With the measurements assumed to be LOS direction, at least 2 sensors are needed to solve for position. However, **ghosting** — false intersections — occurs with 2 sensors for multiple targets in the same plane, as illustrated in Figure 8.8.2-1. With 2 targets and only 2 sensors there is no way to identify the correct pair.

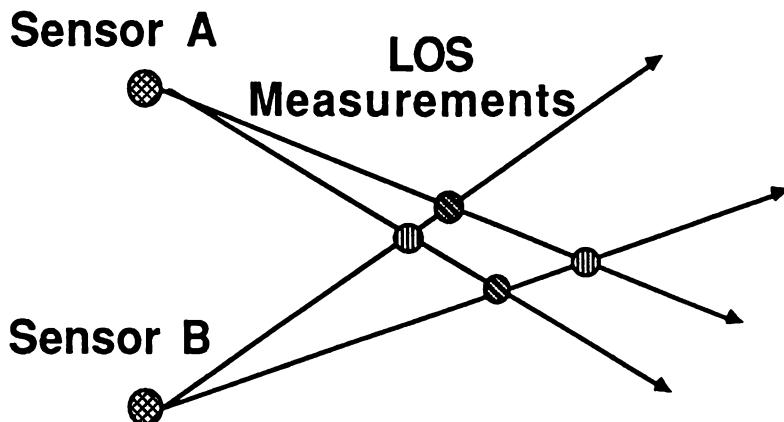


Figure 8.8.2-1: Ghosting with two sensors: two target pairs are possible.

To eliminate or, at least, reduce the ghosting problem, one has to use 3 or more sensors, which makes the problem extremely costly, if not impossible, for a large number of measurements.

8.8.2 Problem Description

Figure 8.8.2-2 shows the situation of 5 targets observed with LOS sensors with noise corrupted measurements. Note that ghosts — false intersections — can show up “better” than correct targets in terms of the goodness of fit.

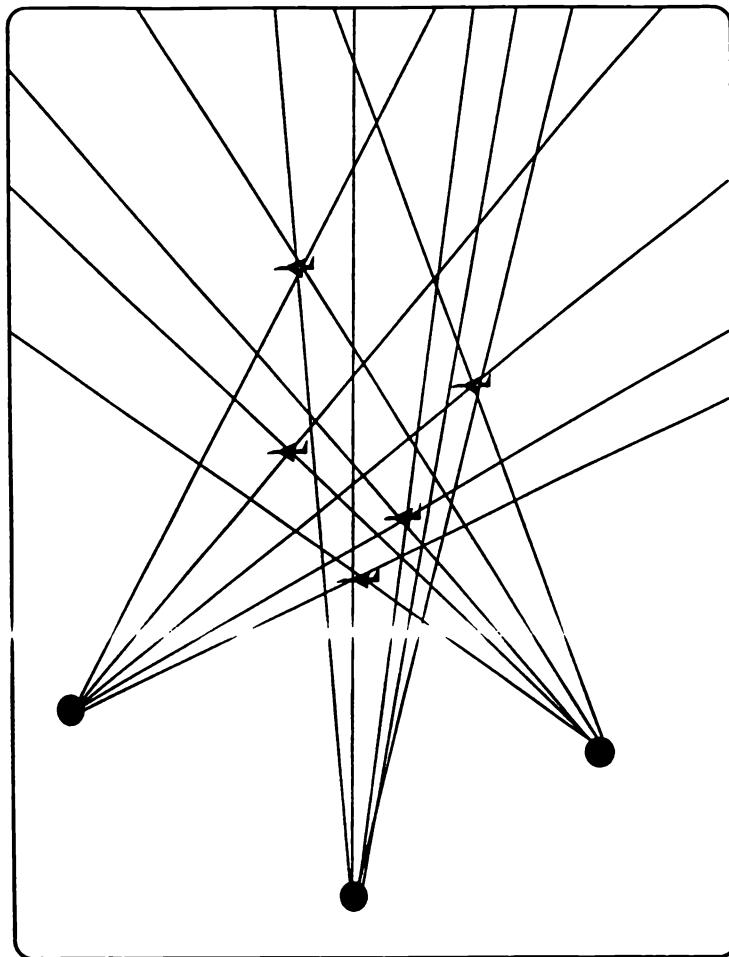


Figure 8.8.2-2: A situation with 3 LOS sensors and 5 targets.

The more realistic situation when there are false alarms and incomplete detections, which is even more complicated, will be considered in the sequel.

8.8.3 Problem Formulation

8.8.3.1 Problem Formulation

An unknown number of targets T are observed by 3 sensors.

There are n_s measurements from sensor s as follows:

- from a passive sensor the LOS direction
 - azimuth θ , elevation ϕ (typical of EO sensors)
 - azimuth θ only (typical of **Electronic Support Measure (ESM)** sensors)
- from an active sensor
 - 2D Radar — range r , azimuth θ
 - 3D Radar — range r , azimuth θ , elevation ϕ .

The target detection and false alarm probabilities for sensor s are $P_D^s \leq 1$, and $P_F^s \geq 0$, respectively.

The unknown target positions and known sensor positions are, respectively,

$$\omega_t = [x_t, y_t, z_t]' \quad (8.8.3-1)$$

$$\omega_s = [x_s, y_s, z_s]' \quad (8.8.3-2)$$

Measurement i_s from sensor s is given by

$$z_{si_s} = \begin{cases} h_s(\omega_t, \omega_s) + w_{si_s} & \text{if the source is target } t \\ v_{si_s} & \text{if spurious} \end{cases} \quad (8.8.3-3)$$

where the measurement noise is

$$w_{si_s} \sim \mathcal{N}(0, R_s) \quad (8.8.3-4)$$

with R_s being the diagonal matrix of the measurement error variances and the pdf of v_{si_s} is

$$p(v) = 1/\Psi_s \quad (8.8.3-5)$$

where Ψ_s is the volume of the field of view of sensor s . Spurious measurements are independent of each other and of the target measurements.

8.8.4 Solution Approach

8.8.4 Solution Approach

To have a complete assignment problem, define “dummy” measurements z_{s0} corresponding to $i_s = 0$.

The goal is to identify triplets of measurements (including dummy ones) that represent

- Targets detected by 3 sensors (no dummy)
- Targets detected by 2 sensors (2 detections and 1 dummy)
- False measurements (1 detection and 2 dummies).

The Likelihood Ratio

Consider a triplet $\{i_1, i_2, i_3\}$. The likelihood that the corresponding measurements originated from target t at the *known* position ω_t is

$$\Lambda(Z_{i_1 i_2 i_3} | \omega_t) = \prod_{s=1}^3 \left[P_D^s p(Z_{s i_s} | \omega_t) \right]^{u(i_s)} [1 - P_D^s]^{1-u(i_s)} \quad (8.8.4-1)$$

where

$$u(i_s) = \begin{cases} 1 & \text{if } i_s \neq 0 \text{ (an actual measurement from sensor } s\text{)} \\ 0 & \text{if } i_s = 0 \text{ (a dummy measurement)} \end{cases} \quad (8.8.4-2)$$

The likelihood that these measurements are spurious (i.e., $t = 0$) is

$$\Lambda(Z_{i_1 i_2 i_3} | t = 0) = \prod_{s=1}^3 \left(\frac{1}{\Psi_s} \right)^{u(i_s)} \quad (8.8.4-3)$$

The *cost* of associating the triplet to target t is given by the negative logarithm of the dimensionless **likelihood ratio**

$$c'_{i_1 i_2 i_3} = -\ln \left(\frac{\Lambda(Z_{i_1 i_2 i_3} | \omega_t)}{\Lambda(Z_{i_1 i_2 i_3} | t = 0)} \right) \quad (8.8.4-4)$$

which is the (negative) **log-likelihood ratio**. Based on the above, a triplet that is more likely to represent a target than be false will have a *negative cost*.

8.8.4 Solution Approach

The Generalized Likelihood Ratio

Since the target position is *not known*, each candidate triplet is used to obtain the ML (actually nonlinear LS) estimate of the target position

$$\hat{\omega}_t = \arg \max_{\omega_t} \Lambda(Z_{i_1 i_2 i_3} | \omega_t) \quad (8.8.4-5)$$

This amounts to triangulation to calculate the full position estimates of a *candidate target*. The candidate triplet has to pass a **coarse gating** test.

The **generalized likelihood ratio (GLR)** approach is to use the estimated position of target t in the negative log-likelihood ratio, written out explicitly as

$$c_{i_1 i_2 i_3} \triangleq \sum_{s=1}^3 [u(i_s) - 1] \ln(1 - P_D^s) + u(i_s) \left[-\ln \frac{P_D^s \Psi_s}{2\pi |R_s|^{1/2}} + [z_{s i_s} - h(\hat{\omega}_t, \omega_s)]' R_s^{-1} [z_{s i_s} - h(\hat{\omega}_t, \omega_s)] \right] \quad (8.8.4-6)$$

Only triplets whose likelihood ratio is greater than 1 are considered for acceptance, i.e., the candidate association $\{i_1, i_2, i_3\}$ is dropped if $c_{i_1 i_2 i_3} > 0$. This is called **fine gating**.

Feasibility Requirements

To identify a target amounts to identifying the measurements that originated from it. The most likely solution is the one that minimizes the negative log-likelihood ratio over all possible associations.

A **feasible solution** is a **partition** of the measurement set

$$\gamma = \left\{ \{ \text{measurement triplets denoting targets} \} \bigcup \{ \text{measurements deemed false} \} \right\} \quad (8.8.4-7)$$

such that each measurement belongs to a target or is a false measurement and each measurement can be used *only once* — unresolved (multisource) measurements are not modeled explicitly in this formulation.

8.8.4 Solution Approach

From Feasibility Requirements to a Matching Problem

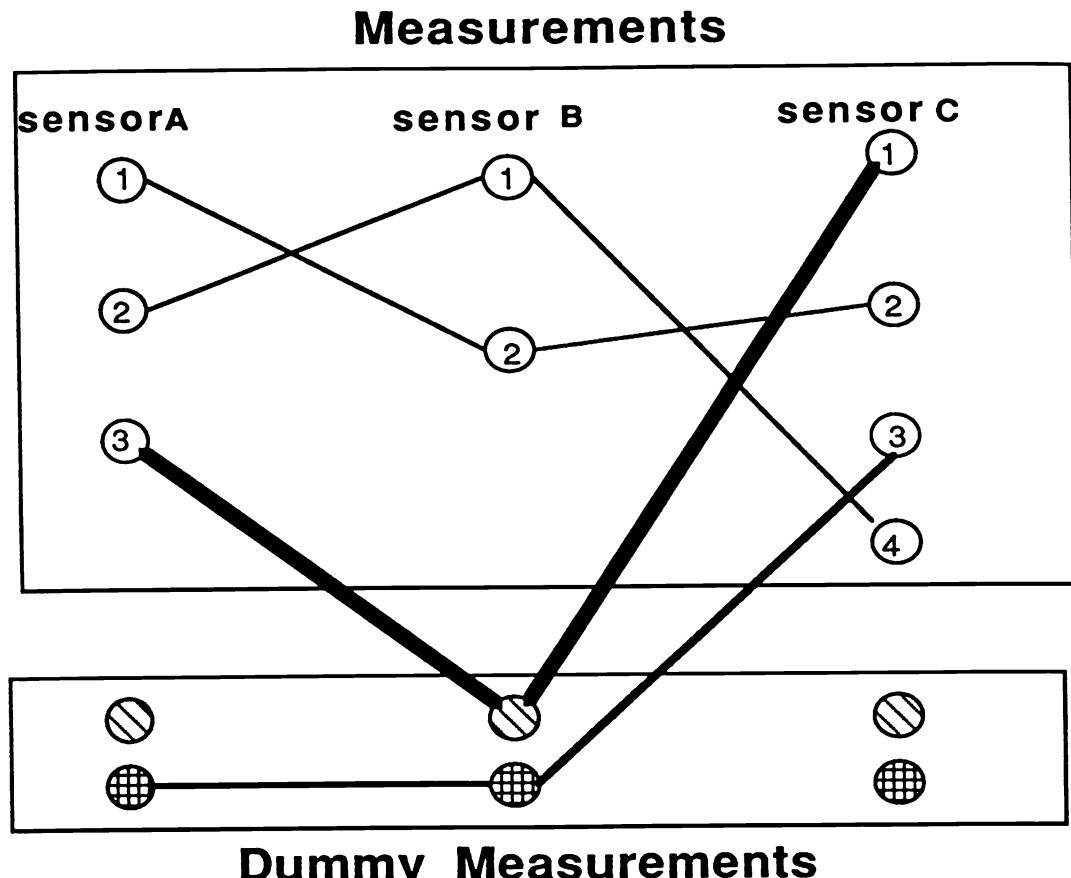


Figure 8.8.4-1: The use of dummy measurements in the matching problem.

The dummy measurements have a dual role:

- A dummy measurement can be used as a “wild-card” measurement to compensate for missed detection by one of the sensors to have a **complete assignment algorithm**.
- Since detection by at least 2 sensors is required to form position estimates, single sensor detections and false alarms are not distinguishable — thus any measurement in a triplet with 2 dummies is *declared as a false alarm*.

8.8.4 Solution Approach

Generalized 3-D Matching Problem

The minimization of the negative log-likelihood ratio can be recast as the following problem (n_s denotes the number of measurements at sensor s):

$$\min_{\rho_{i_1 i_2 i_3}} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} c_{i_1 i_2 i_3} \rho_{i_1 i_2 i_3} \quad (8.8.4-8)$$

subject to the feasibility constraints on the binary variables ρ

$$\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \rho_{i_1 i_2 i_3} = 1 \quad i_3 = 1, 2, \dots, n_3 \quad (8.8.4-9)$$

$$\sum_{i_3=0}^{n_3} \sum_{i_1=0}^{n_1} \rho_{i_1 i_2 i_3} = 1 \quad i_2 = 1, 2, \dots, n_2 \quad (8.8.4-10)$$

$$\sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} \rho_{i_1 i_2 i_3} = 1 \quad i_1 = 1, 2, \dots, n_1 \quad (8.8.4-11)$$

where

$$\rho_{i_1 i_2 i_3} \triangleq \begin{cases} 1 & \text{if measurements } i_1, i_2 \text{ and } i_3 \text{ are assigned to a target} \\ 0 & \text{otherwise} \end{cases} \quad (8.8.4-12)$$

This is a **generalized 3-D assignment problem**. The **dimension of the assignment problem** is the number of sensors or **lists of measurements**.

The 3-D assignment problem — the above problem even without missed detections and false alarms — is NP-hard.

NP-hardness

The 2-D assignment problem can be solved in $O(n^3)$ time (**order of n^3**) where n is the number of measurements — **the size of the problem**.

The 3-D assignment problem is **NP-hard**: its worst case complexity is **non-polynomial** — cannot be bounded by a polynomial of fixed order — it is factorial-exponential, as for the **traveling salesman's problem**.

8.8.4 Solution Approach

Severity of NP-hardness

Assume a particular algorithm requires $1 \mu\text{s}$ to solve a two target ($n = 2$) scenario. Table 8.8.4-1 shows the time it will require to solve the problem for different size problems for factorial, exponential and cubic complexities.

	Complexity	$n!$	3^n	n^3
Problem size				
5 targets		$60 \mu\text{s}$	$27 \mu\text{s}$	$16 \mu\text{s}$
10 targets		1.8s	6.6ms	$125 \mu\text{s}$
20 targets		38.5 centuries	370s	1ms

Table 8.8.4-1: Complexity comparison.

Figure 8.8.4-2 sketches the optimal vs. a suboptimal solution.

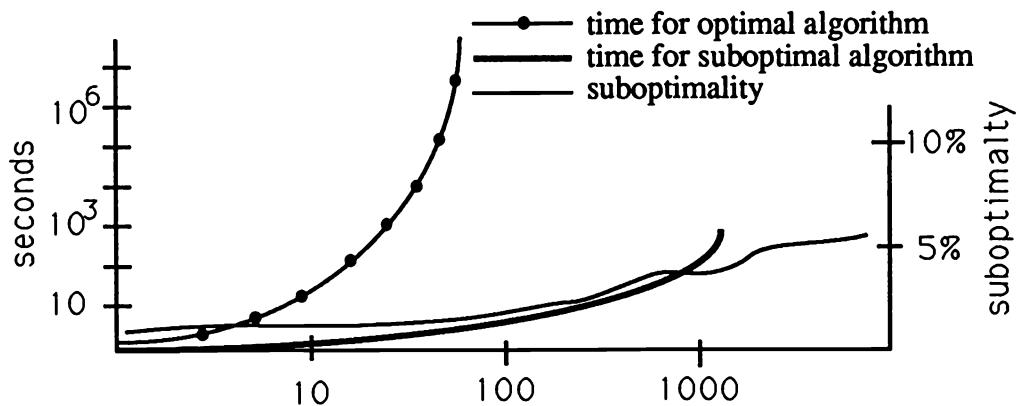


Figure 8.8.4-2: Optimal vs. suboptimal algorithm.

The solution approach is: *trade off some accuracy for a lot of computation.*

8.8.4 Solution Approach

Benefits of a Nearly Optimal Algorithm

For **sparse scenarios** that involve few spatially well separated targets almost anything will work. In this case most targets can be placed in unique planes and resolved from two sensor measurements.

For n targets in a plane and 2 sensors, there will be $n^2 - n$ ghosts.

For **dense clusters**, multiple targets can occur in a plane and the resultant **graph** (possible associations) is dense. The resulting problem has a large number of feasible solutions — factorial/exponential in n .

Search based algorithms (like **set packing** or **branch and bound**) may take too long for graphs with density of 10% or more.

A naive suboptimal algorithm can easily pick a wrong solution.

Suboptimality vs. Association Accuracy

Figure 8.8.4-3 shows the effect of suboptimality on the association accuracy.

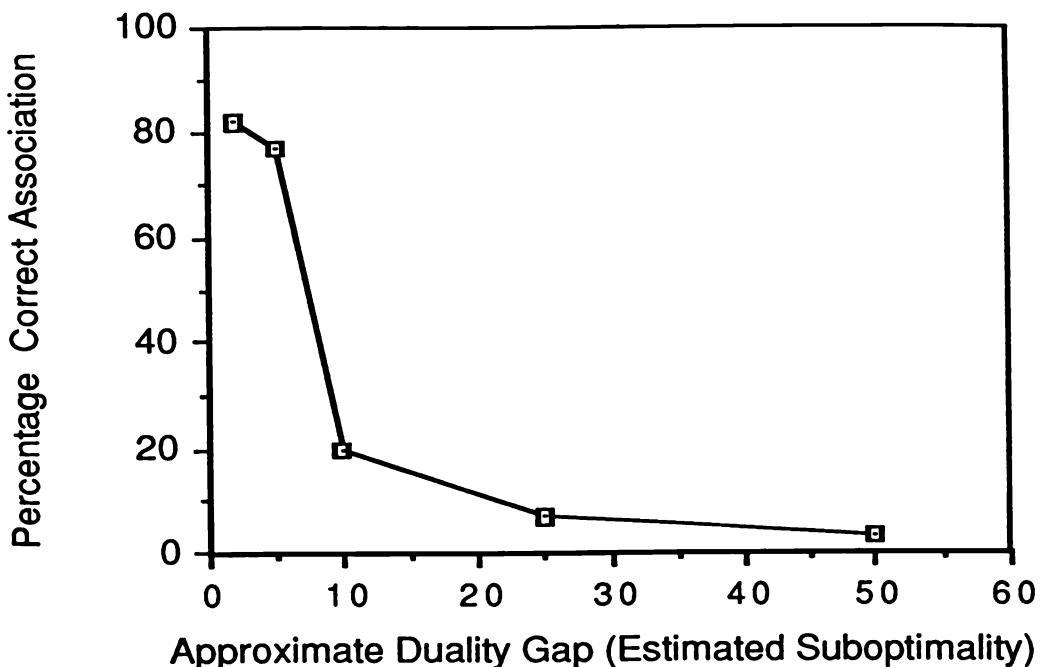


Figure 8.8.4-3: Correct associations vs. cost suboptimality.

8.8.4 Solution Approach

Desirable Features in a Nearly Optimal Algorithm

- Solutions that are within 5% in cost from the optimal solution are needed
- A measure of the accuracy of the solution should be available
- The algorithm should be iterative so it can improve the solution if time is available
- Intermediate solutions should be feasible
- Polynomial storage and time complexity
- Parallelizability.

The relaxation algorithm to be presented has all of the above desirable features and has a complexity of (the *order of*) $O(3kn^3)$ where k is the number of iterations and n the number of the measurements (list size) per sensor.

8.8.5 The Relaxation Algorithm

The Primal-Dual Approach

The approach to carry out the search for a near-optimal assignment according to the generalized likelihood ratio cost (8.8.4-8) consists of the following **relaxation algorithm**, also called the **Primal-Dual approach**:

- 1) Relax one constraint, include it in the cost function with Lagrangian multipliers and solve a simpler problem — the **dual problem**
- 2) Reintroduce the constraint and modify the solution — **primal problem**
- 3) Update the Lagrangian multipliers
- 4) Repeat steps 1, 2 and 3 until a convergence criterion is satisfied.

Remarks

The dual problem

- is a 2-D assignment problem (it enforces only 2 out of the 3 sets of constraints)
- is optimally solvable in polynomial time (e.g., with the **Auction algorithm**)
- provides a *lower bound* on the optimal cost since one of the constraints is relaxed.

The **primal feasible solution** is obtained from the dual by enforcing the constraint that was relaxed via another 2-D assignment problem and

- it yields an *upper bound* on the optimal cost.

The evolution of the cost upper and lower bounds, the difference of which is the approximate **duality gap**, with the iterations is illustrated in Figure 8.8.5-1. This gap gives a pessimistic measure of optimality of the solution.

Properties of the Relaxation Algorithm

- At every iteration there is a feasible solution
- The dual solution satisfies 2 of the 3 constraints of the feasible solution. Therefore, all feasible solutions are also dual solutions
- The minimum dual solution has cost less than or equal to that of the minimum feasible solution
- If the dual solution is a feasible solution (zero gap) then it is *the optimal solution to the 3-D matching problem*.

8.8.5 The Relaxation Algorithm

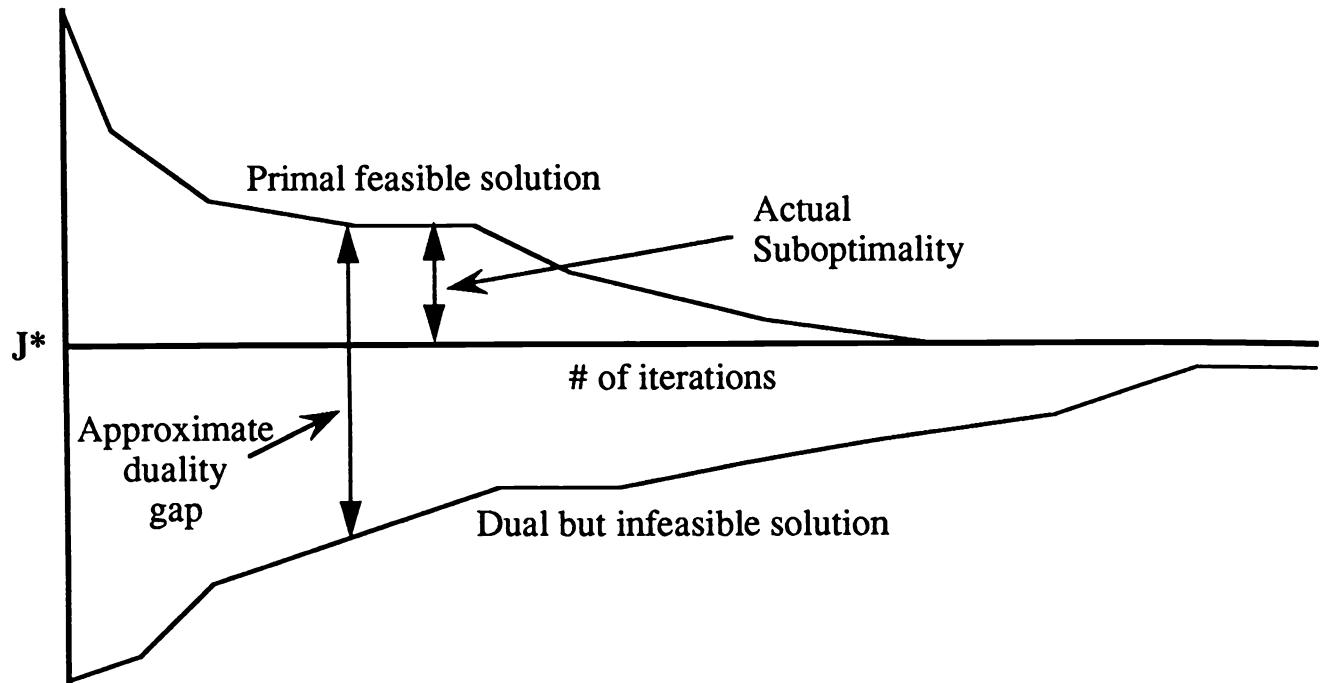


Figure 8.8.5-1: The evolution of the duality gap in the relaxation algorithm.

The Steps of the Relaxation Algorithm

- Form dual 2-D subproblems with costs

$$d_{i_1 i_2} = \min_{i_3} (c_{i_1 i_2 i_3} - u_{i_3}) \quad (8.8.5-1)$$

where u_{i_3} are the Lagrangian multipliers for $i_3 = 1, 2, \dots, n_3$

- Solve the 2-D subproblem using the Auction 2-D assignment algorithm
- Find a feasible solution by retaining the dual 2-D assignments and enforcing the third constraint
- Update the dual variables with the Modified Subgradient from the Auction algorithm
- Repeat the above steps to improve the solution.

The difference between dual and feasible solutions is the approximate duality gap — a conservative measure of suboptimality.

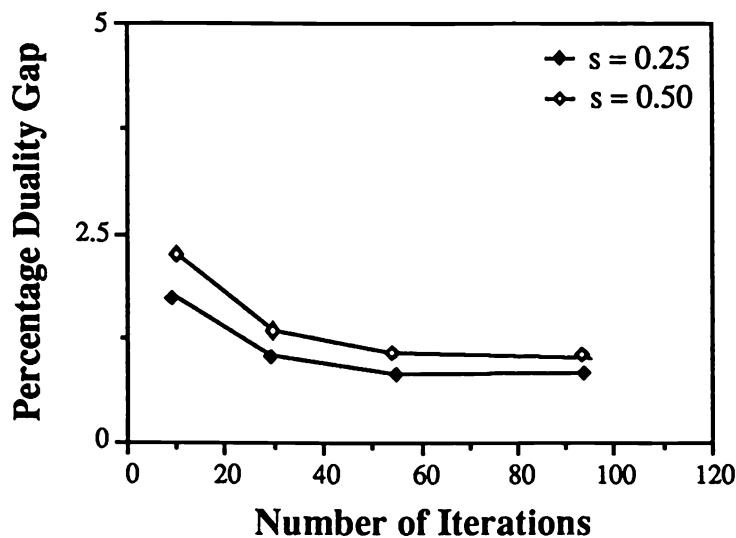


Figure 8.8.5-2: Average duality gap for $N = 10$ nodes (measurements) (sparsity of the graph: $s \triangleq$ total number of candidate associations/ N^3).

Comparison with Optimal Algorithm

Number of nodes	Algorithm	Sparsity	0.05	0.1	0.25	0.5	1.0
5	BB		1.2	1.6	1.7	2.2	16.6
	RA		0.006	0.01	0.04	0.2	0.4
10	BB		3.0	3.8	30.3	3844.0	*
	RA		0.02	0.06	0.6	1.4	2.3
15	BB		4.8	26.4	1030.4	*	*
	RA		0.16	0.27	2.1	3.6	5.8
20	BB		18.5	656.1	*	*	*
	RA		0.2	0.44	2.3	7.5	12.1

Table 8.8.5-1: Speed of optimal (Branch-and-Bound: BB) vs. relaxation algorithm (RA) (the average time in seconds on a Sun 386i workstation; * memory/computational requirements exceeded 12 MB RAM / 0.64 MFLOPS).

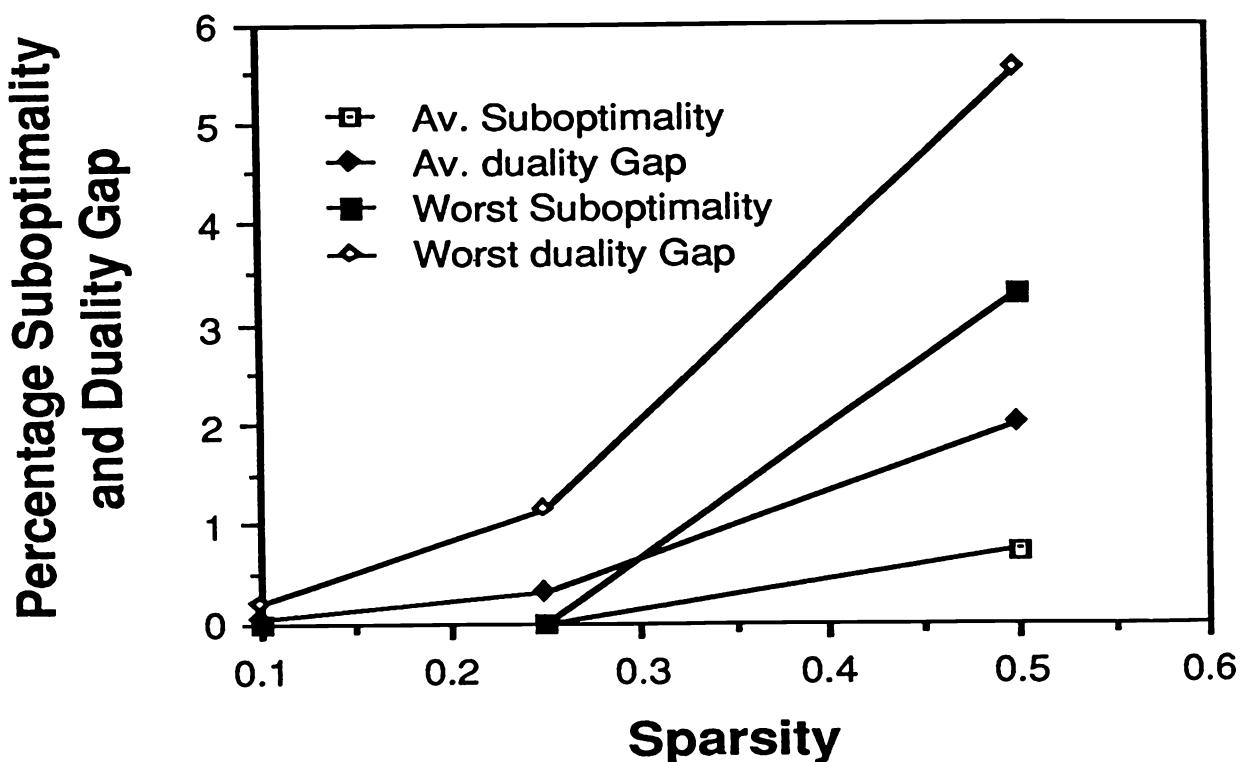


Figure 8.8.5-3: Plot of suboptimality and duality gaps of the relaxation algorithm after 25 iterations (number of nodes = 10).

8.8.5 The Relaxation Algorithm

Overview of the Data Association Algorithm

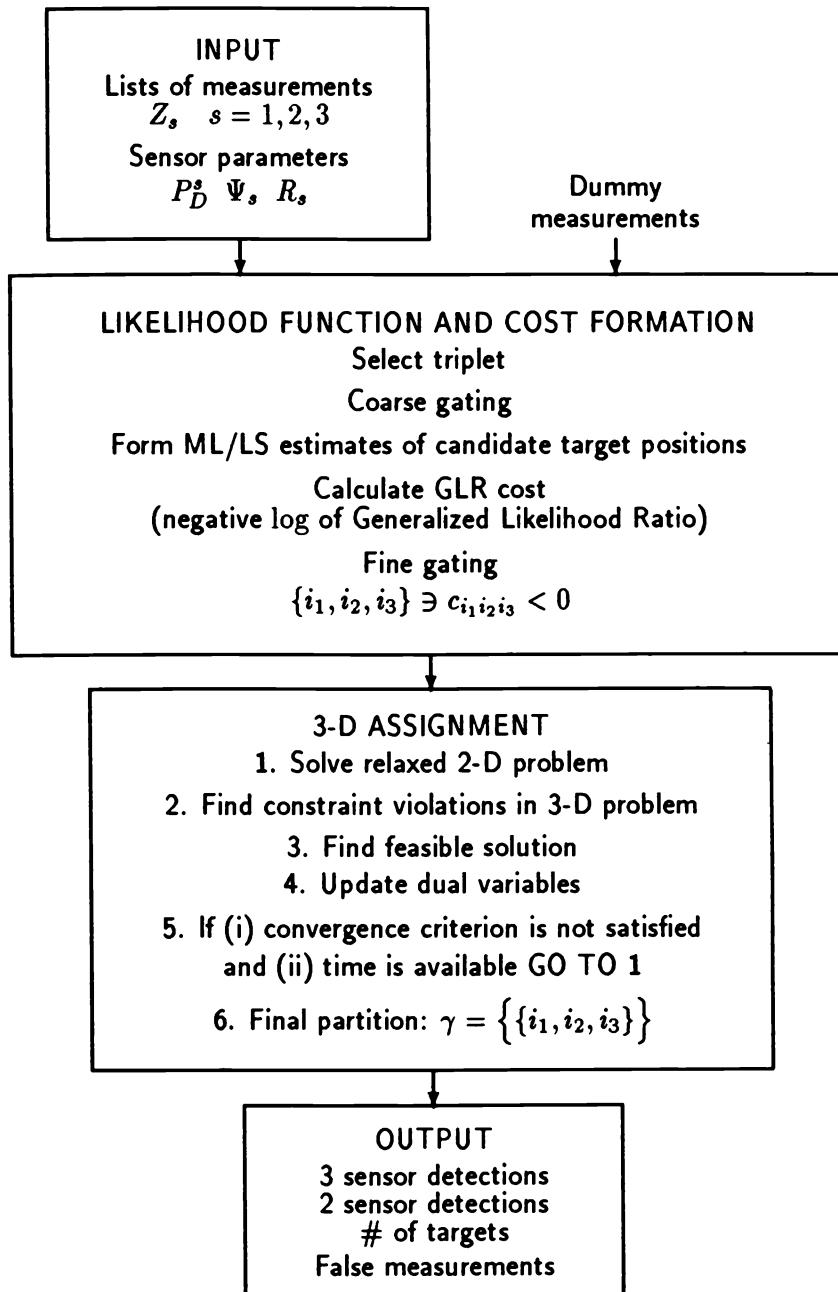


Figure 8.8.5-4: Flowchart of the assignment algorithm for data association (implemented in **PASSDAT™**)

8.8.6 Simulation Results

8.8.6.1 Simulation Results

Scenario Description

The scenario, illustrated in Figure 8.8.6-1, consists of 64 targets in a less than 2° field of view (FOV), observed by 3 sensors, and is characterized by multiple targets per ghosting plane.

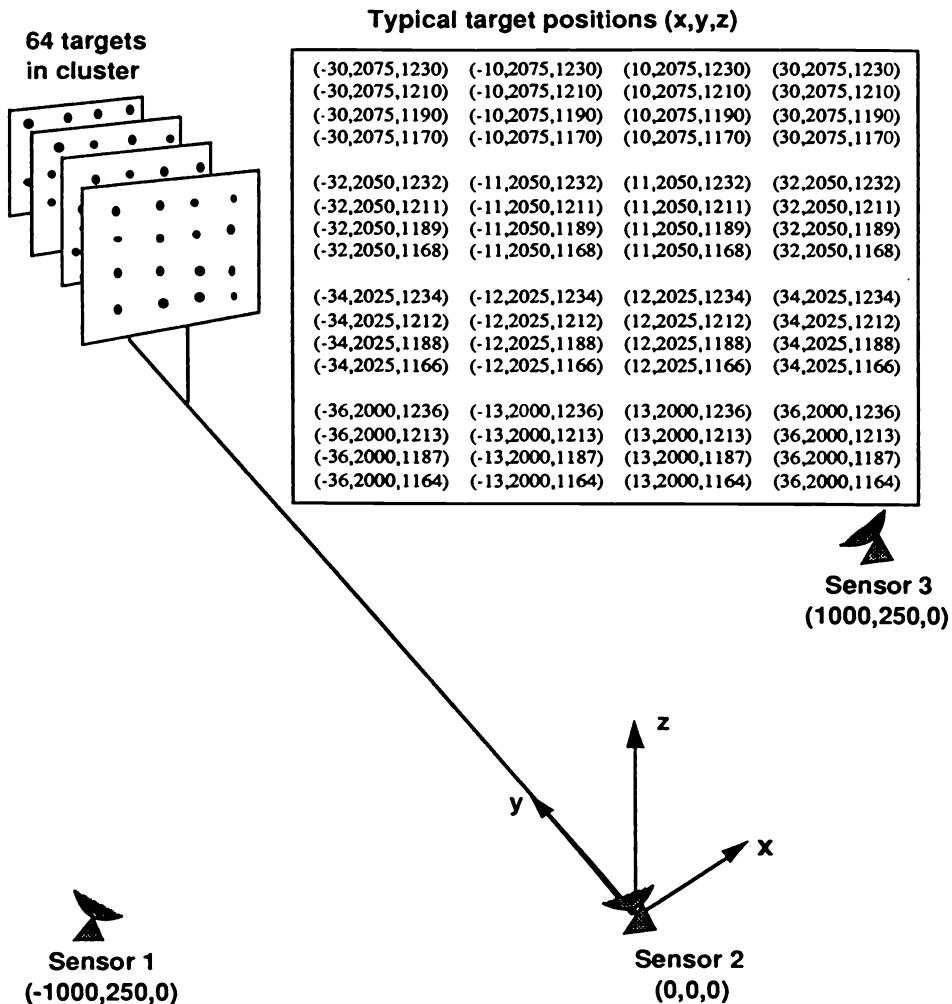


Figure 8.8.6-1: The Scenario.

8.8.6 Simulation Results

Sensor Attributes

	Sensor 1	Sensor 2	Sensor 3
Sensor type	Passive	Passive 3-D radar	Passive
Position (x, y, z)	(-1000, 250, 0)	(0, 0, 0)	(1000, 250, 0)
Detection probability	0.95	0.95	0.95
False alarm rate per resolution cell	10^{-5}	10^{-5} (Passive) 10^{-8} (Radar)	10^{-5}
Azimuth field of view	$58.2^\circ - 63.2^\circ$	$87.5^\circ - 92.5^\circ$	$116.8^\circ - 121.8^\circ$
Elevation field of view	$28^\circ - 33^\circ$	$28^\circ - 33^\circ$	$28^\circ - 33^\circ$
Range field of view	Not applicable	2275–2425 km (Radar only)	Not applicable

Table 8.8.6-1: Sensor characteristics.

Three sensor resolution cases were considered: 500, 1000 and 2000 cells in azimuth and elevation each.

The standard deviation of the angle measurement noise was taken as 1/5 cell in each case, yielding accuracies of 0.002° , 0.001° and 0.0005° , respectively. The range measurement noise had 50m standard deviation.

Not all targets are resolved: there were about 2 unresolved targets per sensor in the 500 resolution cells case.

8.8.6 Simulation Results

The Results

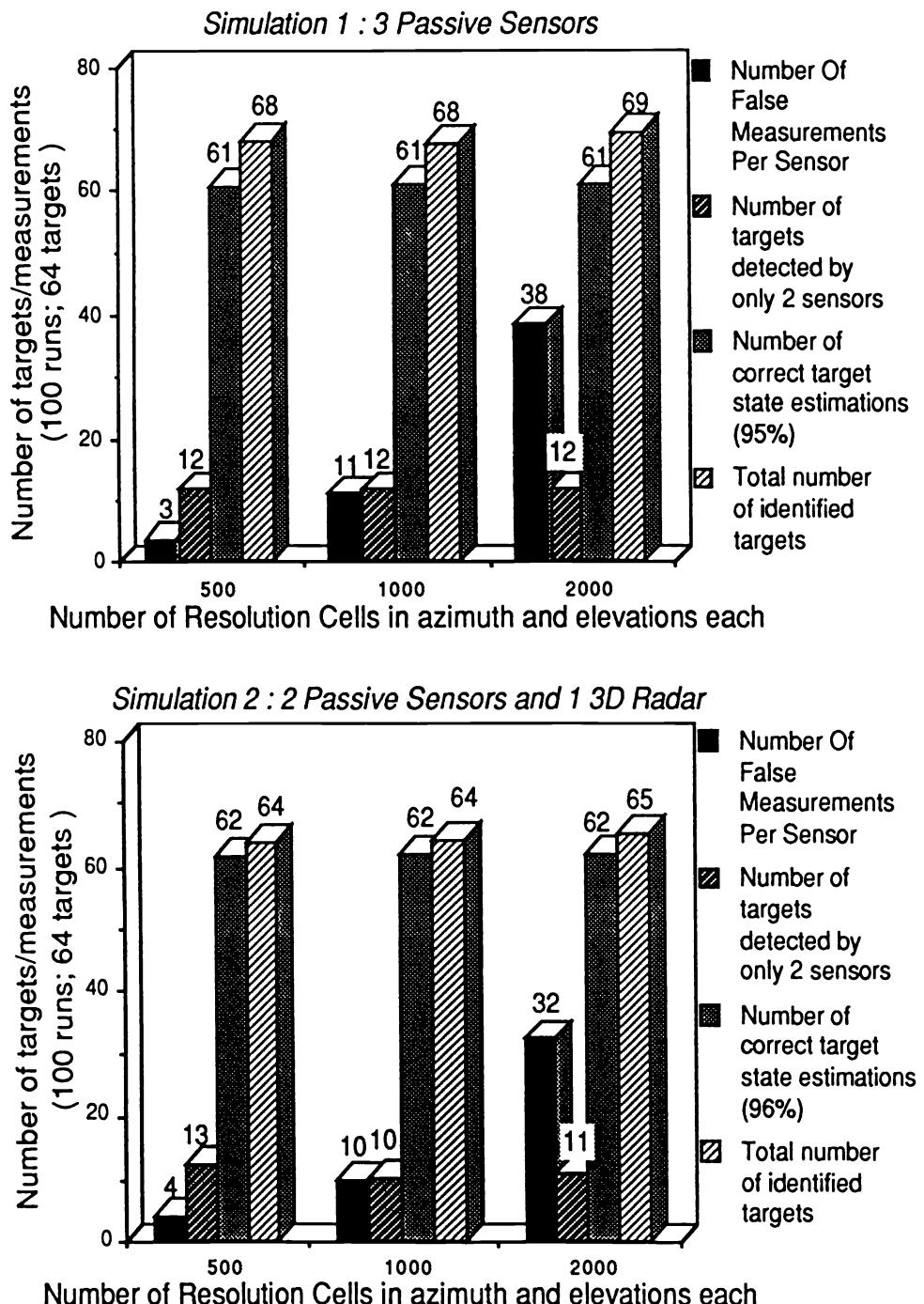


Figure 8.8.6-2: Results for two 3-sensor configurations.

8.8.6 Simulation Results

	Case 1 500 cells	Case 2 1000 cells	Case 3 2000 cells
Total number of false alarms	9.21	31.5	114
Number of targets detected by 2 or more sensors	63.13	63.02	62.74
Number of targets detected by only 2 sensors	11.58	11.63	11.77
Number after coarse gating	240.39	220.8	214.8
Number after fine gating	240.35	220.8	214.8
Average CPU time (in seconds on a SUN 386i/250)	19.5	19	19.3
Number of identified targets	68.08	67.58	69.26
Percent correct association	94.8	95.13	94.95
Average error in position estimate (km)	0.147	0.074	0.037

Table 8.8.6-2: Details of simulation with 3 passive sensors (100 runs).

	Case 1 500 cells	Case 2 1000 cells	Case 3 2000 cells
Total number of false alarms	11.75	29.24	97.3
Number of targets detected by 2 or more sensors	63.2	63.42	63.3
Number of targets detected by only 2 sensors	12.45	10.38	10.86
Number after coarse gating	239.61	224.68	215.84
Number after fine gating	208.54	206.99	202.6
Average CPU time (in seconds on a SUN 386i/250)	15.88	16.44	16.02
Number of identified targets	63.87	64.06	65.18
Percent correct association	96.31	96.94	96.94
Average error in position estimate (km)	0.085	0.052	0.031

Table 8.8.6-3: Details of simulation with heterogeneous sensors (100 runs; sensors 1 and 3 passive, sensor 2 a 3-D radar with $\sigma_r = 50\text{m}$).

8.8.6 Simulation Results

In the high resolution case the number of false alarms was around 50% of the number of targets; this resulted in about 7 spurious targets for configuration 1 and only 2 for configuration 2. The association accuracy was about 95%.

The candidate target position calculation (via nonlinear LS, using a gradient technique) takes up about 90% of the computations.

Suboptimality and the Association Accuracy

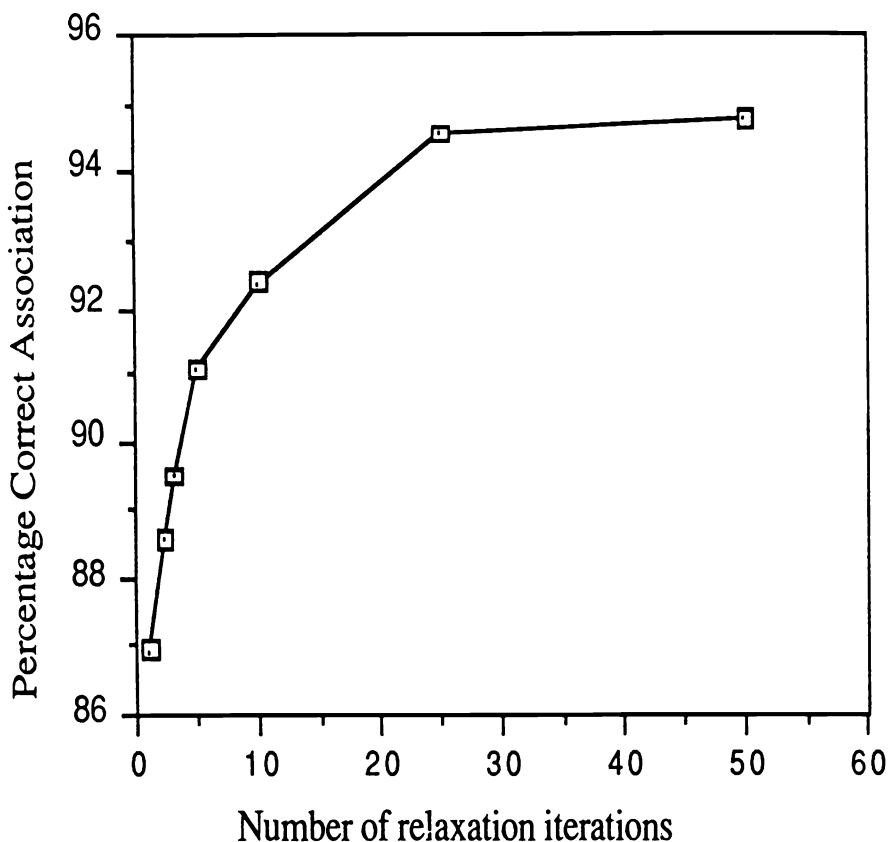


Figure 8.8.6-3: Correct association vs. number of iterations.

An 8% improvement in the accuracy is obtained in 25 iterations of the relaxation algorithm. The solution is typically within 2% in cost of the (perhaps unknowable) optimal solution after 25 iterations.

8.8.7 Association for Synchronized Sensors — Summary

The relaxation-based association algorithm for *synchronized frames of multiple sensors* has a robust performance:

- uniformly good association accuracy ($\approx 95\%$) even in the presence of heavy clutter
- about 50% spurious measurements in a 64 target scenario yielded 10% extra targets.

The approach, based on the Generalized Likelihood Ratio — dimensionless costs — enables optimization over sets of 2 sensor and 3 sensor detections:

- in the example considered, 20% of the targets were detected by only 2 sensors.

The algorithm can associate data from heterogeneous sensors.

The algorithm is parallelizable — this is important since the GLR calculations take up 90% of the time. The GLR calculations are *completely* parallelizable — one can obtain *linear speedup in the number of processors*.

This problem is mathematically equivalent to the *multiple scan (frame) association problem based on single sensor data*.

8.9 MEASUREMENT ASSOCIATION FROM DIRECTION FINDERS

8.9.1 The S-D Assignment via Lagrangian Relaxation

The S -D problem is solved as a series of relaxed 2-D subproblems. The constraints $r = S, S - 1, \dots, 3$ are successively relaxed and appended to the cost with Lagrange multipliers. At each stage the best choice for list r is computed for all combinations from the previous lists. At stage $r = 3$ one has a 2-D problem, which can be optimally solved using the **generalized Auction algorithm** [PDBW92] in polynomial (cubic) time.

The constraints are then reimposed one at a time ($r = 3, \dots, S$), and the corresponding Lagrange multipliers are updated; at each stage the resulting feasible (but most likely suboptimal) solution is computed, until all constraints are met. The duality gap — the difference between the cost from the maximally relaxed problem and the fully constrained one — is calculated and the iterations continue until this gap is small enough.

This is illustrated schematically in Figure 8.9.1-1.

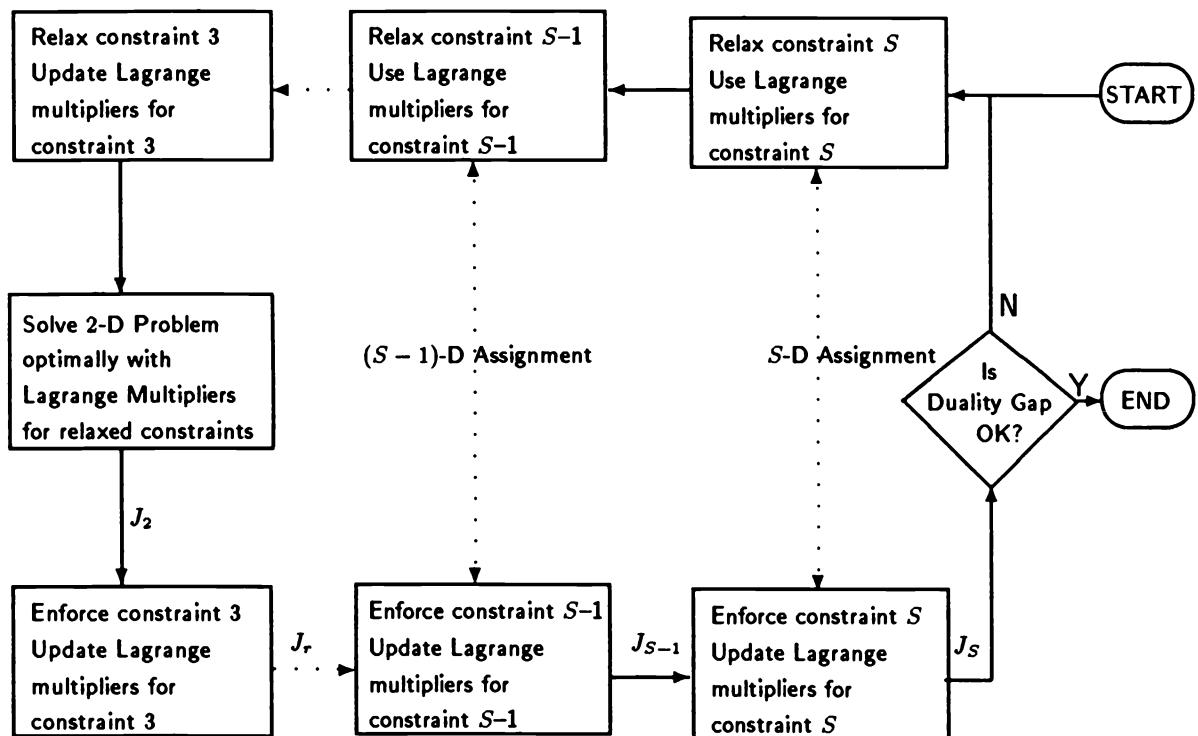


Figure 8.9.1-1: Outline of the S -D assignment via Lagrangian relaxation.

8.9.2 Example of Localization of Emitters

8.9.2 Example of Localization of Emitters

The following problem, illustrated in Fig. 8.9.2-1 is considered. There are N_S sensors at known fixed locations in a plane arranged in a semicircle of radius 1000 km centered at the origin of the coordinate system. Each sensor is a **direction finding ionosonde**, i.e., a multielement passive interferometer which estimates the azimuth of an RF emission, for example a radio message from an aircraft. These sensors have extremely high range and can “see” beyond the curvature of the earth, because they receive the signals bouncing off the ionosphere. However, because of instabilities in the ionosphere, the RMS accuracy of these measurements varies widely: 0.3° to 2° .

Azimuth measurement error standard deviations $\sigma_\theta = 0.5^\circ, 1.0^\circ$ and 2.0° were used. The targets are sufficiently separated so that there are no unresolved targets. The sensors are assumed to be forward looking with a field of view of 180° , with detection probability of $P_D = 0.9$. The false alarm rate of the sensors was assumed 0.8/radian. With $N_t = 5$ targets, the average number of detections per scan at each sensor is therefore 7 (with $N_t P_D = 4.5$ true detections and $0.8\pi = 2.5$ spurious detections).

The targets were assumed on the $y = 500$ line, with an intertarget separation of 400km. When $\sigma_\theta = 2^\circ$, the intertarget separation, as seen by the middle sensor, is $7.5\sigma_\theta$.

This scenario results in huge number of candidate associations. Any two LOS measurements intersect at a point in a plane, implying a target at that position could have produced these two measurements. In the 7 sensor scenario, the possible number of sets of two LOS intersections is approximately 1 million. To reduce the number of candidate associations, **only detections by the majority of the sensors** were considered in the association process. Thus, in the 7 sensor scenario, a candidate association must include at least 4 non-dummy measurements.⁵

The results for this scenario are presented in Tables 8.9.2-1 and 8.9.2-2. While the position estimation error can be reduced by increasing the number of sensors, the computational requirements explode with the number of sensors — the number of candidate associations increases from approximately 500 for the 3 sensor scenario to up to 100,000 for the 7 sensor scenario.

The problem becomes rather easy if the azimuth error is 1.0° or 0.5° , because the association problem becomes sparse, with only about 1000 candidate associations.

⁵With $P_D = 0.9$, a target is detected by 4 or more sensors in a 7 sensor scenario with a probability of 0.997. If there are only 3 sensors, a target is detected by 2 or more sensors with probability 0.97. Therefore, this requirement does not lead to a significant loss of performance.

8.9.2 Example of Localization of Emitters

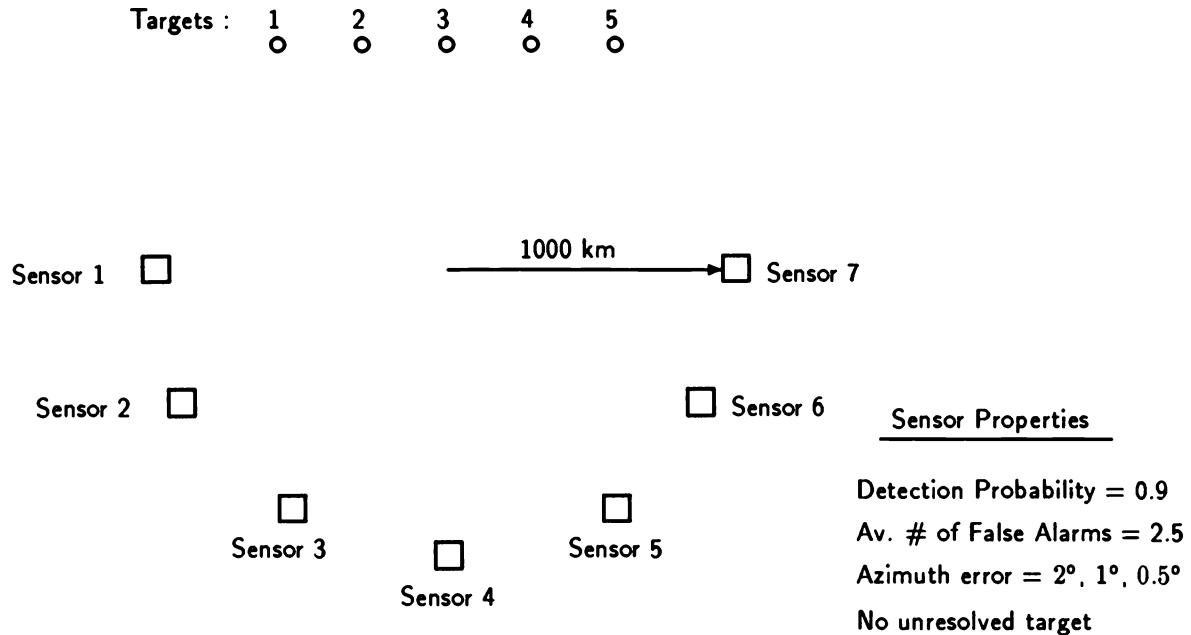


Figure 8.9.2-1: Scenario for the 7 sensor 5 target direction finding example.

	S = 3	S = 5	S = 7
Association accuracy (Fraction of true sources identified correctly)	48%	69%	74%
Average number of false targets (ghosts)	6.2	5.1	3.5
Position RMS error (km)	58.9	37.5	24

Table 8.9.2-1: Simulation results for high frequency direction finding scenario averaged over 100 runs for 3, 5 and 7 sensors with $\sigma_\theta = 2.0^\circ$.

	$\sigma_\theta = 2.0^\circ$	$\sigma_\theta = 1.0^\circ$	$\sigma_\theta = 0.5^\circ$
Association accuracy (Fraction of true sources identified correctly)	74%	92%	94%
Average number of false targets (ghosts)	3.5	2.1	1.9
Position RMS error (km)	24	9.9	5.9

Table 8.9.2-2: Simulation results for high frequency direction finding scenario averaged over 100 runs with $S = 7$ sensors for various σ_θ .

8.9.3 Emitter Localization — Summary

The problem of emitter localization using S direction finders is, for $S > 3$, NP-hard. For $S = 7$ and 5 targets it becomes infeasible for all but a sophisticated assignment algorithm which has polynomial complexity.

An S -D assignment algorithm based on

- successive Lagrangian relaxation
- modified Auction to solve optimally 2-D assignment problem
- iteration until the duality gap is small enough

was described and applied successfully to this problem.

Such an algorithm can be used on the data from a time window, during which the targets can be assumed not to move — a static association problem — the results of which are “supermeasurements”.

Following this, a dynamic state estimation can be carried out, keeping in mind that there will be unavoidable false supermeasurements and the target detection probability in a window is less than unity.

8.10 NOTES AND PROBLEMS

8.10.1 Bibliographical Notes

The classification of the multisensor information processing configurations presented in Section 8.2 is based mostly on [Dru90].

The issue of *track-to-track association* (or “correlation”) was first dealt in [SS71] assuming they have independent estimation errors as discussed in Section 8.3. This was corrected for the fact that the errors are correlated in [Bar81, BC86], which forms the basis of the more detailed discussion from Section 8.4. The issue of optimality of the fusion was brought up in [RM88, Bar88].

More details on the material of Section 8.5 can be found in Ch. 8 of [Bar90].

The optimal fusion in a decentralized configuration, discussed in Section 8.6, is based on [Cho79].

The relaxation algorithm for multisensor data fusion, presented in Section 8.8, is based on Chapter 7 of [Bar90] and [DMPB90, PDBW92].

Recent trends in multisensor-multitarget tracking with the use of advanced probabilistic algorithms combined with artificial intelligence techniques have been discussed in [Yan86] and Chapters 10 and 11 of [Bar92].

8.10.2 Problems

8-1 Centralized estimation vs. fusion of local estimates. Given the scalar state equation

$$x(k+1) = x(k) + v(k) \quad k = 0, 1, \dots$$

with the process noise $v(k)$ zero-mean white with variance q and the two measurement equations corresponding to two independent sensors

$$z^m(k) = x(k) + w^m(k) \quad k = 1, 2, \dots; \quad m = i, j$$

with the measurement noises $w^m(k)$ zero-mean white with variance r and uncorrelated with the process noise. The initial estimates at each sensor are $\hat{x}^m(0|0)$, $m = i, j$, each with variance p_0 and the corresponding errors mutually uncorrelated and uncorrelated with both noises.

- 1) Find the local (decentralized), i.e., single-sensor-based state estimates at $k = 1$ and their covariances.
- 2) Find the cross-covariance of these two local estimates.
- 3) Using the state estimate fusion equations at $k = 1$, find the global (fused) estimate and its associated MSE. Indicate the weights of the initial estimates and the measurements at $k = 1$.
- 4) Find the centralized estimate at $k = 1$ obtained by a filter running on the fused data (initial estimates and measurements) at each point in time, and the associated covariance.
- 5) Compare the weightings of the data points (initial estimates and measurements) in the centralized estimate 4) to those in 3).
- 6) Evaluate the MSE in 3) and 4) for $p_0 = q = r = 1$. Comment on the results.
The following questions are all for steady-state:
- 7) Find the covariance for each decentralized filter in terms of q and r .
- 8) Find the cross-covariance of the estimation errors from the two local filters.

8.10.2 Problems

- 9) Find the MSE of fused estimate based on the local filters' outputs.
- 10) Find the covariance of the estimate from the centralized filter.
- 11) Compare the results from 9) and 10) for $q = r = 1$. Comment on the result.

Chapter 9

TRACKING WITH IMAGING SENSORS

9.1 INTRODUCTION

9.1.1 Outline

In all the previous algorithms the measurements were “points” — the coordinates of the single resolution cell or pixel in which a detection was declared by the signal processor.

This chapter deals with **measurement extraction** and their processing in the case where a detection is a set of cells or pixels in which the received signal indicates the existence of a target.

The first case considered deals with an **extended target** — when due to the size of the target or the sensor (e.g., its optics) the target appears as a “blob”: a collection of several (up to about 10) pixels. Assuming a “hot spot” target, one can use the **measured pixel intensities** in the **tracking window** (part of the sensor’s frame) to calculate the centroid of the target. This information can then be further processed by a tracker to estimate the target motion.

The second case deals with a **small image** — around 100 pixels — too small for feature extraction but too large for the pixel intensities to be meaningful for centroid calculation. Following **image segmentation** that leads to a clustered binary array, the centroid is calculated from the segmented area and then used in a tracker.

9.1.1 Outline

Section 9.2 describes the centroid estimation for the “hot spot” extended target and relates the video SNR to the single frame based centroid estimate accuracy (measurement noise variance). Also the offset estimation from two adjacent frames is considered and its measurement noise statistics are presented.

The corresponding tracking filter models for a single target are discussed in Section 9.3, and Section 9.4 treats the situation of two targets with overlapping images in the sensor’s field of view.

Section 9.5 presents a segmentation technique for small images and the use of the results to track the centroid of a real target.

9.1.2 Summary of Objectives

9.1.2 Summary of Objectives

Relate the measurement noise statistics of

- the single frame based centroid estimate
- the centroid offset estimate based on two adjacent frames

for a “hot spot” target to the video SNR for an infrared sensor.

Present the tracking filters for

- a single target
- two targets with crossing images

utilizing these measurements, with the filter using the offset measurements accounting for its measurement noise autocorrelation.

Show a multilayer segmentation technique that yields a binary image, the evaluation of the accuracy of the centroid of a segmented image and how the centroid of such an image can be used to track its motion.

9.2 MEASUREMENT EXTRACTION FOR AN EXTENDED TARGET

9.2.1 Introduction

In this section we first discuss the extraction of measurements for precision tracking of the **centroid** of an **hot spot target** from a **forward looking infrared (FLIR)** imaging sensor. This will be done using the *pixel intensities in the sensor's frame*.

The same methodology is applicable for other sensors where the target image is about 10 pixels, i.e., it is an **extended target**.

The statistical characterization of the centroid of a frame as a noisy linear measurement of the centroid of the target is obtained.

Another type of measurements is the **centroid offset (displacement)** measurement derived from the correlation of adjacent frames. It will be shown that the offset measurement noise is *autocorrelated* and is the output of a moving average system driven by white noise.

9.2.2 Modeling of the Image

9.2.2.2 Modeling of the Image

The **tracking window** — set of pixels used for tracking — consists of

$$m = m_x \times m_y, \quad (9.2.2-1)$$

pixels, with the pixels denoted by a single index $i = 1, \dots, m$. The measured intensity of pixel i is

$$I_i = s_i + n_i \quad (9.2.2-2)$$

where s_i is the signal (target) intensity and n_i is the noise intensity in pixel i .

Noise Model

The image noise is modeled as i.i.d. with first and second moments

$$E[n_i] = 0 \quad (9.2.2-3)$$

$$E[n_i n_j] = \sigma^2 \delta_{ij} \quad (9.2.2-4)$$

The mean of the noise is assumed to have been already subtracted from the measured intensities.

Notations

The **total target-related intensity** is denoted as

$$s = \sum_{i=1}^m s_i \quad (9.2.2-5)$$

where the summation is over all the pixels (the limits of the summations will be omitted from now on).

Denoting by m_s the number of target pixels — the **target extent** — the **average target intensity over its extent** is

$$\mu_s = \frac{s}{m_s} \quad (9.2.2-6)$$

The **average target intensity over the entire frame** is denoted as

$$\bar{s} = \frac{s}{m} \quad (9.2.2-7)$$

9.2.2 Modeling of the Image

The **contrast** of the target image w.r.t. the frame will be defined as

$$\kappa = \sum (s_i - \bar{s})^2 = \sum s_i^2 - \frac{s^2}{m} \quad (9.2.2-8)$$

This quantity, which is the signal's "variance" (multiplied by m) plays a key role in the accuracy of the target offset measurement obtained from correlating two consecutive frames.

The **frame SNR** is defined as

$$r = \frac{\sum s_i}{\text{RMS}(\sum n_i)} = \frac{s}{\sqrt{m}\sigma} = \frac{m_s \mu_s}{\sqrt{m}\sigma} \quad (9.2.2-9)$$

This differs from the **average pixel SNR** (over the target's extent)

$$r' = \frac{\mu_s}{\sigma} = r \frac{\sqrt{m}}{m_s} \quad (9.2.2-10)$$

While (9.2.2-10) is the "intuitive" SNR, as it will be seen later, it is (9.2.2-9) that enters into the measurement noise variances.

The Cartesian coordinates of pixel i will be denoted as (ξ_i, η_i) . Assuming the center of coordinates to be in the center of the frame, one has

$$\sum \xi_i = 0 \quad \sum \eta_i = 0 \quad (9.2.2-11)$$

The analysis will be done for the ξ coordinate; a similar analysis holds for the η coordinate.

Numerical Example

For the sake of illustration of the results, they will be evaluated numerically for an 8×8 array, i.e., $m = 64$.

The target is assumed to cover $m_s = 10$ pixels with constant intensity of $\mu_s = 6$, i.e., a "flat" signal is considered. The total target intensity is then $s = 60$. With noise RMS of $\sigma = 1$, one has

From (9.2.2-8) the contrast of the target image: $\kappa = 303.75 \approx 300$.

From (9.2.2-9) the frame SNR: $r = 7.5$.

From (9.2.2-10) the pixel SNR: $r' = 6$.

9.2.3 Estimation of the Centroid

The estimate of the ξ -coordinate of the location of the **image centroid based on the tracking window** (9.2.2-1) is, in pixel units, the “center of mass”

$$\hat{\xi}_c = \frac{\sum \xi_i I_i}{\sum I_i} \quad (9.2.3-1)$$

where ξ_i is the ξ coordinate of the center of pixel i and the measured intensity I_i corresponding to this pixel is given by (9.2.2-2).

The two sources of error in (9.2.3-1) are

- (a) the image noise: the measured pixel intensities are noisy,
- (b) the image discretization: the pixels are of finite size and their intensities are assumed to be located in their centers, rather than their centroids (which are not available due to the sensor’s resolution limitation).

The **coefficient of variation** c_v of a random variable x is defined as the ratio of its standard deviation to its mean

$$c_v(x) \triangleq \frac{\sigma_x}{\bar{x}} \quad (9.2.3-2)$$

Assumption

It will be assumed that

$$c_v \left(\sum I_i \right) = c_v \left(\sum (s_i + n_i) \right) = \frac{\sqrt{m}\sigma}{s} = \frac{1}{r} \ll 1 \quad (9.2.3-3)$$

where notations (9.2.2-3)–(9.2.2-5) have been used. In other words, the coefficient of the variation of the denominator of (9.2.3-1) is assumed “small” — practically 0.15 or less. For the numerical example described in Subsection 9.2.2, it is 0.13.

With this assumption, derivations presented in ([Bar92]: Ch. 5) yield the **image centroid measurement** equation

$$\hat{\xi}_c = \xi_c + w_c \quad (9.2.3-4)$$

where the centroid measurement noise w_c is zero-mean and has variance

$$\sigma_w^2 = \xi_c^2 \frac{m\sigma^2}{s^2 + m\sigma^2} + \frac{\sigma^2 \sum \xi_i^2}{s^2 + m\sigma^2} + \frac{\sum s_i^2}{12s^2} \quad (9.2.3-5)$$

where ξ_c above is the actual centroid.

Using (9.2.2-8) and (9.2.2-9), the above can be rewritten as

$$\sigma_w^2 = \xi_c^2 \frac{1}{r^2 + 1} + \frac{\sum \xi_i^2}{m(r^2 + 1)} + \frac{\kappa + s^2/m}{12s^2} \quad (9.2.3-6)$$

9.2.3 Estimation of the Centroid

Numerical Example

For the numerical example described in Subsection 9.2.2, with $m = 64$, $r = 7.5$, $\kappa = 300$, and assuming $\xi_c = 1$, the variance (9.2.3-6) is

$$\sigma_c^2 = (0.13)^2 + (0.3)^2 + (0.09)^2 \approx (0.34)^2 \quad (9.2.3-7)$$

i.e., about 1/3 of a pixel RMS value.

9.2.4 The Offset Measurement from Image Correlation

9.2.4 The Offset Measurement from Image Correlation

The **correlation coefficient between two frames** from times k and $k - 1$ with displacement δ (along one axis) is

$$\hat{\rho}_\delta(k) = \frac{\sum[I_i(k) - \bar{I}(k)][I_{i_\delta}(k - 1) - \bar{I}(k - 1)]}{\left\{\sum[I_i(k) - \bar{I}(k)]^2 \sum[I_{i_\delta}(k - 1) - \bar{I}(k - 1)]^2\right\}^{1/2}} \quad (9.2.4-1)$$

where $I_i(k)$ is the intensity of pixel i at time k ; i_δ is the index of the pixel in the image at $k - 1$ corresponding to pixel i at k when the image from $k - 1$ has been shifted by δ pixels along the axis under consideration (δ is an integer; with two axes the shifts form a 2-dimensional array of integers). The summations are only over the overlying pixels and the denominator is a normalizing factor. The average intensities $\bar{I}(k)$ and $\bar{I}(k - 1)$ are also based only on the overlying pixels.

Let the three highest image correlations for displacements in one direction be for the displacements $\delta - 1$, δ , $\delta + 1$ with the highest in the center, i.e.,

$$\hat{\rho}_{\delta-1} < \hat{\rho}_\delta > \hat{\rho}_{\delta+1} \quad (9.2.4-2)$$

The maximum correlation, using a **parabolic interpolation** between the above three points, is at location

$$\hat{d} = \delta + \frac{\alpha_1}{\alpha_2} \quad (9.2.4-3)$$

where

$$\alpha_1 \triangleq \hat{\rho}_{\delta+1} - \hat{\rho}_{\delta-1} \quad \alpha_2 \triangleq 2(2\hat{\rho}_\delta - \hat{\rho}_{\delta-1} - \hat{\rho}_{\delta+1}) \quad (9.2.4-4)$$

The estimated location (9.2.4-3) of the peak correlation is the **image offset measurement**. As shown in ([Bar92]: Ch. 5), this can be written as

$$\hat{d} = d + w_d \quad (9.2.4-5)$$

where d is the true offset (displacement) and the measurement noise w_d is zero-mean with variance given below.

9.2.4 The Offset Measurement from Image Correlation

This measurement noise sequence is *autocorrelated*:

- the correlation coefficient magnitude between two such noises adjacent in time can be in magnitude up to 0.5;
- if the separation is more than one sampling period they are, however, uncorrelated.

This is a consequence of the fact that the image correlation is affected by the video noises in the two consecutive frames it correlates: the offsets from $k - 1$ to k and from k to $k + 1$ have half the noises (from time k) common.

The variance of the offset (displacement) measurement noise in (9.2.4-5) is

$$\boxed{\sigma_d^2 = \frac{2(2\kappa\sigma^2 + m\sigma^4)}{\kappa^2(\alpha_2^2 + 12\alpha_1^2)}} \quad (9.2.4-6)$$

The actual value of the correlation between adjacent offset noises is -0.5 .

Numerical Example

For the numerical example of Subsection 9.2.2, with nominal values $\hat{\rho}_\delta = 0.8$, $\hat{\rho}_{\delta-1} = \hat{\rho}_{\delta+1} = 0.6$, one has $\sigma_d = 0.15$, i.e., 1/6 of a pixel RMS value.

These results show that one can obtain subpixel accuracy even from single measurements. Further improvement in this can be obtained by filtering these measurements, as will be shown in Section 9.3.

9.2.5 Application to a Gaussian Plume Target

9.2.5 Application to a Gaussian Plume Target

Consider a target with a **Gaussian plume intensity** in the sensor's focal plane

$$S(\xi, \eta) = S_{\max} e^{-\frac{1}{2} \left[\frac{(\xi - \xi_c)^2}{a_\xi^2} + \frac{(\eta - \eta_c)^2}{a_\eta^2} \right]} \quad (9.2.5-1)$$

with the center at (ξ_c, η_c) , semiaxes a_ξ, a_η (**footprint**, assumed for simplicity oriented along the sensor's coordinates).

The total target-related intensity (9.2.2-5) is

$$s = \sum s_i = S_{\max} 2\pi a_\xi a_\eta \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi a_\xi a_\eta} e^{-\frac{1}{2}[\cdot]} d\xi d\eta = S_{\max} 2\pi a_\xi a_\eta \quad (9.2.5-2)$$

where the exponent is the same as in (9.2.5-1).

The sum of the squares of the intensities, needed in (9.2.3-5), can be obtained numerically using (9.2.5-1), or one can use the following approximation

$$\sum s_i^2 = S_{\max}^2 2\pi \frac{a_\xi a_\eta}{2} \int \int \frac{1}{2\pi \frac{a_\xi a_\eta}{2}} e^{-[\cdot]} d\xi d\eta = S_{\max}^2 \pi a_\xi a_\eta \quad (9.2.5-3)$$

With (9.2.5-2) and (9.2.5-3), the variance (9.2.3-5) of the centroid measurement is

$$\boxed{\sigma_c^2 = \frac{\xi_c^2 m \sigma^2 + \sigma^2 \sum \xi_i^2}{(S_{\max} 2\pi a_\xi a_\eta)^2 + m \sigma^2} + \frac{1}{48\pi a_\xi a_\eta}} \quad (9.2.5-4)$$

The "contrast" (9.2.2-8) is

$$\kappa = \sum (s_i - \bar{s})^2 = \sum s_i^2 - \frac{s^2}{m} = S_{\max}^2 \pi a_\xi a_\eta \left[1 - \frac{4\pi a_\xi a_\eta}{m} \right] \quad (9.2.5-5)$$

Numerical Example and Validation

Using the numerical values $a_\xi = 1.5, a_\eta = 1$ one has

$$s = 3\pi S_{\max} \quad \sum s_i^2 = 1.5\pi S_{\max}^2 \quad (9.2.5-6)$$

With $m = 64$ and $\sigma = 1$ the centroid measurement variance (9.2.5-4) is

$$\sigma_c^2 = \frac{64\xi_c^2 + 336}{64 + 9\pi^2 S_{\max}^2} + \frac{1}{72\pi} \quad (9.2.5-7)$$

The contrast (9.2.5-5) is, for the above numbers and $S_{\max} = 10$, i.e., a *peak pixel SNR* of 10,

$$\kappa = 100\pi \cdot 1.5 \left[1 - \frac{6\pi}{64} \right] = 332 \quad (9.2.5-8)$$

9.2.5 Application to a Gaussian Plume Target

To check the theoretical derivations, 100 Monte Carlo runs were made for a Gaussian plume image with the parameters as above. The image was centered at the origin, i.e., $\xi_c = 0$.

Table 9.2.5-1 shows the comparison of the experimental and theoretical results.

	Theoretical value	Sample average
σ_c	0.195	0.18
σ_d	0.104	0.105
Correlation	-0.5	-0.52

Table 9.2.5-1: Measurement noise statistics — theoretical values vs. Monte Carlo averages.

For the offset measurements, the average correlations were $\hat{\rho}_\delta = 0.85$, $\hat{\rho}_{\delta+1} \approx \hat{\rho}_{\delta-1} = 0.58$, which yield $\alpha_1 = 0$, $\alpha_2 = 1.1$ and from (9.2.4-6) one obtains $\sigma_d = 0.104$.

The sample correlation coefficient between two consecutive offset measurement noises was obtained from 100 runs as -0.52 and one can conclude that the actual correlation coefficient is -0.5.

9.2.6 Statistical Properties of Centroid Measurements — Summary

9.2.6 Statistical Properties of Centroid Measurements — Summary

The extraction of two measurements pertaining to the centroid of the image of an extended target (size around 10 pixels) in the focal plane of a FLIR has been discussed:

- the centroid location in a single frame
- the centroid displacement (offset) between two consecutive frames.

The single frame based centroid estimate has been shown to be a noisy measurement of the actual centroid with an additive zero-mean white noise whose variance has been expressed in terms of

- the target image parameters — size, intensity
- the background (video) noise.

The offset estimate, obtained from a spatial correlation of two frames is a noisy measurement of the true offset, with an additive measurement noise that is

- zero-mean
- with a variance whose explicit expression has been presented
- autocorrelated: adjacent noises have a correlation coefficient, shown to be -0.5 .

9.3 PRECISION TRACKING OF THE IMAGE CENTROID FOR A SINGLE TARGET

9.3.1 Introduction

This section presents the state models to be used for the estimation of the centroid of a moving target using the following measurements:

- (a) The **centroid measurement** based on a single frame, and
- (b) The **centroid offset measurement** based on two consecutive frames.

As discussed in the previous section, the offset measurement noise is autocorrelated.

First the estimation is presented assuming the offset measurement noise to be white — the autocorrelation is ignored to be able to use a simpler filter.

Then the correlation of these noises is accounted for via an *augmented state model*.

The simulation results presented in Subsection 9.3.3 indicate that the best estimation performance is obtained by accounting for the autocorrelated offset measurement noise. This can yield very good *subpixel accuracy* — in the range of 0.1 pixel RMS error.

9.3.2 Filter Model for White Measurement Noise

9.3.2 Filter Model for White Measurement Noise

Denoting the position of the image centroid at time k in one of the coordinates as $x_1(k)$, the centroid measurement (9.2.3-4) is

$$z_1(k) = x_1(k) + w_c(k) \quad (9.3.2-1)$$

where the centroid measurement noise $w_c(k)$ has variance σ_c^2 given in (9.2.3-5).

The displacement measurement (9.2.4-5) can be written as

$$z_2(k) = x_1(k) - x_1(k-1) + w_d(k) \quad (9.3.2-2)$$

where w_d is the displacement measurement noise from (9.2.4-5) and its variance σ_d^2 is given in (9.2.4-6).

Assuming the displacement measurement noise to be white, the state equation for tracking a nearly constant velocity target using the above measurements is

$$\begin{aligned} x(k+1) &= \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix} v(k) \end{aligned} \quad (9.3.2-3)$$

where the state consists of the current position (x_1), current velocity (x_2) and previous position (x_3), T is the sampling period and $v(k)$ is the zero mean, white motion process noise (acceleration), with variance q .

The measurement equation is

$$z(k) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} x(k) + w(k) \quad (9.3.2-4)$$

The covariance of the measurement noise is

$$R = \text{diag}(\sigma_c^2, \sigma_d^2) \quad (9.3.2-5)$$

The state estimation problem based on (9.3.2-3) and (9.3.2-4) is a standard one.

9.3.3 Filter Model for Autocorrelated Measurement Noise

9.3.3 Filter Model for Autocorrelated Measurement Noise

The autocorrelated measurement noise in the displacement observation can be incorporated into the state estimation as follows.

From the discussion of Subsection 9.2.4, it follows that

$$E[w_d(k)w_d(j)] = \begin{cases} \sigma_d^2 & k = j \\ \alpha\sigma_d^2 & |k - j| = 1 \\ 0 & |k - j| > 1 \end{cases} \quad (9.3.3-1)$$

where $\alpha = -0.5$. Such a noise is the output of the following ***moving average system*** driven by white noise

$$w_d(k) = \beta_1 v_2(k) + \beta_2 v_2(k-1) \quad (9.3.3-2)$$

where $v_2(k)$ is a zero-mean white sequence with variance σ_d^2 . The weighting coefficients in (9.3.3-2) are

$$\beta_1 = \frac{1}{2}(\sqrt{1+2\alpha} + \sqrt{1-2\alpha}) \quad \beta_2 = \frac{1}{2}(\sqrt{1+2\alpha} - \sqrt{1-2\alpha}) \quad (9.3.3-3)$$

This type of autocorrelated noise is different from the usual autoregressive model (see [BL93], Section 8.4) and requires ***state augmentation***.

The augmented state equation needed to incorporate the autocorrelated measurement noise model (9.3.3-2) is

$$\begin{aligned} x(k+1) &= \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = Fx(k) + Gv(k) \\ &= \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \end{aligned} \quad (9.3.3-4)$$

Note that in (9.3.2-3) the state was a 3-dimensional vector while in (9.3.3-4) the state is a 4-dimensional vector.

9.3.3 Filter Model for Autocorrelated Measurement Noise

The measurement equation is

$$z(k) = Hx(k) + w(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \quad (9.3.3-5)$$

where $v_1(k)$ is the motion process noise, $v_2(k)$ is the white noise input to the autocorrelated noise (9.3.3-2) in the displacement measurement (with variance σ_d^2), $w_1(k)$ is the noise in the centroid measurement, and

$$w_2(k) \triangleq \beta_1 v_2(k) \quad (9.3.3-6)$$

The measurement and process noise sequences are *white* but $v(k)$ and $w(k)$ are *cross-correlated*. This is described as follows:

$$E[Gv(k)v(j)'G'] = GQG'\delta_{kj} \triangleq G \begin{bmatrix} q & 0 \\ 0 & \sigma_d^2 \end{bmatrix} G' \delta_{kj} \quad (9.3.3-7)$$

$$E[w(k)w(j)'] = R\delta_{kj} \triangleq \begin{bmatrix} \sigma_c^2 & 0 \\ 0 & \beta_1^2 \sigma_d^2 \end{bmatrix} \delta_{kj} \quad (9.3.3-8)$$

$$E[v(k)w(j)'] = U\delta_{kj} \triangleq \begin{bmatrix} 0 & 0 \\ 0 & \beta_1 \sigma_d^2 \end{bmatrix} \delta_{kj} \quad (9.3.3-9)$$

where the last equation indicates cross-correlation between the two noise sequences, i.e., it is *not a standard filtering problem*.

Thus, having carried out the *prewhitening of the measurement noise*, one has a cross-correlation problem.

Next the state equation will be rewritten in a manner that one obtains *decorrelation of the noise sequences*.

9.3.3 Filter Model for Autocorrelated Measurement Noise

The New State Equation in Standard Form

The technique described in [BL93], Section 8.3 can be used to obtain the state estimation filter. The plant equation is rewritten such that it has a *new process noise* uncorrelated with the measurement noise. Using an arbitrary matrix T , to be determined later, one can write

$$\begin{aligned} x(k+1) &= Fx(k) + Gv(k) + T[z(k) - Hx(k) - w(k)] \\ &= (F - TH)x(k) + Gv(k) - Tw(k) + Tz(k) \end{aligned} \quad (9.3.3-10)$$

Then one can write the new state equation

$$x(k+1) = F^*x(k) + v^*(k) + u^*(k) \quad (9.3.3-11)$$

with the new transition matrix

$$F^* \triangleq F - TH \quad (9.3.3-12)$$

new process noise

$$v^*(k) \triangleq Gv(k) - Tw(k) \quad (9.3.3-13)$$

and input

$$u^*(k) \triangleq Tz(k) \quad (9.3.3-14)$$

To accomplish the **noise sequence decorrelation**, the cross-correlation between the new process noise and the measurement noise is set to zero

$$E[v^*(k)w(k)'] = 0 \quad (9.3.3-15)$$

which yields

$$T = GUR^{-1} \quad (9.3.3-16)$$

The covariance of the new process noise can be shown to be

$$Q^* \triangleq E[v^*(k)v^*(k)'] = G(Q - UR^{-1}U')G' \quad (9.3.3-17)$$

The state estimation problem based on (9.3.3-11) and (9.3.3-5) with the noise covariances given by (9.3.3-8), (9.3.3-15) and (9.3.3-17) is now a *standard one* and can be solved using the standard Kalman filter technique.

9.3.4 Hot Spot Target Tracking Simulation Results

9.3.4 Hot Spot Target Tracking Simulation Results

A target whose intensity in the sensor's focal plane is a Gaussian plume as given by (9.2.5-1) with a peak SNR = 10 was considered for simulations.

Scenario Parameters

Target motion described by a second order kinematic model with white noise acceleration with $q = (0.05)^2$ in each of the two coordinates was considered. The sampling period was $T = 1$ and the filters were initialized using two-point differencing.

Using the following values of the parameters: $S_{\max} = 10$, $a_\xi = 1.5$, $a_\eta = 1$ yields $\sigma_c = 0.195$, $\sigma_d = 0.104$ and the measurement noise correlation coefficient is $\alpha = -0.5$.

Possible Filter Designs

The following filters were run:

Filter A. *Centroid measurements only.* In this case the state equation is (9.3.2-3) without the third state component, which is needed only when displacement measurements are also available.

Filter B. Centroid and displacement measurements with the *displacement measurement noise assumed white*. The state equation is (9.3.2-3) and the measurement equation (9.3.2-4).

Filter C. With centroid and displacement measurements, with the *displacement measurement noise accounted for as autocorrelated* according to (9.3.3-1) with $\alpha = -0.5$. The state equation is then (9.3.3-11) and the measurement equation (9.3.3-5) with noise covariances according to (9.3.3-8), (9.3.3-15) and (9.3.3-17).

Table 9.3.4-1 summarizes the results in terms of achievable accuracies for the problem considered. Also, the Normalized Estimation Error Squared (NEES — see [BL93], Section 5.4) from 100 runs is shown to be very close to its theoretical average indicating that the filters are consistent.

These results show that one can obtain very good subpixel accuracy in the range of 0.1 pixel RMS error. The filter designed taking into consideration the autocorrelated displacement measurement noise is by far the best.

9.3.4 Hot Spot Target Tracking Simulation Results

Filter	Position variance	Velocity variance	Average NEES	Expected NEES
A	0.0194	0.0058	3.87	4
B	0.0138	0.0041	5.53	6
C	0.0071	0.0032	7.79	8

Table 9.3.4-1: Position and velocity variances (in steady state) for each coordinate and filter consistency verification from 100 runs.

Relationship to Real Target Parameters

The target maneuvering index considered here is about 0.25.

The process noise can be related as follows to the target acceleration: assuming that a pixel corresponds to 1m (target extent is about 2×5 pixels) and the actual sampling rate is $1/T_0 = 30\text{Hz}$, the process noise standard deviation corresponds to an acceleration

$$a_0 T_0^2 = \sqrt{q} T^2 \quad (9.3.4-1)$$

This yields for the assumed value of q an acceleration $a_0 \approx 5g$, which is quite significant.

For targets maneuvering less, one can use lower process noise q in the filter, which results in even smaller estimation errors.

9.3.5 Image Centroid Tracking — Summary

9.3.5 Image Centroid Tracking — Summary

The filtering of the following measurements pertaining to the centroid of the image of a extended target has been discussed:

- the centroid location in a single frame
- the centroid displacement (offset) between two consecutive frames.

The following possible filter designs have been considered:

- Design A: Filter using centroid measurements only. In this simplest case, which does not require the frame correlation evaluations for the offset, the state equation is 2-dimensional per coordinate and the filter is a standard one.
- Design B: Filter using centroid and displacement measurements *assuming the displacement measurement noise white*. The state is 3-dimensional per coordinate and the filter is a standard one.
- Design C: Filter with centroid and displacement measurements, with the displacement measurement noise accounted for as autocorrelated. The state is 4-dimensional per coordinate and the filter requires prewhitening of the autocorrelated measurement noise and decorrelation of the two noise sequences.

Design A is the least expensive, both in the measurement extraction and in filtering.

Design C, the most expensive, can achieve an MS error about 2.5 times smaller in position than Design A.

Design B is probably closer in cost to C but in performance closer to A.

9.4 PRECISION TRACKING OF THE IMAGE CENTROIDS FOR TWO TARGETS WITH OVERLAPPING IMAGES

9.4.1 Introduction

The problem of crossing targets with measurements obtained with a FLIR sensor can occur when the tracking is done based on a small image of a jet engine exhaust or a missile exhaust plume.

The overlap of the two images generates a *mixed (merged) target image* with a centroid that can be assumed a linear combination of the centroids of the individual targets.

The crossing of the lines of sight results in *ambiguity of the offset measurements* from the two targets over several sampling times. In light of this, proper association of the measurements to targets can be difficult.

Furthermore, this “sharing” of the measurements over several sampling times, also causes a dependence of the state estimation errors for the two targets. The association problem can be handled using the JPDA augmented with a measurement merging model (JPDAM — Section 6.4).

The extension of the JPDAF called the Joint Probabilistic Data Association Coupled Filter (JPDACF) which performs filtering in a coupled manner for targets with *common* measurements (Subsection 6.2.7) is employed for state estimation.

The resulting algorithm is the ***Joint Probabilistic Data Association Merged-Measurement Coupled Filter (JPDAMCF)***.

9.4.2 The Overlapping Images

9.4.2 The Overlapping Images

Modeling of the Image Overlap

The intensity in pixel i is the sum of the two individual intensities

$$I_i = s_{1i} + s_{2i} + n_i \quad i = 1, \dots, m \quad (9.4.2-1)$$

where s_{ji} is the signal (target) intensity of target j , $j = 1, 2$, in pixel i and n_i is the noise intensity in pixel i after subtracting its mean; m is the total number of pixels in the frame.

The Centroid of the Combined Image

The estimate of the ξ -coordinate of the location of the centroid of the combined image at time k , is, in pixel units

$$\hat{\xi}_c = \frac{\sum \xi_i I_i}{\sum I_i} \quad (9.4.2-2)$$

where ξ_i is the ξ coordinate of the center of pixel i and the measured intensity I_i corresponding to this pixel is given by (9.4.2-1). The summations above are over the entire frame.

As shown in Chapter 5 of [Bar92], the estimated centroid (9.4.2-2) of the combined image relates to the centroids of the two targets according to

$$\hat{\xi}_c = \iota \xi_{c_1} + (1 - \iota) \xi_{c_2} + w_c \quad (9.4.2-3)$$

where ξ_{ci} is the true centroid of target i ; ι is the **image-mixing parameter** and w_c is the zero-mean white measurement noise with variance given by (9.2.3-6) with the total target related intensity

$$s = \sum_{i=1}^m (s_{1i} + s_{2i}) \quad (9.4.2-4)$$

The image mixing parameter has to be estimated before the targets overlap — this is also discussed in Chapter 5 of [Bar92].

9.4.2 The Overlapping Images

The Offset Measurement with Two Targets

The correlation coefficient between two frames from times $k, k - 1$ with displacement (offset) Δ is, similarly to (9.2.4-1)

$$\hat{\rho}_\Delta(k) = \frac{\sum [I_i(k) - \bar{I}(k)][I_{i_\Delta}(k-1) - \bar{I}(k-1)]}{\{\sum [I_i(k) - \bar{I}(k)]^2 \sum [I_{i_\Delta}(k-1) - \bar{I}(k-1)]^2\}^{1/2}} \quad (9.4.2-5)$$

where in this case Δ is a 2-dimensional vector of integers

$$\Delta = [\delta_\xi \ \delta_\eta]' \quad (9.4.2-6)$$

and $I_i(k)$ is the intensity of pixel i at time k given by (9.4.2-1).

Due to the existence of two targets in the image plane, the correlation function (9.4.2-5) is *multimodal*.

Example

Figure 9.4.2-1 shows three consecutive sensor frames using some numerical values for the intensity distributions and the velocities for the two targets.

The image of target 1 is moving with heading of 135° with a speed of approximately one pixel per sampling period and the image of target 2 is moving with heading of 27° with a speed of approximately two pixels per period.

The image correlation matrix elements between frames at $k - 1$ and k , and between frames at k and $k + 1$ are shown in Tables 1 and 2, respectively. The displacement measurements are obtained using parabolic interpolation for each peak with its neighbors.

From Tables 9.4.2-1 and 9.4.2-2 it can be seen that there are two peaks. While in this case there are no false peaks, one does not know which peak belongs to which target: there exists an *ambiguity in the offset measurements* that can be accounted for by using the JPDA.

The algorithm assumes two peaks in the correlation matrix. If the signal to noise ratio is significantly lower one might have additional peaks and the algorithm can be modified to account for false peaks.

9.4.2 The Overlapping Images

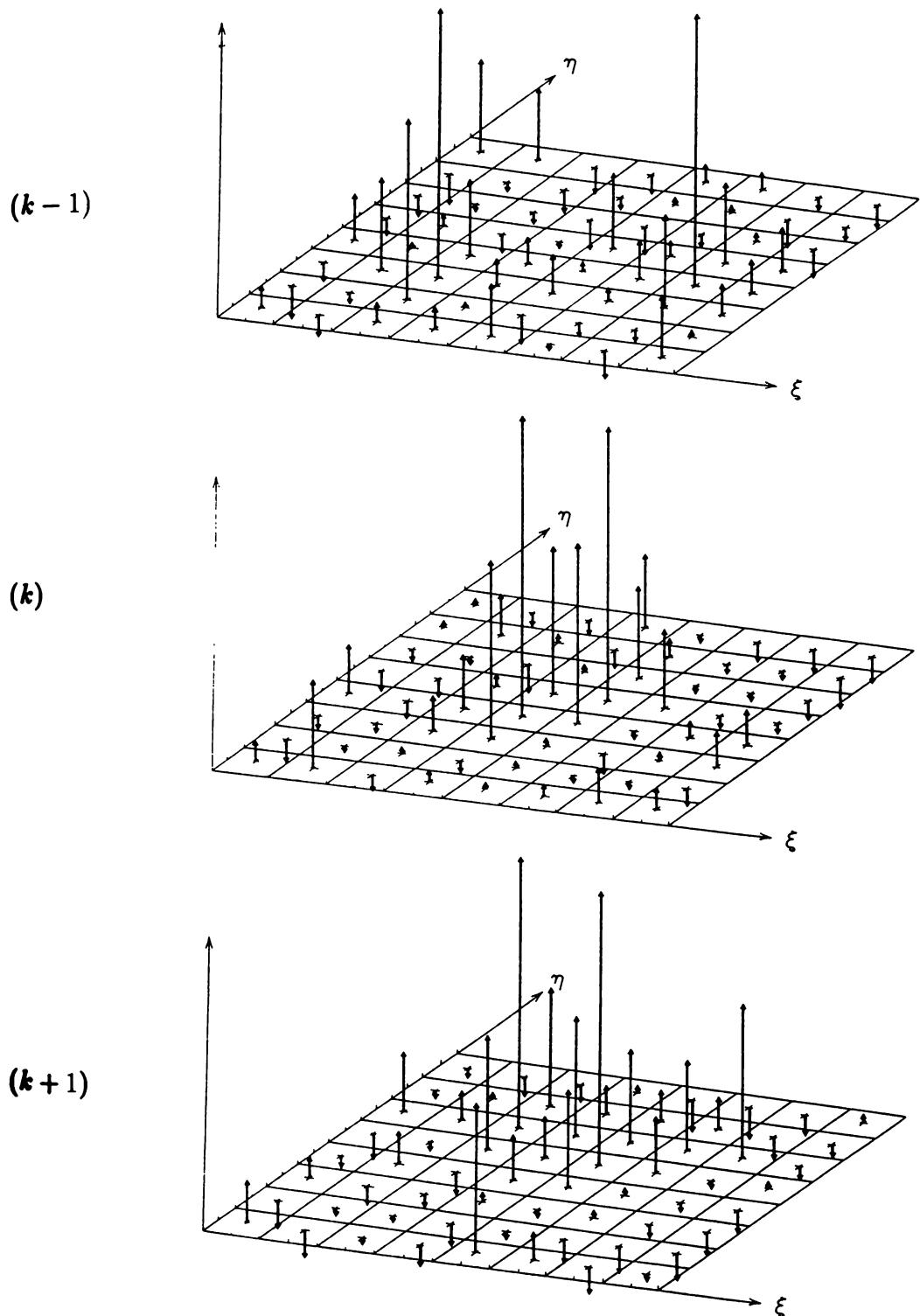


Figure 9.4.2-1: Intensities for two targets in the frame at three consecutive times.

9.4.2 The Overlapping Images

0.52	-0.07	-0.15	0.42	0.13	-0.16	-0.03	0.19
0.13	-0.20	-0.16	0.72	0.51	-0.05	-0.25	0.12
-0.20	-0.24	-0.16	0.23	0.38	0.00	-0.32	-0.27
-0.18	-0.26	-0.12	0.00	0.02	-0.18	-0.23	-0.17
-0.12	-0.18	-0.06	0.09	0.07	-0.07	-0.21	-0.09
-0.10	-0.08	-0.04	0.17	0.46	0.29	-0.06	-0.10
-0.19	-0.19	-0.08	0.07	0.42	0.55	0.28	0.05
-0.27	-0.18	0.26	-0.04	-0.05	0.01	-0.21	-0.26

Table 9.4.2-1: Image correlation coefficient matrix between noisy frames at $k - 1$ and k .

-0.10	-0.32	-0.26	-0.19	-0.04	-0.08	-0.17	-0.16
-0.32	-0.28	-0.29	-0.15	-0.09	-0.13	-0.36	-0.31
-0.12	-0.22	-0.22	0.10	0.41	0.21	0.17	-0.28
0.21	-0.08	-0.03	0.09	0.32	0.45	0.15	-0.19
0.03	-0.14	-0.09	0.35	0.36	0.26	0.04	-0.17
-0.17	-0.22	-0.10	0.24	0.48	0.17	-0.20	-0.15
-0.30	-0.32	-0.20	-0.09	0.18	0.10	-0.29	-0.40
-0.25	-0.28	-0.27	-0.17	-0.11	0.01	-0.15	-0.22

Table 9.4.2-2: Image correlation coefficient matrix between noisy frames at k and $k + 1$.

Summary of the Available Measurements

There are two kinds of measurements available during the overlap period for the filter to process:

- a single merged measurement for the centroid of the mixed image, and
- two centroid displacement measurements with ambiguous origin.

9.4.3 The State Estimation

The state estimation for two crossing targets is done stagewise as follows:

1. If there are two distinct regions in the distributed image (obtained from segmentation), the states of the two targets under consideration are estimated separately using two filters (one associated with each target, based on the models from Subsections 9.3.2 or 9.3.3).
2. If the images at the sensor overlap (this is evaluated by the segmentation algorithm) the measurements available are:

- (i) A single merged measurement for the centroid of the overlapping images and
- (ii) Two displacement measurements which are “shared” by both the targets (i.e., have ambiguous origin) during the overlap period.

In this situation the two targets can no longer be tracked reliably with separate filters using a “nearest neighbor” assignment rule for the ambiguous measurements.

The problem can be handled by using the **Joint Probabilistic Data Association Merged-Measurement Coupled Filter (JPDA-MCF)** which is presented in the next subsection in two versions:

- A. Assuming the displacement measurement noise as white;
- B. Modeling this noise correctly as autocorrelated — this requires state augmentation, but yields better performance.

9.4.4 Filter Model for White Measurement Noise

9.4.4 Filter Model for White Measurement Noise

Assuming the displacement measurement noise white, the state model is

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{v}(k) \quad (9.4.4-1)$$

with the stacked vector of the states of the two targets under consideration

$$\mathbf{x}(k) = \begin{bmatrix} x^1(k) \\ x^2(k) \end{bmatrix} \quad (9.4.4-2)$$

and the block-diagonal state transition matrix, noise distribution and stacked process noise vector given by

$$\mathbf{F} = \begin{bmatrix} F^1 & 0 \\ 0 & F^2 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} G^1 & 0 \\ 0 & G^2 \end{bmatrix} \quad \mathbf{v}(k) = \begin{bmatrix} v^1(k) \\ v^2(k) \end{bmatrix} \quad (9.4.4-3)$$

and $x^t(k)$, F^t , G^t , $v^t(k)$, $t = 1, 2$ are as in Subsection 9.3.2.

The measurement model has to be modified due to the **merged centroid measurement**. For each coordinate the measurement vector is

$$\mathbf{z}(k) = \begin{bmatrix} z_c(k) \\ z_{d_1}(k) \\ z_{d_2}(k) \end{bmatrix} = \begin{bmatrix} \iota & 0 & 0 & 1-\iota & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1^1(k) \\ x_2^1(k) \\ x_3^1(k) \\ x_1^2(k) \\ x_2^2(k) \\ x_3^2(k) \end{bmatrix} + \begin{bmatrix} w_c(k) \\ w_{d_1}(k) \\ w_{d_2}(k) \end{bmatrix} \quad (9.4.4-4)$$

where the subscripts c and d_i , $i = 1, 2$, represent the centroid measurement and the centroid displacement measurements, respectively.

The mixing parameter ι depends on the targets' relative intensities and has to be determined experimentally (see Chapter 5 of [Bar92]).

The stacked measurement vector in (9.4.4-4) has the first component the merged centroid measurement, while components two and three are from targets 1 and 2, or 2 and 1 (in which case they are switched around), respectively, resulting in the **ambiguous displacement measurements**.

The **sequential updating procedure** for the state of the system (9.4.4-1) is used for the two types of measurements available at a given sampling time.

9.4.4 Filter Model for White Measurement Noise

The measurement j at time k is

$$z(k, j) = H(k, j)\mathbf{x}(k) + w(k, j) \quad j = 1, 2 \quad (9.4.4-5)$$

where

$$z(k, 1) = z_c(k) \quad H(k, 1) = [\iota \ 0 \ 0 \ 1 - \iota \ 0 \ 0] \quad (9.4.4-6)$$

and

$$z(k, 2) = \begin{bmatrix} z_{d_1}(k) \\ z_{d_2}(k) \end{bmatrix} \quad H(k, 2) = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \quad (9.4.4-7)$$

Thus (9.4.4-6) and (9.4.4-7) represent the merged centroid measurement component and the centroid displacement measurement components in (9.4.4-4), respectively.

The Sequential Updating Procedure

(a) Update with the merged centroid measurement

Denote the predicted state at time k and its covariance as

$$\hat{\mathbf{x}}(k|k, 0) \triangleq \hat{\mathbf{x}}(k|k - 1) \quad (9.4.4-8)$$

$$\mathbf{P}(k|k, 0) \triangleq \mathbf{P}(k|k - 1) \quad (9.4.4-9)$$

The updated state with the measurement (9.4.4-6) at time k is

$$\hat{\mathbf{x}}(k|k, 1) = \hat{\mathbf{x}}(k|k, 0) + W(k, 1)[z(k, 1) - \hat{z}(k, 1)] \quad (9.4.4-10)$$

where

$$\hat{z}(k, 1) = H(k, 1)\hat{\mathbf{x}}(k|k, 0) \quad (9.4.4-11)$$

$$W(k, 1) = \mathbf{P}(k|k, 0)H(k, 1)'S(k, 1)^{-1} \quad (9.4.4-12)$$

$$S(k, 1) = H(k, 1)\mathbf{P}(k|k, 0)H(k, 1)' + R(k, 1) \quad R(k, 1) = \sigma_c^2 \quad (9.4.4-13)$$

and

$$\mathbf{P}(k|k, 1) = \mathbf{P}(k|k, 0) - W(k, 1)S(k, 1)W(k, 1)' \quad (9.4.4-14)$$

9.4.4 Filter Model for White Measurement Noise

(b) *Update with the centroid displacement measurement*

The state update with (9.4.4-7) is completed as follows

$$\hat{\mathbf{x}}(k|k, 2) = \hat{\mathbf{x}}(k|k, 1) + W(k, 2) \sum_{\theta} P\{\theta|Z^k\} [z(k, 2, \theta) - \hat{z}(k, 2)] \quad (9.4.4-15)$$

where the joint association event $\theta \in \{\theta_1, \theta_2\}$ and

$$z(k, 2, \theta) = \begin{bmatrix} z_{j_1(\theta)}(k, 2) \\ z_{j_2(\theta)}(k, 2) \end{bmatrix} \quad (9.4.4-16)$$

is the measurement (9.4.4-7); $j_t(\theta)$ is the index of the measurement associated with target t in event θ at time k and

$$\hat{z}(k, 2) = H(k, 2)\hat{\mathbf{x}}(k|k, 1) \quad (9.4.4-17)$$

The conditional probability for a joint association event θ is

$$P\{\theta|Z^k\} = \frac{1}{c} f_{t_{j_1}, t_{j_2}}[z_{j_1(\theta)}(k, 2), z_{j_2(\theta)}(k, 2)] \quad (9.4.4-18)$$

where $f_{t_{j_1}, t_{j_2}}$ is the joint pdf of the measurements of the targets under consideration; t_j is the target to which $z_j(k, 2)$ is associated in event $\theta \in \{\theta_1, \theta_2\}$ and c is the normalizing constant.

The filter gain in (9.4.4-15) and the innovation covariance are

$$W(k, 2) = \mathbf{P}(k|k, 1)H(k, 2)'S(k, 2)^{-1} \quad (9.4.4-19)$$

$$S(k, 2) = H(k, 2)\mathbf{P}(k|k, 1)H(k, 2)' + R(k, 2) \quad (9.4.4-20)$$

where $H(k, 2)$ is given in (9.4.4-7) and

$$R(k, 2) = \begin{bmatrix} \sigma_{d_1}^2 & 0 \\ 0 & \sigma_{d_2}^2 \end{bmatrix} \quad (9.4.4-21)$$

The covariance update equation and the final updated state are

$$\mathbf{P}(k|k, 2) = \mathbf{P}(k|k, 1) - W(k, 2)S(k, 2)^{-1}W(k, 2)' \triangleq \mathbf{P}(k|k) \quad (9.4.4-22)$$

$$\hat{\mathbf{x}}(k|k) \triangleq \hat{\mathbf{x}}(k|k, 2) \quad (9.4.4-23)$$

The above summarizes the **JPDA-MCF** with sequential updating for the centroid and displacement measurements, with the noise in the latter assumed *white*.

9.4.5 Filter Model for Autocorrelated Measurement Noise

9.4.5 Filter Model for Autocorrelated Measurement Noise

As shown in Subsection 9.3.3 the autocorrelated displacement measurement noise is the output of the following white noise driven subsystem

$$w_d^t(k) = \beta_1 v_2^t(k) + \beta_2 v_2^t(k) \quad (9.4.5-1)$$

Augmenting the target state equation with the above leads to a state equation with measurement noise and process noise sequences for the two targets that are white but the process noise $v^t(k)$ and the measurement noise $w^t(k)$ are correlated, as discussed in Subsection 9.3.3.

The augmented state equations for each target are, for one coordinate and $t = 1, 2$, given by (9.3.3-4) as

$$\begin{bmatrix} x_1^t(k+1) \\ x_2^t(k+1) \\ x_3^t(k+1) \\ x_4^t(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^t(k) \\ x_2^t(k) \\ x_3^t(k) \\ x_4^t(k) \end{bmatrix} + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} v_1^t(k) \\ v_2^t(k) \end{bmatrix} \quad (9.4.5-2)$$

and the measurement equation is

$$z^t(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1^t(k) \\ x_2^t(k) \\ x_3^t(k) \\ x_4^t(k) \end{bmatrix} + \begin{bmatrix} w_1^t(k) \\ w_2^t(k) \end{bmatrix} \quad t = 1, 2 \quad (9.4.5-3)$$

where $v_1^t(k)$ is the motion process noise, $v_2^t(k)$ is the white noise input to the colored noise in the displacement measurement from (9.4.5-1) with variance $(\sigma_d^t)^2$, $w_1^t(k)$ is the noise in the centroid measurement, and

$$w_2^t(k) \triangleq \beta_1 v_2^t(k) \quad (9.4.5-4)$$

Until the targets do not overlap the estimation is done separately for each target with individual filters as indicated earlier. When the targets overlap the filtering is done in a ***coupled manner*** as described below.

9.4.5 Filter Model for Autocorrelated Measurement Noise

The Modified State Equation

The measurement equation is

$$\mathbf{z}(k) = \begin{bmatrix} z_c(k) \\ z_d^1(k) \\ z_d^2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1^1(k) \\ x_2^1(k) \\ x_3^1(k) \\ x_4^1(k) \\ x_1^2(k) \\ x_2^2(k) \\ x_3^2(k) \\ x_4^2(k) \end{bmatrix} + \begin{bmatrix} w_c(k) \\ w_d^1(k) \\ w_d^2(k) \end{bmatrix}$$

$$\triangleq \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) \quad (9.4.5-5)$$

where the subscripts c and d , represent the centroid measurement and the centroid displacement measurements respectively.

The modified state equation that has the process noise uncorrelated from the measurement noise is

$$\mathbf{x}(k+1) = \mathbf{F}^*(k)\mathbf{x}(k) + \mathbf{u}^*(k) + \mathbf{v}^*(k) \quad (9.4.5-6)$$

where the stacked vector of the states of the two targets under consideration is as in (9.4.4-2); the modified (block-diagonal) state transition matrix is

$$\mathbf{F}^*(k) = \mathbf{F}(k) - \mathbf{T}(k)\mathbf{H}(k) \quad (9.4.5-7)$$

where \mathbf{H} is the measurement matrix from (9.4.5-5), and \mathbf{T} is the stacked matrix with blocks T^i given in (9.4.5-11).

The stacked input vector in (9.4.5-6) is

$$\mathbf{u}^*(k) = \begin{bmatrix} u^{1*}(k) \\ u^{2*}(k) \end{bmatrix} \quad (9.4.5-8)$$

and

$$\mathbf{v}^*(k) = \begin{bmatrix} v^{1*}(k) \\ v^{2*}(k) \end{bmatrix} = \begin{bmatrix} G^1 v^1(k) - T^1 w^1(k) \\ G^2 v^2(k) - T^2 w^2(k) \end{bmatrix} \quad (9.4.5-9)$$

9.4.5 Filter Model for Autocorrelated Measurement Noise

The elements of the stacked input vector in (9.4.5-8) are given by

$$u^{i*}(k) = T^i \sum_{\theta} P\{\theta|Z^k\} z_{j_i(\theta)}(k) \quad i = 1, 2; \quad j = 1, 2 \quad (9.4.5-10)$$

where the notation from (9.4.4-18) is used, and

$$T^i(k) = G^i(k)U^i(k)R^i(k)^{-1} \quad i = 1, 2 \quad (9.4.5-11)$$

and the block diagonal covariance matrix between the measurement and process noise is

$$\mathbf{U} = \begin{bmatrix} U^1 & 0 \\ 0 & U^2 \end{bmatrix} \quad (9.4.5-12)$$

where

$$U^i = E\{v^i(k)w(k)'\} \quad i = 1, 2 \quad (9.4.5-13)$$

The modified (block) diagonal process noise covariance matrix is

$$\mathbf{Q}^*(k) = \mathbf{G}(k)[\mathbf{Q}(k) - \mathbf{U}(k)\mathbf{R}(k)^{-1}\mathbf{U}(k)']\mathbf{G}(k)' \quad (9.4.5-14)$$

where

$$\mathbf{Q} = \begin{bmatrix} Q^1 & 0 \\ 0 & Q^2 \end{bmatrix} \quad (9.4.5-15)$$

The association probabilities for each target are evaluated in a similar manner as in Subsection 9.4.4.

The JPDAMCF for the state model (9.4.5-6) and measurement model (9.4.5-5) is implemented by a sequential updating procedure as discussed in Subsection 9.4.4, considering first the update with the centroid merged measurement and then the update with the ambiguous centroid displacement measurements.

9.4.6 Targets with Crossing Images — Simulation Results

9.4.6 Targets with Crossing Images — Simulation Results

The Scenario

Two targets with Gaussian plume images as in Subsection 9.3.4 and with motion described by second order kinematic models with white noise acceleration with variance $q = (0.05)^2$ in each of the two coordinates were considered.

The image of target 1 is moving with a heading of 135° with a nearly constant speed of one pixel per period, and image of target 2 is moving with a heading of 27° with a nearly constant speed of two pixels per period.

The sampling period is $T = 1$. The filters were initialized using two-point differencing and the initial state vectors of the targets in the frame were

$$\text{Target 1: } [-3.5 \ 1 \ -4.5 \ 3.5 \ -1 \ 4.5]'$$

$$\text{Target 2: } [-3.5 \ 1 \ -4.5 \ -8.5 \ 2 \ -10.5]'$$

The state of each target in each coordinate is as in (9.3.2-3).

The target state estimation was done for ten sampling times. The crossing of the target images takes place at times 3, 4 and 5. The centroid and the displacement measurement noises have variances $\sigma_c^2 = (0.195)^2$, $\sigma_d^2 = (0.104)^2$, respectively.

Filter with White Measurement Noise Model

The results of $N = 100$ Monte Carlo runs are shown for target 1 and target 2 in Tables 9.4.6-1 and 9.4.6-2, respectively.

The average normalized errors are seen to be within the bounds of a zero-mean Gaussian random variable with standard deviation $1/\sqrt{N} = 0.1$ and the values of the average NEES are also close to their theoretical value of 6 (it has a standard deviation of $\sqrt{2 \cdot 6N}/N = 0.35$).

The results indicate that the filters are consistent. The results are given only for the ξ coordinate but the NEES pertains to the entire state.

9.4.6 Targets with Crossing Images — Simulation Results

Time	Avg. pos.	Pos.	Avg. norm.	Avg. vel.	Vel.	Avg. norm.	Avg. NEES
k	error	s.d.	pos. error	error	s.d.	vel. error	(6 states)
0		0.195			0.276		
1	-0.004	0.14	-0.025	0.0086	0.088	0.098	5.90
2	0.013	0.13	0.098	0.006	0.07	0.09	5.96
3	0.017	0.122	0.136	0.007	0.066	0.11	6.33
4	-0.01	0.118	-0.11	0.01	0.064	0.156	6.10
5	0.002	0.118	0.016	-0.005	0.0639	-0.08	5.82
6	-0.0004	0.1175	-0.003	-0.0016	0.0637	-0.025	5.86
7	0.007	0.1175	0.062	0.003	0.0637	0.05	6.03
8	0.027	0.1175	0.23	0.011	0.0637	0.22	6.44
9	-0.004	0.1175	-0.03	-0.004	0.0637	-0.06	6.11

Table 9.4.6-1: Average errors in ξ direction and average NEES for the entire state for Target 1 (100 Monte Carlo runs).

Time	Avg. pos.	Pos.	Avg. norm.	Avg. vel.	Vel.	Avg. norm.	Avg. NEES
k	error	s.d.	pos. error	error	s.d.	vel. error	(6 states)
0		0.195			0.276		
1	-0.005	0.14	-0.032	0.016	0.088	0.19	6.63
2	-0.013	0.13	-0.1	0.005	0.07	0.07	6.05
3	-0.015	0.122	-0.123	-0.007	0.066	-0.1	5.85
4	0.009	0.118	0.07	0.007	0.064	0.11	5.57
5	-0.02	0.118	-0.173	-0.006	0.0639	-0.2	5.66
6	0.004	0.1175	0.003	0.015	0.0637	0.024	6.12
7	0.013	0.1175	0.113	0.013	0.0637	0.094	6.20
8	-0.008	0.1175	-0.07	-0.006	0.0637	-0.094	5.93
9	-0.01	0.1175	-0.09	-0.001	0.0637	-0.02	6.26

Table 9.4.6-2: Average errors in ξ direction and average NEES for the entire state for Target 2 (100 Monte Carlo runs).

9.4.6 Targets with Crossing Images — Simulation Results

Filter with Autocorrelated Measurement Noise Model

The simulations were repeated for the two targets with the displacement measurement noise modeled by the tracking filter as autocorrelated. The results of $N = 100$ Monte Carlo runs are shown for target 1 in Table 9.4.6-3.

The average normalized errors are again seen to be within the bounds of a zero-mean Gaussian random variable with standard deviation $1/\sqrt{N} = 0.1$ and the values of the average NEES are also close to their theoretical value of 8 (it has a standard deviation of $\sqrt{2 \cdot 8N}/N = 0.4$).

Time	Avg. pos. <i>k</i>	Pos. error	Avg. norm. s.d.	Avg. vel. pos. error	Vel. error	Avg. norm. s.d.	Avg. vel. error (8 states)	Avg. NEES
0			0.195			0.276		
1	-0.006	0.13		-0.047	0.005	0.07	0.072	9.14
2	-0.012	0.12		-0.105	-0.027	0.06	-0.033	8.06
3	-0.05	0.11		-0.048	-0.007	0.057	-0.122	7.99
4	-0.014	0.099		-0.142	0.005	0.056	0.080	8.46
5	-0.015	0.094		-0.159	0.002	0.056	0.034	8.40
6	-0.007	0.092		-0.076	0.004	0.056	0.079	8.06
7	-0.001	0.088		-0.012	0.001	0.056	0.107	7.95
8	-0.0001	0.088		-0.001	-0.0001	0.056	-0.002	8.33
9	-0.024	0.088		-0.027	-0.006	0.056	-0.107	8.35

Table 9.4.6-3: Average errors in ξ direction and average NEES for the entire state for Target 1 (100 Monte Carlo runs).

The results show that the filters are consistent and the best performance is achieved by incorporating the model of the autocorrelated measurement noise (cf. Table 9.4.6-3 vs. 9.4.6-1) and RMS tracking accuracies under a tenth of pixel are obtained.

9.4.7 Tracking of Targets with Overlapping Images — Summary

9.4.7 Tracking of Targets with Overlapping Images — Summary

The filtering of the following measurements pertaining to the centroids of the *overlapping images* of two extended targets has been discussed:

- the combined image centroid location in a single frame — a *merged measurement*
- the centroid displacements (offset) between two consecutive frames — two *ambiguous origin measurements*.

The algorithm that handles this situation is the Joint Probabilistic Data Association Merged-Measurement Coupled Filter (JPDAMCF).

The following possible filter designs have been considered:

- Design A: Filter using centroid and displacement measurements *assuming the displacement measurement noise white*. The state is 3-dimensional per coordinate per target (total dimension 12).
- Design B: Filter with centroid and displacement measurements, with the displacement measurement noise accounted for as autocorrelated. The state is 4-dimensional per coordinate per target (total dimension 16) and the filter requires prewhitening of the autocorrelated measurement noise and decorrelation of the two noise sequences.

Design A is less expensive since it has lower state dimension.

Design B is more expensive but it can achieve an MS error about 1.8 times smaller in position than Design A.

9.5 PRECISION TRACKING WITH SEGMENTATION FOR IMAGING SENSORS

9.5.1 Introduction

This section presents a method for precision target tracking based on data obtained from imaging sensors and characterized by the following:

- The size of the target image is of the order of tens of pixels — the image is too small to extract meaningful features; thus the goal is to estimate the motion of its **centroid**
- The target is not fully visible during tracking — the target pixel detection probability is significantly less than unity
- The background can be *stronger or weaker than the target*; the target is modeled as having an intensity distribution within a certain band and the background in a much wider range, both above and below the target's average intensity.

The Approach to Tracking the Image of Interest

- The entire image is segmented with the following steps:
 1. It is divided according to several layers of gray level intensities — target layer(s) and the rest background layers — and then binarized
 2. The binary image obtained is grouped into clusters
- The association of the various clusters to the track to be estimated is carried out using both motion and pattern recognition characteristics of the target
- Using the centroid measurements of the clusters, the PDAF is employed for state estimation.

Expressions for

- the (single frame based) **target cluster centroid variance** and
- the optimal **clustering** parameter

are given.

Simulation results validate the expressions for the centroid measurement noise variance and the performance predictions of the tracking method.

Applications

The method is first illustrated on a dim synthetic target occupying about 80 pixels within a 64×64 frame in the presence of a noise background which can be stronger than the target:

- The binary image obtained after an **intensity band-pass thresholding** is reduced to clusters by the **nearest neighbor clustering criterion**
- The results show a subpixel accuracy in the range of 0.3 to 0.4 pixel RMS error with moderate (0.7) to low (0.3) target pixel detection probability.

The usefulness of the method for practical applications is demonstrated by considering a sequence of real target images (a moving car with image of about 20 pixels, in an urban background) where

- The measurement noise variance was calculated as having 0.7 pixel RMS value
- The achieved filter accuracy for position was 0.4 pixel RMS in each coordinate and 0.09 pixel/frame for velocity after 10 frames.

9.5.2 Centroid of a Random Cluster

Centroid of an Image with Random Intensities

Consider a **cluster** of n pixels, denoted by a single index i , $i = 1, \dots, N$. If pixel i within this (target) cluster has intensity I_i , then the centroid of the cluster in the n -th coordinate is defined as

$$x_{n_c} = \frac{\sum_{i=1}^N x_{n_i} I_i}{\sum_{i=1}^N I_i} \quad (9.5.2-1)$$

where x_{n_i} is the n -th coordinate of point i .

Assuming the coordinate system origin located at the true centroid and modeling I_i as i.i.d. random variables with mean μ_i and standard deviation σ_i , the variance of the centroid (9.5.2-1) can be shown to be [OKB93]

$$\text{var}(x_{n_c}) = \frac{\sum_{i=1}^N x_{n_i}^2 \sigma_i^2}{(\sum_{i=1}^N \mu_i)^2} \quad (9.5.2-2)$$

Remark

In Subsection 9.2.3 the assumption was that there is a target with a certain *intensity pattern* observed in the presence of *additive noise*.

The target centroid was then computed from a *fixed tracking window* using the *actual intensity measurements*.

Here a *random target* is assumed with *background noise* — the part of the image where there is no target. This noise background has a different intensity distribution and will be accounted for later as perturbing the edge of the image portion segmented as the target.

The centroid calculation is based on the *target cluster* resulting from the segmentation and using the *binarized image*.

9.5.2 Centroid of a Random Cluster

Centroid of a Binary Image

The gray level image is transformed to a binary image, with new intensity β_i , by a hard limiter — **intensity band-pass thresholding**:

$$\beta_i = \begin{cases} 1 & I_L \leq I_i \leq I_H \\ 0 & \text{otherwise} \end{cases} \quad (9.5.2-3)$$

where I_L and I_H are the **target intensity band** limits.

The **detection probability of a pixel** by the threshold rule is

$$p_i \triangleq P\{\beta_i = 1\} = 1 - P\{\beta_i = 0\} \quad (9.5.2-4)$$

In other words, three layers of intensity are considered with the middle one — the target intensity band — used for target detection.

The mean and variance of the binary intensity of a single pixel are

$$\mu_i = E(\beta_i) = p_i \quad (9.5.2-5)$$

$$\sigma_i^2 = \text{var}(\beta_i) = p_i(1 - p_i) \quad (9.5.2-6)$$

For the binary image, the variance of the centroid (9.5.2-2) becomes

$$\text{var}(x_{n_c}) = \frac{\sum_{i=1}^N x_{n_i}^2 p_i(1 - p_i)}{(\sum_{i=1}^N p_i)^2} \quad (9.5.2-7)$$

If the pixel intensities are *identically distributed*, then

$$p_i = p \quad (9.5.2-8)$$

and

$$\text{var}(x_{n_c}) = \frac{(1 - p) \sum_{i=1}^N x_{n_i}^2}{N^2 p} \quad (9.5.2-9)$$

If the pixel intensities are also Gaussian

$$I_i \sim \mathcal{N}(\mu, \sigma^2) \quad (9.5.2-10)$$

then the pixel detection probability is

$$p = \frac{1}{\sqrt{2\pi}\sigma} \int_{I_L}^{I_H} e^{-\frac{(\zeta-\mu)^2}{2\sigma^2}} d\zeta \quad (9.5.2-11)$$

9.5.2 Centroid of a Random Cluster

Examples

Consider a target of arbitrary shape having a size of N_t pixels with i.i.d. pixel intensities $\mathcal{N}(\mu, \sigma^2)$.

1. A circular target

With a radius of R pixels, the sum in (9.5.2-9) is (by approximating it with an easy to evaluate integral)

$$\sum_{i=1}^N x_{n_i}^2 \approx 2 \int_0^R \int_0^\pi R^3 \cos^2 \theta d\theta dR = \frac{R^4 \pi}{4} \quad (9.5.2-12)$$

and the number of pixels in the circle is

$$N_t = \pi R^2 \quad (9.5.2-13)$$

Assuming that the set of N_t target pixels have been clustered, the variance (9.5.2-9) is then

$$\text{var}(x_{n_c}) = \frac{1-p}{4\pi p} \quad (9.5.2-14)$$

2. A rectangular target

With the sides of the rectangle denoted as a_n in the coordinate under consideration and a_m in the other coordinate, one has

$$\sum_{i=1}^N x_{n_i}^2 = \int_{-\frac{a_n}{2}}^{\frac{a_n}{2}} \int_{-\frac{a_m}{2}}^{\frac{a_m}{2}} \rho^2 d\rho d\tau = \frac{a_n^3 a_m}{12} \quad (9.5.2-15)$$

$$N_t = a_n a_m \quad (9.5.2-16)$$

and the variance is

$$\text{var}(x_{n_c}) = \frac{(1-p)a_n^2}{12pN_t} = \frac{(1-p)a_n}{12pa_m} \quad (9.5.2-17)$$

Remark

The variance expressions (9.5.2-14) and (9.5.2-17) assume that the set of target pixels is known. In practice this set, obtained via image segmentation, is imperfect and introduces *additional errors*, discussed next.

9.5.3 Clustering and the Centroid Variance

9.5.3 Clustering and the Centroid Variance

The image segmentation to be presented is a two step process:

- First the original gray scale image is transformed into a binary image by the hard limiter rule (9.5.2-3)
- Then the binary image is grouped into clusters.

A real-time implementable **clustering** algorithm was chosen: the **nearest neighbor clustering** (e.g., [JD88]). In this technique, a pixel belongs to the cluster if its distance to at least one other pixel of the cluster is less or equal to a certain **proximity distance** d_p .

If p_t and p_ν are the detection probabilities of the target and the noise pixels respectively, then the **average distance between neighboring pixels with detections** from the target and from the noise can be shown to be, respectively, (see problem 9-1)

$$d_t \approx \sqrt{1/p_t} \quad (9.5.3-1)$$

$$d_\nu \approx \sqrt{1/p_\nu} \quad (9.5.3-2)$$

in pixel units.

The proximity distance in the clustering affects the size, shape and number of the clusters obtained:

- If $d_p > d_t$ then most of the target pixels will be in the same cluster;
- If $d_p < d_\nu$, then most noise pixels will stay as separate clusters or will be grouped into small clusters.

The Optimal Proximity Distance for Clustering

As shown in [OKB93], based on extensive simulations, the optimal proximity distance that yields the smallest centroid variance is

$$d_p^* = \frac{d_t + d_\nu}{2} \quad (9.5.3-3)$$

Random Target with Noise Background

The effect of the noise background will be a small number of *noise pixel clusters attached to the edge of the target cluster*, which will *increase its centroid variance*.

To evaluate this, consider a target of size (area) N_t pixels surrounded by a “ring” of N_ν noise pixels around it, with pixel detection probabilities p_t for the target and p_ν for the ring area (due to noise), respectively. The intensity of the pixels is given by

$$I_i = \begin{cases} s_i & i \in \mathbf{T} \\ \nu_i & i \in \mathbf{N} \end{cases} \quad (9.5.3-4)$$

where s_i and ν_i denote the target and noise pixel intensities, while \mathbf{T} and \mathbf{N} are the sets of target and noise pixels, respectively.

Then the variance of the centroid is, using (9.5.2-9)

$$\text{var}(x_{n_c}) = \frac{p_t(1 - p_t) \sum_{i \in \mathbf{T}} x_{n_i}^2 + p_\nu(1 - p_\nu) \sum_{i \in \mathbf{N}} x_{n_i}^2}{(p_t N_t + p_\nu N_\nu)^2} \quad (9.5.3-5)$$

Assuming that the target is circular with radius r_t in a noise ring of width d_r (derived in [KB93]), the number of (noise) pixels in this ring is then, from (9.5.2-13)

$$N_\nu = \pi[(r_t + d_r)^2 - r_t^2] \quad (9.5.3-6)$$

Using (9.5.2-12) and (9.5.3-6) in (9.5.3-5) yields the **target centroid variance with the effect of noise background** as

$$\text{var}(x_{n_c}) = \frac{p_t(1 - p_t)r_t^4 + p_\nu(1 - p_\nu)[(r_t + d_r)^4 - r_t^4]}{4\pi[p_t r_t^2 + p_\nu(r_t + d_r)^2 - p_\nu r_t^2]^2} \quad (9.5.3-7)$$

9.5.4 The State Estimation Model

9.5.4 The State Estimation Model

The state equation for tracking a nearly constant velocity target in two dimensions, using the centroid measurements is

$$x(k+1) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} v(k) \quad (9.5.4-1)$$

where the state consists of the current position and velocity in two Cartesian coordinates; T is the sampling period and $v(k)$ is the zero mean, white process noise (acceleration) with variance q_x and q_y in the two Cartesian coordinates. The change in velocity over the sampling period is of the order of $\sqrt{q}T$.

The centroid position measurement at time k is

$$z(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(k) + w(k) \quad (9.5.4-2)$$

where $w(k)$ is the centroid measurement noise.

The covariance of the measurement noise is

$$R = \text{diag}(\sigma_x^2, \sigma_y^2) \quad (9.5.4-3)$$

where the measurement noise variances (same for each coordinate) are given by (9.5.3-7).

The measurement and process noise sequences are uncorrelated. The Probabilistic Data Association Filter (PDAF) was used to obtain the state estimates.

9.5.5 Precision Tracking of a Synthetic Image

9.5.5 Precision Tracking of a Synthetic Image

A two-dimensional array of 64×64 pixels with a 256 level gray scale is considered for the image.

Modeling of the Image

The target is modeled as a white Gaussian random field with mean μ_t and variance σ_t^2 . The background (noise) is also modeled as a white Gaussian **random field** (a random process in two dimensions) with moments μ_ν and σ_ν^2 .

The Segmentation

The image is divided into three layers of gray level intensities with the target assumed to be in the middle layer. With the target layer being the interval $[I_L, I_H]$, the image is binarized.

The binary image is then grouped into clusters by the nearest neighbor technique using the optimal proximity distance d_p^* . All the clusters of size less than I_{\min} are ignored.

Measurement Extraction and Tracking

The centroid of each cluster is calculated and used for processing by the PDAF. For 1σ threshold ($I_L = \mu_t - \sigma_t$, $I_H = \mu_t + \sigma_t$) there are about 15 clusters in the image. But within the validation gate of the PDAF, a maximum of 3 clusters appear. The PDAF probabilistically weights the different measurements (centroids) in the validation gate.

The sampling (scan) period was $T = 1$ and the target motion was modeled by the second order kinematic model (9.5.4-1).

The gray scale image with noise for a single scan is shown in Figure 9.5.5-1 for a circular target of radius 5 pixels.

The binary image is shown in Figure 9.5.5-2.

9.5.5 Precision Tracking of a Synthetic Image

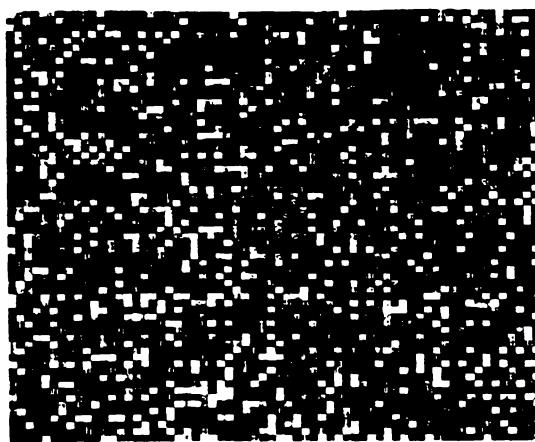


Figure 9.5.5-1: A gray scale image with noise for a single scan.

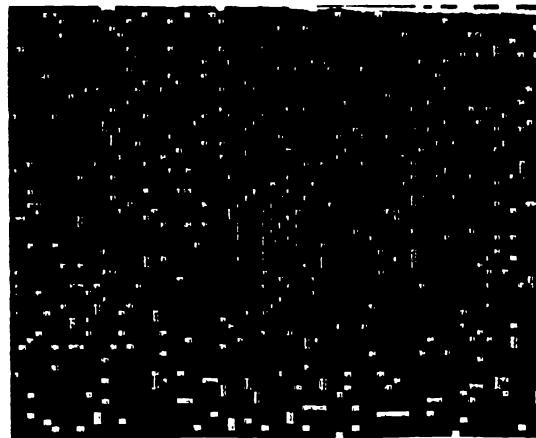


Figure 9.5.5-2: The binary image for a single scan.

9.5.5 Precision Tracking of a Synthetic Image

Numerical Results

Tracking results for a circular target of 78 pixels (radius of 5 pixels — shown in Figure 9.5.5-1) in a 64×64 frame are presented for 20 scans for two cases of target pixel detection probabilities.

The initial state vector of the target in the image (pixel) frame was

$$\mathbf{x}(0) = [11.0 \ 1.5 \ 15.0 \ 1.0]^\top \quad (9.5.5-1)$$

The target and noise parameters used in the simulations were

$$\text{Target pixel intensity: } \mathcal{N}(100, 10^2) \quad (9.5.5-2)$$

$$\text{Noise pixel intensity: } \mathcal{N}(50, 50^2) \quad (9.5.5-3)$$

The threshold levels and detection probabilities were as in Table 9.5.5-1.

Case	I_L	I_H	p_t	p_ν
1	90	110	0.682	0.097
2	95	105	0.383	0.048

Table 9.5.5-1: Threshold levels and detection probabilities for the two cases.

The expected target sizes for Cases 1 and 2 were 61 and 34, respectively. The minimum cluster size I_{\min} for both cases was chosen as 10.

Based on (9.5.3-3), the optimal clustering proximity distances in the two cases were $d_p^* = 2$ and $d_p^* = 3$, respectively.

The PDAF from MULTIDAT™ was used with the following parameters:

- The measurement noise had a variance of 0.33 and 0.76 for Cases 1 and 2, respectively
- The target cluster detection probability was taken as 0.97
- A low level target process noise $q_x = q_y = 10^{-4}$ was assumed by the filter.

9.5.5 Precision Tracking of a Synthetic Image

Table 9.5.5-2 summarizes the results of 100 Monte Carlo runs in terms of achieved accuracies for the problem considered. The variances are given only for the horizontal coordinate but the NEES pertains to the entire four dimensional state.

The last two columns of Table 9.5.5-2 confirm that the NEES is within the 95% probability region. This indicates that the filter is consistent, i.e., the calculated variances match the actual errors.

Case	Position variance	Velocity variance	Average NEES	95% region for NEES
1	0.0632	0.0010	4.10	[3.5, 4.5]
2	0.1367	0.0016	3.79	[3.5, 4.5]

Table 9.5.5-2: Position and velocity variances (in steady state) for each coordinate and filter consistency verification from 100 runs.

The results show that it is possible to achieve subpixel accuracy in the range of 0.35 pixels RMS overall position error for Case 1 and 0.52 pixels for Case 2.

The target layer threshold settings (I_L, I_H) in Case 1 clearly yield superior results compared to Case 2 for the problem considered.

These results show that one can obtain very good subpixel accuracy for a target of this size, even though it is not fully visible most of the time.

9.5.6 Precision Tracking of a Real Target

9.5.6 Precision Tracking of a Real Target

This method was used to track a moving car in a sequence of real images.

The Images

The real images were obtained by an FPA (Focal Plane Array) based platinum-silicide camera (256×256 pixels) in infrared light ($3\text{--}5 \mu\text{m}$ wavelength). The 256×256 images were reduced to 64×64 . The latter were used for segmentation and centroid calculation and these centroid measurements were then passed on to the tracking filter.

The images show two cars moving in opposite directions crossing each other during daytime. The car moving from left to right (with an unknown velocity) was chosen for tracking. Two of the ten frames are shown in Figure 9.5.6-1 with the target clusters indicated.

The Target and Noise Statistics

The target and noise parameters were obtained by considering a single frame of the image:

- The target size was found to be about 20 pixels with intensity mean of about 150 and variance 90
- The noise background statistics were found from the entire image to have a mean of 120 and variance 100.

Remark

While it is clear that the white random field assumption about the background does not hold, the algorithm worked remarkably well.

9.5.6 Precision Tracking of a Real Target

The Segmentation

The image at each scan was divided into three layers of gray level intensities around the average target intensity (150). The gray scale image was then binarized using $\pm 1\sigma$ threshold levels (140 to 160).

These values lead to target and noise pixel detection probabilities of 0.7 and 0.03, respectively.

The image parameters are summarized in Table 9.5.6-1.

μ_t	σ_t	μ_ν	σ_ν	I_L	I_H	p_t	p_ν	d_p^*	$\text{var}(x_{nc})$
150	9.5	120	10	140	160	0.70	0.03	3.53	0.46

Table 9.5.6-1: Image parameters used in the real target tracking example.

With the above parameters, the optimal proximity distance is $d_p^* = 3.5$.

Based on the target size of 20 pixels, the equivalent radius is 2.5 pixels. From (9.5.3-7) the single frame based centroid measurement noise variance is 0.46.

As the target size was relatively small, the minimum cluster size I_{\min} was chosen as 1. On the average there are about 10 clusters in each scan and a maximum of 3 clusters in the validation gate of the PDAF.

Using the PDAF, the estimated values of the centroid position and velocity of the target, along with the filter-calculated accuracies are given in Figures 9.5.6-2 and 9.5.6-3. The filter-calculated RMS errors were 0.4 pixel (variance 0.16 — a reduction by a factor of almost 3 from the raw measurements) for each coordinate in position and 0.09 pixel/frame in velocity.

The position accuracy amounts to about one tenth of the target size.

Filter Consistency

Since the ground truth is not known, in order to check the filter's consistency, the 10-sample time-averaged Normalized Innovation Squared (NIS) was calculated and was found to be 1.16, which is within the chi-square limits [0.96, 3.14] based on $\chi^2_{20}/10$. Thus the filter-calculated accuracies shown in Figures 9.5.6-2 and 9.5.6-3 are remarkably reliable.

These results show that, in spite of the non-white and non-Gaussian noise in the image, the method was successful for this real image.

9.5.6 Precision Tracking of a Real Target

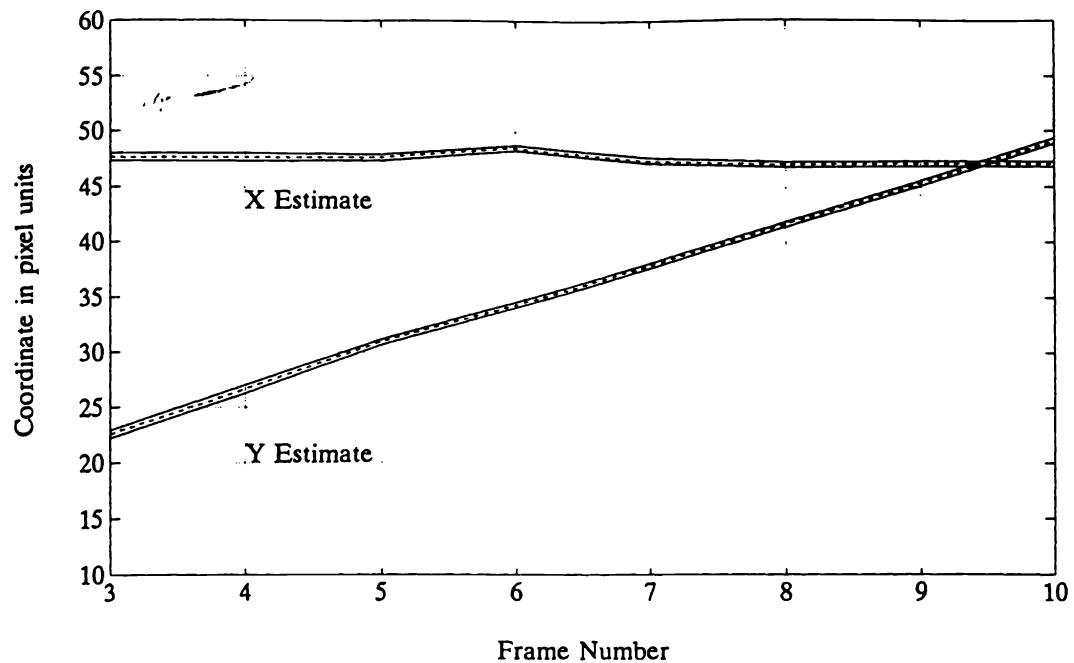


Figure 9.5.6-2: The estimated centroid position along with $\pm 1\sigma$ accuracy.

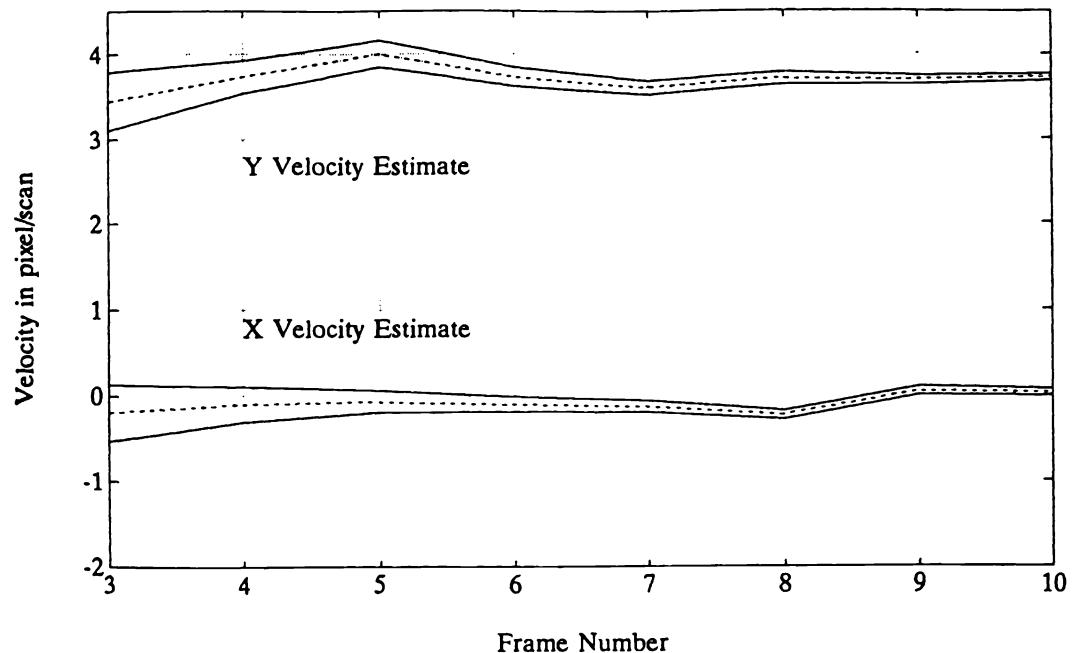


Figure 9.5.6-3: The estimated centroid velocity along with $\pm 1\sigma$ accuracy.

9.5.7 Small Image Precision Tracking — Summary

9.5.7 Small Image Precision Tracking — Summary

A method for precision tracking of targets using data obtained from imaging sensors was presented.

A closed-form single frame based target centroid variance was presented in terms of the target image and noise characteristics. This *measurement noise variance* is an important tracking filter design parameter.

Using image segmentation techniques, the gray scale image is thresholded and the resulting binary image is reduced to clusters by the nearest neighbor criterion. An expression for the *optimal proximity distance* needed for image segmentation was given.

The association of the clusters to the track was done by exploiting both the motion and pattern (object) recognition characteristics of the target. Using the centroid measurements of the clusters, the PDAF was employed for state estimation. The results for a target of about 78 pixels in a 64×64 frame show that it is possible to achieve subpixel tracking accuracy of 0.35 pixel overall RMS position error.

The usefulness of the proposed method was demonstrated by also considering a sequence of real target images, where for a 20 pixel target, the tracking accuracy obtained was 0.4 pixel RMS for each position coordinate and 0.09 pixel/frame for velocity.

Remark

The results shown were obtained with the software **IMDATTM**.

9.6 NOTES AND PROBLEMS

9.6.1 Bibliographical Notes

Sections 9.2 through 9.4 are based on Chapter 5 of [Bar92].

Section 9.5 is based on [OKB93]. A modified version of this algorithm has been implemented successfully in the Arrow ABM. The refinement of this technique, that includes optimization of the target layer based on the analysis of the image for improved segmentation and a more accurate expression of the width of the noise ring surrounding the segmented target, can be found in [KBO93]. The use of the output of an image processor that provides the orientation of an aircraft for rapid maneuver detection to enhance a tracking filter is discussed in [SHK93].

9.6.2 Problems

9-1 **Average distance between neighboring pixels with detections.** Prove (9.5.3-1).

Hint: Use an averaging argument so that the number of detections in N pixels is Np .

PROBLEM SOLUTIONS

CHAPTER 2

2-1 Because of symmetry, the “association region” will be a circle centered at the first measurement. The radius of the circle can be taken as

$$r = v_{\max}T + d(1 - \alpha)$$

where $d(1 - \alpha)$ is the largest distance between two measurements in the absence of motion, i.e., due to the measurement noises only, with some probability $1 - \alpha$. The distance (squared) is, in terms of the measurement noises $w_i(t)$, $i = 1, 2$; $t = 0, T$

$$d^2 = [w_1(0) - w_1(T)]^2 + [w_2(0) - w_2(T)]^2 \sim 2\sigma^2\chi_2^2$$

Using the 1% tail, $\chi_2^2(99\%) = 9.2$ yields

$$d(99\%) = \sqrt{18.4}\sigma = 4.3\sigma$$

The exact solution requires a pdf for the velocity (which is not given) and derivation of the pdf of the distance between the measurements. The above simple calculation is conservative — the actual region will be smaller.

CHAPTER 3

3-1 1) $S(1) = H(1)P(1|0)H(1) + R(1) = 1.$

2) $\mathcal{V} = \{z : (z - 1)^2 \leq \gamma\}$; with $\gamma = 6.635$ the gate is $-1.576 \leq z \leq 3.576$.

3) -1 and 2 are validated; 4 is not.

4)

$$\hat{x}(1|1) = \hat{x}(1|0) + W(1)\nu(1) = 1 + (2/3)0.3377 = 1.225$$

$$P(1|1) = \beta_0(1)P(1|0) + [1 - \beta_0(1)]P^c(1|1) + \tilde{P}(1) = 0.797$$

5) For 2): $-3 \leq z \leq 5$; for 3): all; for 4): $\hat{x}(1|1) = 1.244$, $P(1|1) = 0.827$.

6) For 3): all but the first two are validated; for 4): $\hat{x}(1|1) = 1.05479$; $P(1|1) = 0.6987$.

The reason the variance is smaller in spite of having more measurements, is that there is one measurement exactly in the center of the validation region.

3-2 1) $P_G = P\{\xi \leq \gamma\} = \int_0^\gamma p(\xi)d\xi = 1 - e^{-\frac{\gamma}{2}}$.

2) Denoting by $n[0, x]$ the number of points from the Poisson process in the corresponding interval, one has

$$\begin{aligned} P\{M\} &= P\{M|\xi > \gamma\}P\{\xi > \gamma\} + P\{M|\xi \leq \gamma\}P\{\xi \leq \gamma\} \\ &= P\{n[0, \gamma] > 0\}(1 - P_G) + \int_0^\gamma P\{M|\xi\}p(\xi|\xi \leq \gamma)d\xi P_G \\ &= (1 - e^{-\lambda\gamma})e^{-\frac{\gamma}{2}} + \int_0^\gamma P\{n[0, \xi] > 0\}\frac{1}{P_G}p(\xi)d\xi P_G \\ &= (1 - e^{-\lambda\gamma})e^{-\frac{\gamma}{2}} + \int_0^\gamma (1 - e^{-\lambda\xi})\frac{1}{2}e^{-\frac{\xi}{2}}d\xi \\ &= (1 - e^{-\lambda\gamma})e^{-\frac{\gamma}{2}} + 1 - e^{-\frac{\gamma}{2}} - \frac{1}{2}\int_0^\gamma e^{-(\lambda+\frac{1}{2})\xi}d\xi \\ &= 1 - e^{-(\lambda+\frac{1}{2})\gamma} - \frac{1}{2\lambda+1}[1 - e^{-(\lambda+\frac{1}{2})\gamma}] = \frac{2\lambda}{2\lambda+1}[1 - e^{-(\lambda+\frac{1}{2})\gamma}] \end{aligned}$$

3)

$$\begin{aligned} P\{M'\} &= P\{M'|D\}P_D + P\{M'|\bar{D}\}(1 - P_D) \\ &= P\{M\}P_D + P\{n[0, \gamma] > 0\}(1 - P_D) \\ &= \frac{2\lambda}{2\lambda+1}[1 - e^{-(\lambda+\frac{1}{2})\gamma}]P_D + (1 - e^{-\lambda\gamma})(1 - P_D) \end{aligned}$$

4)

$$\begin{aligned} P\{A\} &= \int_0^\gamma P\{A|\xi\}p(\xi)d\xi P_D = \int_0^\gamma P\{n[0, \xi] = 0\}p(\xi)d\xi P_D \\ &= \int_0^\gamma e^{-\lambda\xi}\frac{1}{2}e^{-\frac{\xi}{2}}d\xi P_D = \frac{1}{2\lambda+1}[1 - e^{-(\lambda+\frac{1}{2})\gamma}]P_D \end{aligned}$$

Problem Solutions

5)

$$\begin{aligned} P\{\bar{A}\} &= P\{\bar{A}|D\}P_D + P\{\bar{A}|\bar{D}\}(1 - P_D) \\ &= P\{n[0, \gamma] = 0\}(1 - P_G)P_D + P\{n[0, \gamma] = 0\}(1 - P_D) \\ &= e^{-\lambda\gamma}[e^{-\frac{\gamma}{2}}P_D + 1 - P_D] \end{aligned}$$

6)

$$P\{M'\} + P\{A\} + P\{\bar{A}\} = 1$$

For a more complete treatment of the nearest neighbor filters, see [LB94d].

CHAPTER 4

- 4-1 1) $\max \tilde{\beta} = 1$; $\min \tilde{\beta} = 1/[m(k) + 1]$ when $\beta_i = 1/m(k)$, $\forall i$.
2) Minimum (zero) uncertainty; maximum uncertainty.

CHAPTER 8

8-1 Initial conditions for filters: $\hat{x}(0|0)$, $P^m(0|0) = p_0$, $m = i, j$.

- 1) One step predictions $\hat{x}^m(1|0) = \hat{x}^m(0|0)$, $P^m(1|0) = p_0 + q$

Innovation variances $S^m(1) = p_0 + q + r$

Filter gains $W^m(1) = \frac{p_0 + q}{p_0 + q + r}$

Updated states (in decentralized or local filters)

$$\begin{aligned}\hat{x}^m(1|1) &= \hat{x}^m(0|0) + \frac{p_0 + q}{p_0 + q + r} [z^m(1) - \hat{x}^m(0|0)] \\ &= \frac{r}{p_0 + q + r} \hat{x}^m(0|0) + \frac{p_0 + q}{p_0 + q + r} z^m(1)\end{aligned}$$

with variance

$$P^m(1|1) = \left(1 - \frac{p_0 + q}{p_0 + q + r}\right) (p_0 + q) = \frac{r(p_0 + q)}{p_0 + q + r} \quad m = i, j$$

- 2) The (cross-)covariance between the two updated states

$$P^{ij}(1|1) = \left(1 - \frac{p_0 + q}{p_0 + q + r}\right)^2 q = \frac{r^2 q}{(p_0 + q + r)^2}$$

- 3) Eq. (8.4.4-4) reduces, due to symmetry, to

$$\begin{aligned}\hat{x}(1|1) &= \frac{1}{2} [\hat{x}^i(1|1) + \hat{x}^j(1|1)] \\ &= \frac{r}{2(p_0 + q + r)} [\hat{x}^i(0|0) + \hat{x}^j(0|0)] + \frac{p_0 + q}{2(p_0 + q + r)} [z^i(1) + z^j(1)]\end{aligned}$$

Eq. (8.4.4-5) reduces to

$$P(1|1) = \frac{1}{2} [P^i(1|1) + P^{ij}(1|1)] = \frac{r(p_0 + q)(p_0 + q + r) + r^2 q}{2(p_0 + q + r)^2}$$

4)

$$\hat{x}(0|0) = \frac{1}{2} [\hat{x}^i(0|0) + \hat{x}^j(0|0)] \quad P(0|0) = \frac{1}{2} p_0$$

$$\hat{x}(1|0) = \hat{x}(0|0) \quad P(1|0) = \frac{1}{2} p_0 + q$$

The fused measurement

$$z(1) = \frac{1}{2} [z^i(1) + z^j(1)]$$

Problem Solutions

has noise

$$w(1) = \frac{1}{2}[w^i(1) + w^j(1)]$$

with variance $R = \frac{1}{2}r$.

The innovation variance is

$$S(1) = \frac{1}{2}p_0 + q + \frac{1}{2}r$$

and the centralized filter's gain is

$$W(1) = \frac{\frac{1}{2}p_0 + q}{\frac{1}{2}p_0 + q + \frac{1}{2}r}$$

The centralized estimate is

$$\begin{aligned}\hat{x}(1|1) &= \frac{1}{2}[\hat{x}^i(0|0) + \hat{x}^j(0|0)] \\ &\quad + \frac{\frac{1}{2}p_0 + q}{\frac{1}{2}p_0 + q + \frac{1}{2}r} \left[\frac{1}{2}[z^i(1) + z^j(1)] - \frac{1}{2}[\hat{x}^i(0|0) + \hat{x}^j(0|0)] \right] \\ &= \frac{r}{2(p_0 + 2q + r)}[\hat{x}^i(0|0) + \hat{x}^j(0|0)] + \frac{p_0 + 2q}{2(p_0 + 2q + r)}[z^i(1) + z^j(1)]\end{aligned}$$

with variance

$$P(1|1) = \frac{\left(\frac{1}{2}p_0 + q\right)\frac{1}{2}r}{\frac{1}{2}p_0 + q + \frac{1}{2}r} = \frac{(p_0 + 2q)r}{2(p_0 + 2q + r)}$$

- 5) The weightings of the initial estimates are smaller in the centralized estimate than in the fused one while for the measurements the opposite holds.
- 6) The MSE for the fused estimate is $P(1|1) = 7/18$. For the centralized estimate $P(1|1) = 3/8$. There is an improvement by a factor of $27/28$ in the latter, i.e., a decrease of about 4%.
- 7) In steady state

$$P^m(k|k) = p^m \quad P(k+1|k) = p^m + q \quad W = \frac{p^m + q}{p^m + q + r}$$

and the Riccati equation

$$p^m = \left(1 - \frac{p^m + q}{p^m + q + r}\right)(p^m + q) = \frac{r(p^m + q)}{p^m + q + r}$$

yields

$$(p^m)^2 + pq - rq = 0$$

Problem Solutions

with solution

$$\hat{p}^m = \frac{-1}{2} \left(-q + \sqrt{q^2 + 4rq} \right) = \frac{1}{2}r \left(\sqrt{\frac{q^2}{r^2} + \frac{4q}{r}} - \frac{q}{r} \right) = \frac{1}{2}r \left(\sqrt{\alpha^2 + 4\alpha} - \alpha \right)$$

where $\alpha \triangleq q/r$ is the target maneuvering index (this is the “ α -filter”).

8) In steady state $P^{ij}(k|k) = p^{ij}$ and is obtained from

$$p^{ij} = \left(1 - \frac{p^m + q}{p^m + q + r} \right)^2 (p^{ij} + q) = \frac{r^2}{(p^m + q + r)^2} (p^{ij} + q)$$

as

$$p^{ij} = \frac{r^2 q}{(p^m + q + r)^2}$$

9) With $P(k|k) = m^{ij}$ following 3) above one has

$$m^{ij} = \frac{1}{2}(p^i + p^{ij})$$

10) The centralized filter's covariance follows from 7) with $r \rightarrow \frac{r}{2}$ as

$$p = \frac{1}{2} \left(-q + \sqrt{q^2 + 2rq} \right)$$

11)

$$p^i = \frac{1}{2} (\sqrt{5} - 1) = 0.618$$

$$p^{ij} = \frac{1}{8} = 0.125 \quad m^{ij} = 0.372$$

$$p = \frac{1}{2} (\sqrt{3} - 1) = 0.366 \approx 0.985m^{ij}$$

The centralized estimator's steady state variance is about 1.5% smaller than the MSE resulting from the fusion of the decentralized estimates.

CHAPTER 9

9-1 A pixel is taken as the unit area with dimension 1×1 . The expected number of detections in N pixels, which cover an area N , is

$$Np = \bar{N}$$

Using an averaging argument, it will be assumed that the detections are equally spaced apart in both coordinates, at distances d in each coordinate. Then the area N is covered by \bar{N} detections, with the area covered by each detection being d^2 , i.e.,

$$N = \bar{N}d^2$$

The above equations yield immediately

$$d = \sqrt{\frac{1}{p}}$$

Using a similar argument with a hexagonal fill of the area (instead of square — this yields equidistant detections) one obtains

$$d = \sqrt{\frac{4}{3p}}$$

i.e., almost the same result.

Bibliography

- [ABEM83] J. Arnold, Y. Bar-Shalom, R. Estrada, and R. Mucci. Target Parameter Estimation Using Measurements Acquired with a Small Number of Sensors. *IEEE Journal of Oceanic Engrg.*, OE-8:163–172, July 1983.
- [ABM84] J. Arnold, Y. Bar-Shalom, and R. Mucci. Track Segment Association with a Distributed Field of Sensors. In *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.
- [AH83] V. J. Aidala and S. E. Hammel. Utilization of Modified Polar Coordinates for Bearings-Only Tracking. *IEEE Trans. Automatic Control*, AC-28:283–293, March 1983.
- [AIK91a] A. Averbuch, S. Itzikowitz, and T. Kapon. Parallel Implementation of Multiple Model Tracking Algorithms. *IEEE Trans. Parallel and Distributed Syst.*, PDS-2(2):242–252, Apr. 1991.
- [AIK91b] A. Averbuch, S. Itzikowitz, and T. Kapon. Radar Target Tracking — Viterbi versus IMM. *IEEE Trans. Aerospace and Electronic Systems*, AES-27(3):550–563, May 1991.
- [AL79] D. L. Alspach and R. N. Lobbia. A Score for Correct Data Association in Multitarget Tracking. In *Proc. 18th IEEE Conf. Decision and Control*, Ft. Lauderdale, FL, Dec. 1979.
- [Als75] D. L. Alspach. A Gaussian Sum Approach to the Multi-Target Identification-Tracking Problem. *Automatica*, 11:285–296, May 1975.
- [AM79] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [And91] D. Andrisani . A Nonlinear Helicopter Tracker Using Attitude Measurements. *IEEE Trans. Aerospace and Electronic Systems*, AES-27:40–47, Jan. 1991.
- [AR90] D. Avitzour and S. R. Rogers. Optimal Measurement Scheduling for Prediction and Estimation. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-38(10):1733–1739, Oct. 1990.
- [AWG77] M. Athans, R. H. Whiting, and M. Gruber. A Suboptimal Estimation Algorithm with Probabilistic Editing for False Measurements with Applications to Target Tracking with Wake Phenomena. *IEEE Trans. Automatic Control*, AC-22:372–384, June 1977.
- [Bar74] Y. Bar-Shalom. Extension of the Probabilistic Data Association Filter to Multitarget Environments. In *Proc. 5th Symp. Nonlinear Estimation*, San Diego, CA, Sept. 1974.
- [Bar81] Y. Bar-Shalom. On the Track-to-Track Correlation Problem. *IEEE Trans. Automatic Control*, AC-26:571–572, Apr. 1981.
- [Bar85] Y. Barniv. Dynamic Programming Solution to Detecting Dim Moving Targets. *IEEE Trans. Aerospace and Electronic Systems*, AES-21:144–156, Jan. 1985.

BIBLIOGRAPHY

- [Bar86] Y. Bar-Shalom. Comments on ‘Track Biases and Coalescence with PDA’. *IEEE Trans. Aerospace and Electronic Systems*, AES-22:661, Sept. 1986.
- [Bar88] Y. Bar-Shalom. Comments on Comparison of Two-Sensor Tracking Methods Based on State Vector Fusion and Measurement Fusion. *IEEE Trans. Aerospace and Electronic Systems*, AES-24:456–457, July 1988.
- [Bar90] Y. Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Advanced Applications*. Artech House, Norwood, MA, 1990.
- [Bar92] Y. Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Applications and Advances*. Volume II, Artech House, Norwood, MA, 1992.
- [BB82] Y. Bar-Shalom and K. Birmiwal. Variable Dimension Filter for Maneuvering Target Tracking. *IEEE Trans. Aerospace and Electronic Systems*, AES-18(5):621–629, Sept. 1982.
- [BB83] Y. Bar-Shalom and K. Birmiwal. Consistency and Robustness of PDAF for Target Tracking in Cluttered Environments. *Automatica*, 19:431–437, July 1983.
- [BB89] I. P. Bottlik and S. S. Blackman. Coordinated Presentation of Multiple Hypotheses in Multi-target Tracking. In *Proc. 1989 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 1096, pages 152–157, March 1989.
- [BC86a] Y. Bar-Shalom and L. Campo. The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance. *IEEE Trans. Aerospace and Electronic Systems*, AES-22:803–805, Nov. 1986.
- [BC86b] T. J. Broida and R. Chellappa. Estimation of Object Motion Parameters from Noisy Images. *IEEE Trans. Patt. Anal. and Mach. Intellig.*, PAMI-8:90–99, Jan. 1986.
- [BCB89] Y. Bar-Shalom, K. C. Chang, and H. A. P. Blom. Tracking a Maneuvering Target Using Input Estimation vs. the Interacting Multiple Model Algorithm. *IEEE Trans. Aerospace and Electronic Systs*, AES-25:296–300, March 1989.
- [BCB92] Y. Bar-Shalom, K. C. Chang, and H. A. P. Blom. Tracking Splitting Targets in Clutter by Using an Interacting Multiple Model Joint Probabilistic Data Association filter. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Applications and Advances*. Vol. II, chapter 4, Artech House, Norwood, MA, 1992.
- [BCL90] Y. Bar-Shalom, L. Campo, and P. B. Luh. From Receiver Operating Characteristic to System Operating Characteristic. *IEEE Trans. Automatic Control*, AC-35(2):172–179, Feb. 1990.
- [BCS89] Y. Bar-Shalom, K. C. Chang, and H. M. Shertukde. Performance Evaluation of a Cascaded Logic for Track Formation in Clutter. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:873–878, Nov. 1989.
- [BDP90] Y. Bar-Shalom, S. Deb, and K. R. Pattipati. Passdat: Passive Sensor Data Association for Tracking. In *Proc. SPIE Conf. on Signal and Data Processing for Small Targets*, Orlando, FL, March 1990.
- [BF88] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1988.
- [BFS80] Y. Bar-Shalom, T. E. Fortmann, and M. Scheffe. Joint Probabilistic Data Association for Multiple Targets in Clutter. In *Proc. 1980 Conf. Information Sciences and Systems*, Princeton, NJ, March 1980.
- [BGBD94] W. D. Blair, G. W. Groves, Y. Bar-Shalom, and E. Daeipour. Frequency Agility and Fusion for Tracking Targets in the Presence of Multipath Propagation. In *Proc. 1994 National Radar Conf.*, Atlanta, GA, Mar. 1994.

BIBLIOGRAPHY

- [BHv90] H. A. P. Blom, R. A. Hogendoorn, and F. J. van Schaik. Bayesian Multisensor Tracking for Advanced Air Traffic Control Systems. In A. Benoit, editor, *Aircraft Trajectories: Computation, Prediction and Control*, 1990. AGARDograph 301.
- [BJ72] Y. Bar-Shalom and A. Jaffer. Adaptive Nonlinear Filtering for Tracking with Measurements of Uncertain Origin. In *Proc. 1972 IEEE Conf. Decision and Control*, pages 243–247, New Orleans, LA, Dec. 1972.
- [BKBG94] Y. Bar-Shalom, A. K. Kumar, W. D. Blair, and G. W. Groves. Tracking Low Elevation Targets in the Presence of Multipath Propagation. *IEEE Trans. Aerospace and Electronic Systems*, AES-30(4), Oct. 1994.
- [BL71a] F. Bourgeois and J. C. Lassalle. Algorithm for the Assignment Problem. *Comm. of the ACM*, 14:805–806, Dec. 1971.
- [BL71b] F. Bourgeois and J. C. Lassalle. An Extension of the Numbers Algorithm for the Assignment Problem to Rectangular Matrices. *Comm. of the ACM*, 14:802–804, Dec. 1971.
- [BL91] Y. Bar-Shalom and X. R. Li. Effectiveness of the Likelihood Function in Logic-Based Track Formation. *IEEE Trans. Aerospace and Electronic Systems*, AES-27:184–187, Jan. 1991.
- [BL93] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Boston, MA, 1993.
- [Bla86] S. S. Blackman. *Multiple Target Tracking with Radar Applications*. Artech House, Norwood, MA, 1986.
- [Bla94] W. D. Blair. Toward the Integration of Tracking and Signal Processing for Phased Array Radar. In *Proc. 1994 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 2235, Orlando, FL, Apr. 1994.
- [Blo84] H. A. P. Blom. A Sophisticated Tracking Algorithm for ATC Surveillance Data. In *Proc. Int. Radar Conf.*, Paris, France, May 1984.
- [BM80] Y. Bar-Shalom and G. D. Marcus. Tracking with Measurements of Uncertain Origin and Random Arrival. *IEEE Trans. Automatic Control*, AC-25:802–807, Aug. 1980.
- [BM81] C. L. Bowman and M. S. Murphy. An Architecture for Fusion of Multisensor Ocean Surveillance Data. In *Proc. 20th IEEE Conf. Decision and Control*, San Diego, CA, Dec. 1981.
- [Bog87] P. L. Bogler. Shafer-Dempster Reasoning with Applications to Multisensor Target Identification Systems. *IEEE Trans. Systems, Man and Cybernetics*, SMC-17:968–977, Nov. /Dec. 1987.
- [Bog90] P. L. Bogler. *Radar Principles with Applications to Tracking Systems*. Wiley, 1990.
- [Bow79] C. L. Bowman. Maximum Likelihood Track Correlation for Multisensor Integration. In *Proc. 18th IEEE Conf. Decision and Control*, Ft. Lauderdale, FL, Dec. 1979.
- [BPKS91] Y. Bar-Shalom, F. Palmieri, A. Kumar, and H. M. Shertukde. Analysis of Wide-Band Cross-Correlation for Target Detection and Time Delay Estimation. In *Proc. 1991 International Conf. Acoustics, Speech, and Signal Processing*, Toronto, ONT, May 1991.
- [BPKS93] Y. Bar-Shalom, F. Palmieri, A. K. Kumar, and H. M. Shertukde. Analysis of Wide-Band Cross-Correlation for Time Delay Estimation. *IEEE Trans. Signal Processing*, SP-41:385–387, Jan. 1993.
- [BS84] T. E. Bullock and S. Sangsuk-Iam. Maneuver Detection and Tracking with a Nonlinear Target Model. In *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 1984.

BIBLIOGRAPHY

- [BS86a] S. N. Balakrishnan and J. L. Speyer. Coordinate-Transformation-Based Filter for Improved Target Tracking. *AIAA Journal of Guidance*, 9(6):704–709, Nov.-Dec. 1986.
- [BS86b] V. P. Broman and M. J. Shensa. Bearings Tracking of Targets with Unobservable States. In *Proc. 1986 American Control Conf.*, Seattle, WA, June 1986.
- [BSP89] Y. Bar-Shalom, H. M. Shertukde, and K. R. Pattipati. Extraction of Measurements from an Imaging Sensor for Precision Target Tracking. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:863–872, Nov. 1989.
- [BSP90] Y. Bar-Shalom, H. M. Shertukde, and K. R. Pattipati. Precision Target Tracking for Small Extended Objects. *Optical Engr.*, 29:121–126, 1990.
- [BT73] Y. Bar-Shalom and E. Tse. Tracking in a Cluttered Environment with Probabilistic Data Association. In *Proc. 4th Symp. Nonlinear Estimation, San Diego, CA*, Sept. 1973.
- [BT75] Y. Bar-Shalom and E. Tse. Tracking in a Cluttered Environment with Probabilistic Data Association. *Automatica*, 11:451–460, 1975.
- [BW87] S. Blake and S. C. Watts. A Multitarget Track-While-Scan Filter. In *Proc. IEE Radar 87 Conf.*, London, England, Oct. 1987.
- [BWA91] W. D. Blair, G. A. Watson, and A. T. Alouani. Tracking Constant Speed Targets Using a Kinematic Constraint. In *Proc. 1991 IEEE Southeast Conf.*, 1991.
- [BWH94] W. D. Blair, G. A. Watson, and S. A. Hoffman. Benchmark Problem for Beam Pointing Control of Phased Array Radar Against Maneuvering Target. In *Proc. 1994 American Control Conf.*, Baltimore, MD, June 1994.
- [BWR91] W. D. Blair, G. A. Watson, and T. R. Rice. Tracking Maneuvering Targets with an Interacting Multiple Model Filter Containing Exponentially Correlated Acceleration Models. In *Southeastern Symp. Systems Theory*, Columbia, SC, March 1991.
- [BY91] Y. Bar-Shalom and C. Yang. Trajectory Estimation in Inertial Navigation Using Landmarks. In *Proc. 1991 American Control Conf.*, Boston, MA, June 1991.
- [Cas76] F. R. Castella. Sliding Window Detection Probabilities. *IEEE Trans. Aerospace and Electronic Systems*, AES-12:815–819, Nov. 1976.
- [Cas81] F. R. Castella. Tracking Accuracies with Position and Rate Measurements. *IEEE Trans. Aerospace and Electronic Systems*, AES-17:433–437, May 1981.
- [CB84] K. C. Chang and Y. Bar-Shalom. Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers. *IEEE Trans. Automatic Control*, AC-29(7):585–594, July 1984.
- [CB86] K. C. Chang and Y. Bar-Shalom. A Simplification of the JPDA Algorithm. *IEEE Trans. Automatic Control*, AC-31:989–991, Oct. 1986.
- [CB87] K. C. Chang and Y. Bar-Shalom. Distributed Adaptive Estimation with Probabilistic Data Association. In *Proc. 10th IFAC World Congress*, Munich, Germany, July 1987.
- [CB94] K. C. Chang and Y. Bar-Shalom. FUSEDAT: A Software Package for Fusion and Data Association with Multiple Sensors. In *Proc. 1994 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 2235, Orlando, FL, Apr. 1994.
- [CCB86] K. C. Chang, C. Y. Chong, and Y. Bar-Shalom. Joint Probabilistic Data Association in Distributed Sensor Networks. *IEEE Trans. Automatic Control*, AC-31:889–897, Oct. 1986.

BIBLIOGRAPHY

- [CCB91] K. C. Chang, C. Y. Chong, and Y. Bar-Shalom. Distributed Estimation in Distributed Sensor Networks. In K. Watanabe and S. Tsafestas, editors, *Large Scale Stochastic Systems: Control, Estimation and Detection*, Dekker, 1991.
- [CCM86] C. Y. Chong, K. C. Chang, and S. Mori. Distributed Tracking in Distributed Sensor Networks. In *Proc. 1986 American Control Conf.*, Seattle, WA, June 1986.
- [CCM87] C. Y. Chong, K. C. Chang, and S. Mori. Tracking Multiple Targets with Distributed Acoustic Sensors. In *Proc. 1987 American Control Conf.*, Minneapolis, MN, June 1987.
- [CDA86] S. B. Colegrove, A. W. Davis, and J. K. Ayliffe. Track Initiation and Nearest Neighbors Incorporated into Probabilistic Data Association. *J. of Electrical and Electronics Engr. - Australia*, 6(3), Sept. 1986.
- [CDY84] C. B. Chang, K. P. Dunn, and L. C. Youens. A Tracking Algorithm for Dense Target Environments. In *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.
- [CGE90] Y. N. Chung, D. L. Gustafson, and E. Emre. Extended Solution to Multiple Maneuvering Target Tracking. *IEEE Trans. Aerospace and Electronic Systems*, AES-26:876–887, Sept. 1990.
- [Cho79] C. Y. Chong. Hierarchical Estimation. In *Proc. MIT/ONR Workshop on CS*, 1979.
- [Cho80] S. I. Chou. *Anti-Submarine Warfare Passive Target Tracking*. Technical Report, Naval Oceans Systems Center, San Diego, CA, Jan. 1980. Final Report N00014-79, Naval Ocean Systems Center.
- [CLR83] P. L. Cowell, M. J. Larson, and A. J. Rockmore. A Bayesian Algorithm for Multiple Sensor Correlation. In *Proc. 1983 American Control Conf.*, San Francisco, CA, June 1983.
- [CM84] C. Y. Chong and S. Mori. Hierarchical Multi-Target Tracking and Classification — a Bayesian Approach. In *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.
- [CM87] R. L. Carroll and M. A. Mayor. An Exponentially Weighted Probabilistic Data Association Filter for Maneuvering Targets. In *Proc. 1987 American Control Conf.*, Minneapolis, MN, June 1987.
- [CMC85] C. Y. Chong, S. Mori, and K. C. Chang. Information Fusion in Distributed Sensor Networks. In *Proc. 1985 American Control Conf.*, Boston, MA, June 1985.
- [Col87] S. B. Colegrove. Multiple Tracking in a Cluttered Environment. In *ISSPA 87*, Brisbane, Australia, Aug. 1987.
- [CY82] C. B. Chang and L. C. Youens. Measurement Correlation for Multiple Sensor Tracking in a Dense Target Environment. *IEEE Trans. Automatic Control*, AC-27, 1250–1252, Dec. 1982.
- [Dau90] F. E. Daum. Bounds on Performance for Multiple Target Tracking. *IEEE Trans. Automatic Control*, AC-35(4):443–446, Apr. 1990.
- [DB88] O. E. Drummond and S. S. Blackman. Multiple Sensor, Multiple Target Tracking. In *Proc. 1st Nat'l Symp. Sensor Fusion*, Apr. 1988. (GACIAC PR-88-01).
- [DB93] J. Dezert and Y. Bar-Shalom. Joint Probabilistic Data Association for Autonomous Navigation. *IEEE Trans. Aerospace and Electronic Systems*, 29(4):1275–1286, Oct. 1993.
- [DB95] E. Daeipour and Y. Bar-Shalom. An IMM Approach for Target Tracking with Glint Noise. *IEEE Trans. Aerospace and Electronic Systems*, AES-31, Jan. 1995.
- [DBL94] E. Daeipour, Y. Bar-Shalom, and X. R. Li. Adaptive Beam Pointing Control of a Phased Array Radar Using an IMM Estimator. In *Proc. 1994 American Control Conf.*, pages 2093–2097, Baltimore, MA, Jun. 1994.

BIBLIOGRAPHY

- [DBP90] O. E. Drummond, S. S. Blackman, and G. C. Petrisor. Tracking Clusters of Extended Objects with Multiple Sensors. In *Proc. SPIE Conf. 1305*, pages 362–371, 1990.
- [DF93] O. E. Drummond and G. Frenkel. Glossary of Tracking Terms of SDI Panels on Tracking. In *Proc. 1993 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 1954, Apr. 1993.
- [DMPB90] S. Deb, R. Mallubhatla, K. R. Pattipati, and Y. Bar-Shalom. A Multisensor-Multitarget Data Association Algorithm. In *Proc. IEEE Intn'l Conf. Systems Engrg.*, Pittsburgh, PA, Aug. 1990.
- [DPB92a] S. Deb, K. R. Pattipati, and Y. Bar-Shalom. A New Algorithm for the Generalized Multidimensional Assignment Problem. In *Proc. IEEE Intn'l. Conf. on Systems, Man and Cybernetics*, Chicago, IL, Nov. 1992.
- [DPB92b] S. Deb, K. R. Pattipati, and Y. Bar-Shalom. A S-dimensional Assignment Algorithm for Track Initiation. In *Proc. IEEE Intn'l. Conf. on Engrg. Systems*, Kobe, Japan, Sept. 1992.
- [DPB93] S. Deb, K. R. Pattipati, and Y. Bar-Shalom. A Multisensor-Multitarget Data Association Algorithm for Heterogeneous Sensors. *IEEE Trans. Aerospace and Electronic Systems*, AES-29(2):560–568, Apr. 1993.
- [DPBY94] S. Deb, K. R. Pattipati, Y. Bar-Shalom, and M. Yeddanapudi. A Generalized S -dimensional Algorithm for Multisensor Multitarget State Estimation. In *Proc. 33rd IEEE Conf. Decision and Control*, Orlando, FL, Dec. 1994.
- [DR80] J. V. DiFranco and W. L. Rubin. *Radar Detection*. Artech House, Dedham, MA, 1980.
- [DRBB90] G. C. Demos, R. A. Ribas, T. J. Broida, and S. S. Blackman. Applications of MHT to Dim Moving Targets. In *Proc. SPIE Conf.*, Orlando, FL, 1990.
- [Dru90] O. E. Drummond. Multiple Target Tracking Lecture Notes. 1990. Copyright O.E.D.
- [FBS80] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-Target Tracking Using Joint Probabilistic Data Association. In *Proc. 1980 IEEE Conf. Decision and Control*, Albuquerque, NM, Dec. 1980.
- [FBS83] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association. *IEEE Journal Oceanic Engineering*, OE-8:173–184, July 1983.
- [FBG81] T. E. Fortmann, Y. Bar-Shalom, M. Scheffe, and S. Gelfand. Detection Thresholds for Multitarget Tracking in Clutter. In *Proc. 20th IEEE Conf. Decision and Control*, San Diego, CA, Dec. 1981.
- [FBG85] T. E. Fortmann, Y. Bar-Shalom, M. Scheffe, and S. Gelfand. Detection Thresholds for Tracking in Clutter — A Connection Between Estimation and Signal Processing. *IEEE Trans. Automatic Control*, AC-30(2):221–229, March 1985.
- [FC89] J. L. Fisher and D. P. Casasent. Fast JPDA Tracking Algorithm. *Applied Optics*, 28(2):371–376, Jan. 1989.
- [FD91] B. E. Fridling and O. E. Drummond. Performance Evaluation Methods for Multiple Target Tracking Algorithms. In *Proc. 1991 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 1481, 1991.
- [Fit85] R. J. Fitzgerald. Track Biases and Coalescence with Probabilistic Data Association. *IEEE Trans. Aerospace and Electronic Systems*, AES-21:822–825, Nov. 1985.
- [Fit86] R. J. Fitzgerald. Development of Practical PDA Logic for Multitarget Tracking by Microprocessor. In *Proc. 1986 American Control Conf.*, Seattle, WA, June 1986.

BIBLIOGRAPHY

- [Fit90] R. J. Fitzgerald. Development of Practical PDA Logic for Multitarget Tracking by Microprocessor.—In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, 1990.
- [FP78] A. Farina and S. Pardini. Track-While-Scan Algorithm in a Clutter Environment. *IEEE Trans. Aerospace and Electronic Systems*, AES-14:769–779, Sept. 1978.
- [FP81] B. Friedlander and B. Porat. Localization of Multiple Targets by Sensor Arrays: A Modeling Approach. In *Proc. 20th IEEE Conf. Decision and Control*, San Diego, CA, Dec. 1981.
- [FS84] B. Friedlander and J. D. Smith. Localization of Multiple Targets from Doppler Measurements. In *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.
- [FS85] A. Farina and F. A. Studer. *Radar Data Processing, Vol. I: Introduction and Tracking, Vol. II: Advanced Topics and Applications*. Research Studies Press, Letchworth, Hertfordshire, England, 1985.
- [Gau84] M. Gauvrit. Bayesian Adaptive Filter for Tracking with Measurements of Uncertain Origin. *Automatica*, 20:217–224, March 1984.
- [GFB84] S. Gelfand, T. E. Fortmann, and Y. Bar-Shalom. Adaptive Detection Threshold Optimization for Tracking in Clutter. In *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.
- [GM87] H. Gish and R. Mucci. Target State Estimation in a Multitarget Environment Using Multiple Sensors. *IEEE Trans. Aerospace and Electronic Systems*, AES-23:60–72, Jan. 1987.
- [GMB85] H. Gish, R. Mucci, and Y. Bar-Shalom. A New Approach to Target Detection and Estimation in Clutter. In *Proc. 1985 American Control Conf.*, Boston, MA, June 1985.
- [Gon86] K. F. Gong . Data Fusion in a Multisensor-Multicontact Environment. In *Proc. 20th Asilomar Conf. Signals, Systems and Computers, Pacific Grove, CA*, Nov. 1986.
- [Goo79] I. R. Goodman. A General Model for the Multiple Target Correlation and Tracking Problem. In *Proc. 18th IEEE Conf. Decision and Control*, Ft. Lauderdale, FL, Dec. 1979.
- [Goo81] I. R. Goodman. Application of a Combined Probabilistic and Fuzzy Set Technique to the Attribute Problem in Ocean Surveillance. In *Proc. 20th IEEE Conf. Decision and Control*, San Diego, CA, Dec. 1981.
- [GWW80] I. R. Goodman, H. L. Wiener, and W. W. Willman. *Naval Ocean Surveillance Correlation Handbook, 1979*. Technical Report, Naval Research Lab., Washington, DC, Sept. 1980. Tech. Report 8402.
- [HB89] A. Houles and Y. Bar-Shalom. Multisensor Tracking of a Maneuvering Target in Clutter. *IEEE Trans. Aerospace and Electronic Systems*, AES-25(2):176–189, March 1989.
- [HMB93] R. D. Hilton, D. A. Martin, and W. D. Blair. *Tracking with Time-Delayed Data in Multisensor Systems*. Technical Report, Naval Surface Warfare Center, Dahlgren, VA, Aug. 1993. Tech. Rep. NSWCDD/TR-93/351.
- [Hol77] J. E. Holmes. The Development of Algorithms for the Formation and Updating of Tracks. In *Proc. IEEE 1977 Int'l Radar Conf.*, London, Oct. 1977.
- [JB72] A. G. Jaffer and Y. Bar-Shalom. On Optimal Tracking in Multiple Target Environments. In *Proc. 3rd Symp. Nonlinear Estimation Theory and Its Applications*, pages 112–117, San Diego, CA, Sept. 1972.
- [JB90] C. Jauffret and Y. Bar-Shalom. Track Formation with Bearing and Frequency Measurements in the Presence of False Detections. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:999–1010, Nov. 1990.

BIBLIOGRAPHY

- [JD88] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood, NJ, 1988.
- [Kal60] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME, J. Basic Engineering*, 82:34–45, March 1960.
- [Kam89] E. W. Kamen. Multiple Target Tracking Based on Symmetric Measurement Equations. In *Proc. 1989 American Control Conf.*, Pittsburgh, PA, June 1989.
- [KB61] R. E. Kalman and R. Bucy. New Results in Linear Filtering and Prediction Theory. *Trans. ASME, J. Basic Engineering*, 83:95–108, March 1961.
- [KB93] A. Kumar and Y. Bar-Shalom. Time-Domain Analysis of Cross-Correlation for Time Delay Estimation with an Autocorrelated Signal. *IEEE Trans. Signal Processing*, SP-41(4):1664–1668, Apr. 1993.
- [KBO93] A. Kumar, Y. Bar-Shalom, and E. Oron. Precision Tracking Based on Segmentation with Optimal Layering for Imaging Sensors. In *Proc. 1993 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 1954, Orlando, FL, Apr. 1993.
- [Ken81] R. Kenefic. Optimum Tracking of a Maneuvering Target in Clutter. *IEEE Trans. Automatic Control*, AC-26:750–753, June 1981.
- [Ken91] R. Kenefic. Local and Remote Track File Registration Using Minimum Description Length. In *Proc. SPIE Conf.*, Orlando, FL, 1991. Also in *IEEE Trans. Aerospace and Electronic Systs.*
- [Ker89] T. H. Kerr. Duality Between Failure Detection and Radar/Optical Maneuver Detection. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:520–528, July 1989.
- [KMR81] J. D. Kendrick, P. S. Maybeck, and J. G. Reid. Estimation of Aircraft Target Motion Using Orientation Measurements. *IEEE Trans. Aerospace and Electronic Systems*, 17:254–260, March 1981.
- [Kos84] M. Kosaka. A Track Correlation Algorithm for Multisensor Integration. In *Proc. 5th Digital Avionics Conference*, 1984.
- [Kro84] T. Kronhamn. Multitarget Tracking by Dynamic Scene Analysis. In *Proc. 7th MIT-ONR Workshop on C3 Systems*, 1984.
- [KW85] T. Kurien and R. B. Washburn. Multiobject Tracking Using Passive Sensors. In *Proc. 1985 American Control Conf.*, Boston, MA, June 1985.
- [KZ84] K. S. P. Kumar and H. Zhou. Tracking of Multimaneuvering Targets in a Dense Multireturn Environment. In *Proc. 23th IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 1984.
- [Lac87] R. T. Lacoss. Distributed Mixed Sensor Aircraft Tracking. In *Proc. 1987 American Control Conf.*, pages 1827–1830, Minneapolis, MN, June 1987.
- [LB91a] D. Lerro and Y. Bar-Shalom. Automatic Track Formation with Target Amplitude Information. In *Proc. OCEANS 1991 Conf.*, Honolulu, HA, Oct. 1991.
- [LB91b] X. R. Li and Y. Bar-Shalom. Stability Evaluation and Track Life of the PDAF for Tracking in Clutter. *IEEE Trans. Automatic Control*, AC-36(5):588–602, May 1991.
- [LB92] X. R. Li and Y. Bar-Shalom. Mode-Set Adaptation in Multiple Model Approach to Hybrid State Estimation. In *Proc. 1992 American Control Conf.*, pages 1794–1799, Chicago, IL, June 1992.
- [LB93a] D. Lerro and Y. Bar-Shalom. Interacting Multiple Model Algorithm with Target Amplitude Feature. *IEEE Trans. Aerospace and Electronic Systems*, AES-29(2):494–509, Apr. 1993.

BIBLIOGRAPHY

- [LB93b] D. Lerro and Y. Bar-Shalom. Tracking with Debiased Consistent Converted Measurements vs. EKF. *IEEE Trans. Aerospace and Electronic Systems*, AES-29(3):1015–1022, July 1993.
- [LB93c] X. R. Li and Y. Bar-Shalom. Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking. *IEEE Trans. Control Systems Technology*, 1(3):186–194, Sept. 1993. Special issue on Air Traffic Control.
- [LB93d] X. R. Li and Y. Bar-Shalom. Performance Prediction of the Interacting Multiple Model Algorithm. *IEEE Trans. Aerospace and Electronic Systems*, AES-29(3):755–771, July 1993.
- [LB94a] X. R. Li and Y. Bar-Shalom. A Hybrid Conditional Averaging Technique for Performance Prediction of Algorithms with Continuous and Discrete Uncertainties. In *Proc. 1994 American Control Conf.*, pages 1530–1534, Baltimore, MD, June 1994.
- [LB94b] X. R. Li and Y. Bar-Shalom. Detection Threshold Selection for Tracking Performance Optimization. *IEEE Trans. Aerospace and Electronic Systems*, AES-30(3):742–749, July 1994.
- [LB94c] X. R. Li and Y. Bar-Shalom. Performance Prediction of Tracking in Clutter with Nearest Neighbor Filters. In *Proc. 1994 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 2235, pages 429–440, Orlando, FL, Apr. 1994.
- [LB95] X. R. Li and Y. Bar-Shalom. Performance Prediction of Hybrid Algorithms. In C. T. Leondes, editor, *Control and Dynamic Systems*, Academic Press, New York, 1995.
- [LBC89] P. B. Luh, Y. Bar-Shalom, and K. C. Chang. Centralized and Distributed Algorithms for Multitarget-Multisensor Tracking Systems. In C. T. Leondes, editor, *Advances in Aerospace Systems Dynamics and Control Systems*, Academic Press, 1989.
- [Lef84] C. C. Lefas. Using Roll-Angle Measurement to Track Aircraft Maneuvers. *IEEE Trans. Aerospace and Electronic Systems*, AES-20:672–681, Nov. 1984.
- [Lef91] C. C. Lefas. Improved Tracking with Mode-s Data Linked Velocity Measurements. *IEEE Trans. Aerospace and Electronic Systems*, AES-27:709–714, July 1991.
- [Li92] X. R. Li. Generation of Random Points inside an Arbitrary Hyper-Ellipsoid. In *Proc. 1st IEEE Conf. Control Applications*, pages 654–658, Dayton, OH, Sept. 1992.
- [Li93] X. R. Li. The PDF of Nearest Neighbor Measurement and Probabilistic Nearest Neighbor Filter for Tracking in Clutter. In *Proc. 32nd IEEE Conf. Decision and Control*, pages 918–923, San Antonio, TX, Dec. 1993.
- [LIN86] A. G. Lindgren, J. Irza, and S. C. Nardone. Trajectory Estimation with Uncertain and Nonassociated Data. *IEEE Trans. Aerospace and Electronic Systems*, AES-22:71–78, Jan. 1986.
- [MAB85] R. Mucci, J. Arnold, and Y. Bar-Shalom. Track Segment Association with a Distributed Field of Sensors. *J. Acoust. Soc. of America*, 78:1317–1324, Oct. 1985.
- [MC85] S. Mori and C. Y. Chong. A Multitarget Tracking Algorithm — Independent but not Poisson Cases. In *Proc. 1985 American Control Conf.*, Boston, MA, June 1985.
- [MC87] M. A. Mayor and R. L. Carroll. A Multitarget Initiation Algorithm. In *Proc. 1987 American Control Conf.*, Minneapolis, MN, June 1987.
- [MCC87] S. Mori, K. C. Chang, and C. Y. Chong. Tracking Aircraft by Acoustic Sensors — Multiple Hypothesis Approach Applied to Possibly Unresolved Measurements. In *Proc. 1987 American Control Conf.*, Minneapolis, MN, June 1987.
- [MCTW86] S. Mori, C. Y. Chong, E. Tse, and R. P. Wishner. Tracking and Classifying Multiple Targets without A Priori Identification. *IEEE Trans. Automatic Control*, AC-31(5):401–409, May 1986.

BIBLIOGRAPHY

- [MCWT83] S. Mori, C. Y. Chong, R. P. Wishner, and E. Tse. Multi-Target Multi-Sensor Tracking Problems: A General Approach. In *Proc. 1983 American Control Conf.*, San Francisco, CA, June 1983.
- [MJH81] P. S. Maybeck, R. L. Jensen, and D. A. Harnby. An Adaptive Extended Kalman Filter for Target Image Tracking. *IEEE Trans. Aerospace and Electronic Systems*, AES-17:173–180, March 1981.
- [MM80] P. S. Maybeck and D. E. Mercier. A Target Tracker Using Spatially Distributed IR Measurements. *IEEE Trans. Automatic Control*, AC-25:222–225, Apr. 1980.
- [MO78] C. E. Muehe and R. M. O'Donnell. Automating Radars for Air Traffic Control. In *Proc. ELECTRO '78*, Boston, MA, May 1978.
- [Mor77] C. L. Morefield. Application of 0-1 Integer Programming to Multi-Target Tracking Problems. *IEEE Trans. Automatic Control*, AC-22:302–312, June 1977.
- [MPG86] A. K. Mahalanabis, S. Prasand, and A. Garg. A Smoothing Algorithm for Improved Tracking in Clutter and Multitarget Environment. In *Proc. 1986 American Control Conf.*, Seattle, WA, June 1986.
- [MR83] P. S. Maybeck and S. K. Rogers. Adaptive Tracking of Multiple Hot-Spot Target IR Images. *IEEE Trans. Automatic Control*, AC-28:937–943, Oct. 1983.
- [MWI80] P. J. McLane, P. H. Wittke, and C. K. S. Ip. Least MSE Adaptation of Parameters in TWS Radar Systems. In *Proc. 1980 IEEE Int'l Radar Conf.*, Arlington, VA, Apr. 1980.
- [NB81] L. C. Ng and Y. Bar-Shalom. Modeling of Unresolved Detections for Tracking of Neighboring Targets. In *Proc. OCEANS 81 Conf.*, Boston, MA, Sept. 1981.
- [NB86] L. C. Ng and Y. Bar-Shalom. Multisensor Multitarget Time Delay Vector Estimation. *IEEE Trans. Acoustics, Speech and Sig. Proc.*, ASSP-34:669–678, Aug. 1986.
- [NCS87a] V. Nagarajan, M. R. Chidambara, and R. N. Sharma. Combinatorial Problems in Multitarget Tracking — a Comprehensive Solution. *IEE Proc.*, 134F:113–118, Feb. 1987.
- [NCS87b] V. Nagarajan, M. R. Chidambara, and R. N. Sharma. New Approach to Improved Detection and Tracking Performance in Track-While-Scan Radars. *IEE Proc.*, 134F:89–112, Feb. 1987.
- [NSC84] V. Nagarajan, R. N. Sharma, and M. R. Chidambara. An Algorithm for Tracking a Maneuvering Target in Clutter. *IEEE Trans. Aerospace Electronic Systems*, AES-20:560–573, Sept. 1984.
- [OKB93] E. Oron, A. K. Kumar, and Y. Bar-Shalom. Precision Tracking with Segmentation for Imaging Sensors. *IEEE Trans. Aerospace and Electronic Systems*, AES-29:977–987, July 1993.
- [OSG89] R. I. Odom, G. M. Stuart, and F. D. Gorecki. Design and Performance Analysis of a JPDAF Tracker for an Electronically Scanned Radar. In *Proc. AIAA Guidance and Control Conf.*, Boston, MA, Aug. 1989.
- [Pap84] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1984.
- [Par60] E. Parzen. *Modern Probability Theory and Its Applications*. Wiley, 1960.
- [PB86] A. G. Prosser and W. G. Bath. Macroscopic Spatial-Temporal Correlation Properties of Radar Clutter. In *Proc. IEEE Radar 86 Conf.*, 1986.
- [PDBW90] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn. Passive Multisensor Data Association Using a New Relaxation Algorithm. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, 1990.
- [PDBW92] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. Washburn. A New Relaxation Algorithm and Passive Sensor Data Association. *IEEE Trans. Automatic Control*, AC-37(2), Feb. 1992.

BIBLIOGRAPHY

- [PE79] D. W. Porter and T. S. Englar. Multiobject Tracking via a Recursive Generalized Likelihood Approach. In *Proc. 18th IEEE Conf. Decision and Control*, Ft. Lauderdale, FL, Dec. 1979.
- [Pre82] J. A. Presley. Censored Near-Level Detections for Multiple Target Environment. In *Proc. 16th Asilomar Conf. Circuits and Systems*, Pacific Grove, CA, Nov. 1982.
- [PSK81] K. R. Pattipati, N. R. Sandell, and L. C. Kramer. A Unified View of Multi-Object Tracking. In *Proc. 4th MIT-ONR C3 Workshop*, San Diego, CA, June 1981.
- [Rei77] D. B. Reid. *A Multiple Hypothesis Filter for Tracking Multiple Targets in a Cluttered Environment*. Technical Report, Lockheed Palo Alto Research Lab., Sept. 1977. Tech. Report LMSC-D560254.
- [Rei79a] D. B. Reid. A Non Gaussian Filter for Tracking Targets Moving over Terrain. In *Proc. 1978 IEEE Conf. Decision and Control*, San Diego, CA, Jan. 1979.
- [Rei79b] D. B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Trans. Automatic Control*, AC-24:843–854, Dec. 1979.
- [RF80] H. E. Rauch and O. Firschein. Track Assembly for Two-Dimensional Images of Binary Data. In *Proc. 19th IEEE Conf. Decision and Control*, Albuquerque, NM, Dec. 1980.
- [RM88] J. A. Roecker and C. D. McGillem. Comparison of Two-Sensor Tracking Methods Based on State Vector Fusion and Measurement Fusion. *IEEE Trans. Aerospace and Electronic Systems*, AES-24:447–449, July 1988.
- [Rog88a] S. R. Rogers. Optimal Measurement Scheduling for Multitarget Tracking with a Single Sensor. In *Proc. 27th IEEE Conf. Decision and Control*, Austin, TX, Dec. 1988.
- [Rog88b] S. R. Rogers. Tracking Multiple Targets with Correlated Measurements and Maneuvers. *IEEE Trans. Aerospace and Electronic Systems*, AES-24:313–315, May 1988.
- [Rog89] S. R. Rogers. Instability of the Decoupled Kalman Tracking Filter. *IEEE Trans. Automatic Control*, AC-34:469–471, Apr. 1989.
- [Rog91] S. R. Rogers. Diffusion Analysis of Track Loss in Clutter. *IEEE Trans. Aerospace and Electronic Systems*, AES-27(2):380–387, March 1991.
- [RWB94] C. Rago, P. Willett, and Y. Bar-Shalom. Tracking with Fused Noncoincident Measurements. In *Proc. 1994 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 2235, pages 351–362, Orlando, FL, Apr. 1994.
- [Sal88] D. J. Salmond. *Mixture Reduction Algorithms for Uncertain Tracking*. Technical Report 88004, Royal Aerospace Establishment, Farnborough, England, Jan. 1988.
- [SB75a] P. Smith and G. Buechler. A Branching Algorithm for Discriminating and Tracking Multiple Objects. *IEEE Trans. Automatic Control*, AC-20:101–104, Feb. 1975.
- [SB75b] J. J. Stein and S. S. Blackman. Generalized Correlation of Multitarget Data. *IEEE Trans. Aerospace and Electronic Systems*, AES-11:1207–1217, Nov. 1975.
- [SB85] M. J. Shensa and V. Broman. The Sequential Processing of Report Detections in Multitarget Data Association. In *Proc. 1985 American Control Conf.*, Boston, MA, June 1985.
- [SB90] H. M. Shertukde and Y. Bar-Shalom. Detection and Estimation for Multiple Targets with Two Omnidirectional Sensors in the Presence of False Measurements. *IEEE Trans. Acoustics, Speech and Sig. Processing*, ASSP-38:749–763, May 1990.
- [Sea71] R. G. Sea. An Efficient Suboptimal Decision Procedure for Associating Sensor Data with Stored Tracks in Real-Time Surveillance Systems. In *Proc. 1971 IEEE Conf. Decision and Control*, pages 33–37, Miami Beach, FL, Dec. 1971.

BIBLIOGRAPHY

- [SH89] D. D. Sworder and R. G. Hutchins. Image-Enhanced Tracking. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:701–710, Sept. 1989.
- [She82] M. J. Shensa. *BAYR — A Data Association Algorithm Based on a Bayesian Recursion*. Technical Report, Naval Ocean Systems Center, San Diego, CA, 1982. Tech. Report 848.
- [She84] S. Sherman. *Monopulse Principles and Techniques*. Artech House, Norwood, MA, 1984.
- [SHK93] D. D. Sworder, R. G. Hutchins, and M. Kent. Utility of Imaging Sensors in Tracking Systems. *Automatica*, 29(2):445–449, Mar. 1993.
- [Sit64] R. W. Sittler. An Optimal Data Association Problem in Surveillance Theory. *IEEE Trans. Mil. Electron.*, MIL-8:125–139, Apr. 1964.
- [SK71] R. A. Singer and A. J. Kanyuck. Computer Control of Multiple Site Correlation. *Automatica*, 7:455–463, July 1971.
- [SOG89] G. M. Stuart, R. I. Odom, and F. D. Gorecki. Implementation of a JPDAF Tracker for an Electronically Scanned Radar. In *Proc. AIAA Guidance and Control Conf.*, Boston, MA, Aug. 1989.
- [SS71] R. A. Singer and J. J. Stein. An Optimal Tracking Filter for Processing Sensor Data of Imprecisely Determined Origin in Surveillance Systems. In *Proc. 1971 IEEE Conf. Decision and Control*, pages 171–175, 1971.
- [SS73] R. A. Singer and R. G. Sea. New Results in Optimizing Surveillance System Tracking and Data Correlation Performance in Dense Multitarget Environments. *IEEE Trans. Automatic Control*, AC-18(6):571–582, Dec. 1973.
- [SSH74] R. A. Singer, R. G. Sea, and K. Housewright. Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environments. *IEEE Trans. Information Theory*, IT-20:423–432, July 1974.
- [Sta87] D. V. Stallard. An Angle-Only Tracking Filter in Modified Spherical Coordinates. In *Proc. 1987 AIAA Guid., Navig. and Control Conf.*, pages 542–550, 1987.
- [Tap85] B. D. Tapley . A New Method for Enhancement of Data Separability and Data Classification in Multitarget-Multisensor Tracking Problems. In *Proc. 1985 American Control Conf.*, Boston, MA, June 1985.
- [TD84] R. R. Tenney and J. R. Delaney. Distributed Aeroacoustic Tracking Algorithm. In *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.
- [TH79] J. K. Tugnait and A. H. Haddad. A Detection-Estimation Scheme for State Estimation in Switching Environments. *Automatica*, 15(4):477–481, July 1979.
- [Tho87] S. C. A. Thomopoulos . Optimal Decision Fusion in Multiple Sensor Systems. *IEEE Trans. Aerospace and Electronic Systems*, AES-23:644–653, Sept. 1987.
- [TL87] G. T. Tso and R. N. Lobbia. Multimode Guidance in Advanced Air-to-Air Missile Applications. *AIAA J. Guidance and Control*, 1987.
- [TS90] M. Tahk and J. L. Speyer. Target Tracking Subject to Kinematic Constraints. *IEEE Trans. Automatic Control*, AC-35:324–326, March 1990.
- [Tsa91] H. Tsaknakis. Tracking Closely-Spaced Objects Using Multiassignment Algorithms. In *Tri-Service Data Fusion Symp.*, Nov. 1991.
- [TW81] G. V. Trunk and J. D. Wilson. Track Initiation of Occasionally Unresolved Targets. *IEEE Trans. Aerospace and Electronic Systems*, AES-17:122–130, Jan. 1981.

BIBLIOGRAPHY

- [TW87] G. V. Trunk and J. D. Wilson. Association of DF Bearing Measurements with Radar Tracks. *IEEE Trans. Aerospace and Electronic Systems*, AES-23:438–447, July 1987.
- [Uhl92] J. K. Uhlmann. Algorithms for Multiple Target Tracking. *American Scientist*, 80:128–141, March 1992.
- [Var88] R. Varad. Scalar Correlation Algorithm Multitarget Multisensor Data Fusion. In *1st Nat'l Symp. Sensor Fusion*, Orlando, FL, Apr. 1988.
- [VB93] G. VanKeuk and S. S. Blackman. On Phased Array Radar Tracking and Parameter Control. *IEEE Trans. Aerospace and Electronic Systems*, AES-29(1), Jan. 1993.
- [VJ88] D. R. VanRheeden and A. Jones. Noise Effects on Centroid Tracker Aim Point Estimation. *IEEE Trans. Aerospace and Electronic Systems*, AES-24, March 1988.
- [Was87] R. B. Washburn. A Random Point Process Approach to Multiobject Tracking. In *Proc. 1987 American Control Conf.*, pages 1846–1852, Minneapolis, MN, June 1987.
- [Wax55] N. Wax. Signal-to-Noise Improvement and the Statistics of Track Populations. *J. Appl. Physics*, Vol. 26:586–595, May 1955.
- [Wax83] M. Wax. Position Location from Sensors with Position Uncertainty. *IEEE Trans. Aerospace and Electronic Systems*, AES-9:183–198, March 1983.
- [WCD76] D. Willner, C. B. Chang, and K. P. Dunn. Kalman Filter Algorithms for a Multisensor System. In *Proc. 1976 IEEE Conf. Decision and Control*, Clearwater Beach, FL, Dec. 1976.
- [WKBW84] R. B. Washburn, T. Kurien, A. L. Blitz, and A. S. Willsky. *Hybrid State Estimation Approach to Multiobject Tracking for Airborne Surveillance Radars*. Technical Report, Alphatech, Inc., Burlington, MA, Oct. 1984. Tech. Rep. 180.
- [Woo85] M. S. Woolfson. An Evaluation of Manoeuvre Detector Algorithms. *GEC J. of Research (Chelmsford, England)*, 3(3):181–190, 1985.
- [WWGK79] H. L. Wiener, W. W. Willman, J. R. Goodman, and J. H. Kullback. *National Ocean-Surveillance Correlation Handbook 1978*. Technical Report, Naval Research Lab., Washington, DC, 1979. Tech. Report 8340.
- [Yan86] R. M. Yannone. Expert Systems in the Fighter of the 1990s. *IEEE Aerospace and Electr. Systems Magazine*, AESM-1:12–16, Feb. 1986.
- [Yav83] Y. Yavin. Strategies Using an Observer for Steering a Random Motion of a Point in a Multitarget Environment. *Computer Methods in Applied Mechanics and Engineering*, 39:297–310, 1983.
- [YBDP94] M. Yeddanapudi, Y. Bar-Shalom, S. Deb, and K. R. Pattipati. Ballistic Missile Track Initiation from Satellite Observations. In *Proc. 1994 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 2235, Orlando, FL, Apr. 1994.
- [YWW76] S. J. Yakowitz, T. L. Williams, and G. D. Williams. Surveillance of Several Markov Targets. *IEEE Trans. Info. Theory*, IT-22:716–724, Nov. 1976.

Index

- k*-D (*k*-dimensional) assignment problem, 374
acceleration onset/termination, 27
acceptance region, 388
acceptance, track, 109
acceptance, true track, 108
accepted false track, 110
accepted track, 104, 107
acoustic arrays, 321
acquisition, target, 11
active sensors, 5
adaptive estimator, 199
age of the track, 260
Air Traffic Control (ATC), 36, 56
alignment for multiple sensors, 20
alignment of sensors in real time, 21
all neighbors modified filter, 136
alpha-beta, 26
alpha-beta-gamma, 26
ambiguous displacement measurements, 556
amplitude information, 247
amplitude information (AI), 188, 247
ancestor, 162
assignment, 373, 380
assignment cost, 481
assignment problem, 500
assignment problem dimension, 380, 507
association, 17
association (acceptance) gate, 405
association gate size, 405
association histories, 336
association hypotheses, 334, 335
association indicator, 313
association probabilities, 358
association probability, 131
association procedure, 85
association process, 6
association region, 94, 95, 213, 388
association uncertainty, 94
association, hard, 373
association, track to track, 440
asynchronous sensors, 467, 469
ATC, 36, 56
ATF, 212, 217
auction algorithm, 511, 522
auction, modified, 374
augmented PDA, 151
automated decision systems for data association and tracking, 8
automatic generation of the Markov chain, 117
automatic track formation, 212
average confirmation time, 108
average covariance update equation, 288
average distance between neighboring pixels with detections, 571, 583
average false track length, 110
average number of accepted false tracks, 110
average number of existing false tracks, 110
average pixel SNR, 534
average Riccati equation, 280, 288
average target intensity over its extent, 533
average target intensity over the entire frame, 533
average track life, 299, 302
average true converted measurement bias, 40
average true converted measurement covariance, 40
azimuth, 36
base state, 33
beacon return, 470
BEARDATTM, 182
bearing, 36
behavior modes, 6
Bernoulli sum, 407
best linear MMSE estimator, 25
bias significance, 44
branch and bound, 509
calculated TTP, 260
candidate track, 378

INDEX

- cascaded logic, 104
cell signal-to-noise ratio, 182
central dynamic association and tracking, 432
central filter, 85
centralized association and tracking, 432
centralized configuration (Type IV), 469
centralized estimation vs. fusion of local estimates, 526
centralized tracking, 437
centroid, 532, 566
centroid displacement, 532
centroid measurement, 542
centroid offset (displacement), 532
centroid offset measurement, 542
centroid variance with the effect of noise background, 572
CGH, 404, 411
classification, 457
cluster, 340, 568
cluster tracking, 7, 340
clustering, 566, 571
clutter points, 103
clutter, heterogeneous, 212
coarse gating, 505
coefficient of variation, 535
combinatorics, 315
combined estimate, 163
combined innovation, 132, 145, 282
combined innovation's covariance, 145
common conditioning, 201
common gate-history (CGH), 411
common gate-history (CGH) algorithm, 404
common validation region, 200
complete assignment algorithm, 506
complexity, 383
complexity order, 32
composite logic, 403
configurations for information processing, 429, 432
confirmation, 108
confirmation of a track, 210
confirmation probability, 226
confirmation time, 215
confirmed status, 107
confirmed track, 104
consistent, 31
consistent covariance, 31
constant angular rate, 255
constant false alarm rate (CFAR), 190
constant speed, 255
continuation, 11
contrast, 534
converted measurement covariance, 47
converted measurement Kalman filter with debiasing (CMKF-D), 46
converted measurement Kalman filter (CMKF), 46
converted measurement Kalman filter with linearization (CMKF-L), 46
converted measurement KF (CMKF), 36
cooperative targets, 470
coordinate conversion consistency, 36
coordinate selection, 16
coordinated presentation, 340
coordinated turn, 35, 255
correct association, 347
correct associations, 399
correlation coefficient between two frames, 537
correlation process, 6
correlator-trackers, 8
cost, 481
coupled estimation, 319
covariance fusion equation, 443
covariance of the combined innovations, 284
cpmf, 108
Cramer-Rao lower bound (CRLB), 176
Cramer-Rao lower bound (CRLB) in the presence of false measurements, 171
credibility of the track, 213
CRLB, 176
CRLB in the presence of false measurements, 171
cross-correlation, 321
cross-covariance, 329, 447
cross-covariance recursion, 448
crossing extended targets, 19
cumulative probability of track loss, 299
cumulative event, 336, 339
cumulative probability mass function (cpmf), 108
current association event, 336
current associations with established tracks, 337
data, 6
data association, 1, 6, 7
debiased, 31
debiased consistent, 25
debiased conversion, 40
debiasing compensation, 36
decentralized configuration, 429
decentralized system, 430
decision, 1
decorrelation of the noise sequences, 545
decoupled estimation, 319
deferred ambiguity resolution, 396
dense clusters, 509

INDEX

density-variance product, 347
dependence of two tracks, 447
design parameters, 57
design parameters for IMM/MSPDAF, 238
design procedure for estimator with variable sampling interval, 72
design selection, 65
detection indicator, 106, 313, 358
detection indicator sequence, 384
detection model, 416
detection probability, 6
detection probability of a pixel, 569
detection sequence indicator vector, 107
detection threshold selection, 279
detection, initial, 11
detection, true track, 108
detection-triggered logic, 408
detector operating point, 404
diffuse prior, 212, 318
diffuse prior model, 135
dimension of the assignment problem, 380, 507
direction finder, 436
direction finding ionosonde, 523
distance vector, 388
distributed configuration, 429, 434
distributed estimation with feedback, 466
distributed estimation without feedback, 462
distributed system, 430
Doppler, 171
Doppler differences (FDOA — frequency difference of arrival), 321
Doppler shift, 321
dual problem, 511
duality gap, 511
dynamic association and tracking algorithm, 436

efficient, 182
EKF, 28
electro-optical (EO) sensors, 500
Electronic Support Measure (ESM), 503
elevation, 36
ellipse (or ellipsoid) of probability concentration, 95
ellipse semiaxes, 444
EO sensors, 500
error independence assumption, 440, 441
ESM, 503
estimation, 1
estimation, raison d'être, 4
estimator design parameters, 57

estimator design procedure for variable sampling interval, 72
event matrix, 312
exhaustive search, 335, 339
expected number of accepted false tracks, 260, 264
expected number of false tracks, 264
expected number of false tracks in the surveillance region, 419
expected number of targets, 340
expected SNR, 249
extended Kalman filter (EKF), 28
extended target, 18, 94, 529, 532
extraneous measurement, 481
extraneous measurements, 347
extraneous true target density, 347

false alarm, 102
false alarm (false measurement), 191
false alarm probability, 250
false alarm probability per look per cell, 403
false alarm, constant rate (CFAR), 190
false alarms, 337
false associations, 399
false measurements, 99, 122
false measurements, FIM and CRLB in the presence of, 176
false target (clutter) density, 347
false track, 418
false track acceptance probability, 259
false track probability, 105, 110, 403, 404
false track probability in the absence of targets, 416
false track probability in the neighborhood of a target, 418
FDOA, 321
feasible association event, 313
feasible associations, 335
feasible events, 358
feasible joint events, 312
feasible partition, 378
feasible solution, 505
feature, 98, 151, 457
feature estimation, 457
feature likelihood ratio, 191
feature measurements, 120
feature pdf, 151
feature variables, 187
filtering, 1
FIM in the presence of false measurements, 176
fine gating, 505
Fisher information matrix (FIM), 176

INDEX

- fixed window, 210
FLIR, 532
floating point operations (Flops), 77
fluctuation, 276
footprint, 539
formation, track, 11
forward looking infrared (FLIR), 532
fraction of correct associations, 399
fraction of false associations, 399
frame, 7, 104
frame SNR, 534
frequency difference of arrival (FDOA), 321
fusion, 21, 442
fusion equation, 464, 466
fusion equation under the error independence assumption, 443
fusion equation with cross-covariance, 450
fusion of feature estimates, 434
fusion of local estimates vs. centralized estimation, 526
fusion, track to track, 440

gate, 94, 95
gate index, 405, 422
gate probability, 96
gate threshold, 95
gate-history vector, 410
gating, 85
gating based on maximum velocity and measurement noise, 118
Gaussian distributed, 95
Gaussian plume intensity, 539
generalized pseudo-Bayesian estimator, 32
generalized 3-D assignment problem, 507
generalized auction algorithm, 522
generalized likelihood ratio (GLR), 505
generalized matching problem, 500
ghosting, 501
global averaging, 295
global estimate, 459, 462
GLR, 505
GPB_n, 32
graph, 509
ground truth, 56, 470, 482

hard association, 373
heterogeneous clutter, 212
hierarchical hybrid configuration, 432
history, 120, 335
history-conditioned estimate, 163
hit, 7

hit pattern, 190
hot spot, 18
hot spot target, 532
hybrid, 32
hybrid approximation of the Riccati equation, 280
hybrid conditional averaging, 295
hybrid conditional averaging performance prediction algorithm, 295
hybrid configuration, 438
hybrid performance prediction algorithm, 296
hybrid system, 295
HYCA, 295
hypothesis dropping threshold, 349
hypothesized track, 334

i.i.d. (independent identically distributed), 129
ICBES, 58
ICBS, 58
ID, 470
image, 18
image centroid, 535
image centroid measurement, 535
image offset measurement, 537
image segmentation, 529
image-mixing parameter, 551
IMDATTM, 582
IMM, 32
IMM estimator design parameters, 57
IMM/MSPDAF, 232
IMM/MSPDAF design parameters, 238
immediate resolution, 395
IMMPDAF, 209
IMMPDAF, multisensor, 227
IMMPDAFAI, 247, 251
incomplete track, 384
incomplete track likelihood function, 384
information processing, 3
information processing configurations, 432
information reduction factor, 176, 288
initial detection, 11
initiating pair, 213
initiator, 213, 389
innovation, combined, 132
intelligent tracker, 213
intensity band-pass thresholding, 567, 569
interacting multiple model estimator (IMM), 32
interacting multiple model PDAF (IMMPDAF), 209
interaction, 33

joint association event, 312, 358

INDEX

- joint association event probabilities, 314, 360, 370
- joint association events, 311
- joint probabilistic data association filter (JPDAF), 319
- joint probabilistic data association (JPDA), 310
- joint probabilistic data association filter (JPDAF), 307, 333
- JPDA, 310
- JPDA coupled filter (JPDACF), 319, 328, 333
- JPDA Merged-Measurement Coupled Filter (JP-DAMCF), 550, 555
- JPDA with merged measurements (JPDAM), 355
- JPDA, nonparametric, 318
- JPDA, parametric, 317
- JPDACF, 319, 328, 333
- JPDAF, 307, 319, 333
- JPDAM, 308, 333, 355
- JPDAMCF, 333, 550, 555, 558
- Kalman filter (KF), 23
- knots/Hertz ratio, 320
- likelihood function, 201, 211, 337
- likelihood function of a track, 125
- likelihood function of an incomplete track, 384
- likelihood function of the feature, 457
- likelihood function of the measurements, 314
- likelihood ratio, 175, 504
- likelihood ratio of a possibly incomplete track, 384
- likelihood ratio vector, 381
- list, 374
- lists of measurements, 507
- local Cartesian coordinate frame, 474
- local data associator/tracker, 434
- local estimates, 459, 462
- log-likelihood function, 126
- log-likelihood ratio, 381, 504
- logic, M out of N , 403
- LOS, 500
- lost track, 270
- low observable, 247
- low observable target), 17
- maintenance, 7
- maneuvering index, 58
- maneuvering index, 26, 112
- maneuvering targets (variable behavior), 15
- maneuvers, 27
- marginal association probabilities, 319
- marginal event probabilities, 360
- Markov chain automatic generation, 117
- Markov chain synthesis, 110
- maximum likelihood (ML), 376
- mean track life, 192
- measured amplitude, 247
- measured bearing, 38
- measured range, 38
- measurement, 7
- measurement formation (extraction), 7
- measurement association from multiple synchronous sensors, 385
- measurement association indicator, 313
- measurement extraction, 529
- measurement history, 162, 335
- measurement indicator matrix, 381
- measurement sequence, 120
- measurement to track association, 6
- measurement to measurement association, 6, 7
- measurement validation example, 184
- measurement-oriented, 334
- measurement-to-measurement association, 104
- measurements, 2, 5
- measurements out of sequence, 87, 91
- merged (unresolved) measurement, 370
- merged centroid measurement, 556
- merged measurement, 355, 369
- merged measurements, 308
- merging, 100
- merging probability, 367, 368
- MHT, 308, 334
- minimum target separation, 422
- miss probability, 178
- mixed coordinate (linearized) filter, 36
- mixed coordinate Extended Kalman Filter (EKF), 46
- ML, 376, 450
- MM estimator, 32
- MM system, 32
- MMSE, 450
- MMSE state estimator, 25
- modal state, 33
- mode, 33
- mode jump process, 33
- mode switching probabilities, 212
- mode-conditioned filtering, 33
- modified auction, 374
- modified filter, all neighbors, 136
- modified filter, selected neighbor, 169
- modified Riccati equation, 288
- modules in Multiple Model tracking algorithms, 27
- most likely partition, 380
- most probable hypothesis, 339

INDEX

moving average system, 544
MULTIDATTM, 217
multiframe signal processing, 14
multiframe thresholding, 14
multipath propagation, 16
multiple frame association, 17
multiple hypothesis tracker (MHT), 308
multiple hypothesis tracking (MHT), 334, 352
multiple model (MM) system, 32
multiple model PDAF (MMPDAF), 199
multiple sensors, 20, 85
multiple synchronous sensors, 385
multiple target track formation from a single passive sensor, 385
multiple targets in clutter, 18
multisensor IMMPDAF, 227
multisensor PDAF (MSPDAF), 227, 230
multisensor-multitarget position estimation problem, 500

narrow band, 171
narrow band target emission, 321
navigation with landmarks, 18
nearest neighbor clustering, 571
nearest neighbor clustering criterion, 567
nearest neighbor standard filter (NNSF), 119, 123
nearly constant acceleration motion, 27
nearly constant velocity motion, 27
NEES, 63
negative log-likelihood function, 381
negative-time measurement update, 91
new targets, 337
NIS, 95
NNSF, 119, 123
no target, 209
noise level switching, 65
noise reduction, 27, 59
noise sequence decorrelation, 546
non-Bayesian data association, 6
non-interference condition, 422
non-overlap, 422, 425
non-polynomial complexity, 383
non-synchronized sensors, 231
nonmaneuvering targets, 15
nonparametric JPDA, 318
nonparametric PDA, 136
nonsimulation performance prediction, 280, 288
normalized error squared (NES), 42
normalized estimation error squared (NEES), 63
normalized innovation squared, 145
normalized innovation squared (NIS), 95

normalized position error squared, 144
normalized velocity error squared, 144
normally distributed, 95
NP-hard, 374, 383, 500, 507
number of continuing tracks, 336
number of false alarms, 103
number of false measurements, 336
number of new tracks, 336

observation, 7
occasional measurement, 341
offspring event, 336
optimal detection threshold, 291
optimal estimator, 2
optimal fusion algorithm, 464
optimal global estimate, 434, 453, 462, 466
optimal global estimates, 434
optimal MM estimator, 32
order of, 507
order of complexity, 32
origin of the measurements, 3
origin uncertainty, 85
origin uncertainty in the measurements, 279
outlier rejection, 482
overall state estimate and covariance, 33

parabolic interpolation, 537
parallel search, 391
parallel updating, 89
parametric JPDA, 317
parametric PDA, 136
parent event, 336, 339
partition, 380, 505
partition indicator vector, 381
PASSDATTM, 515
passive sensors, 5
passive sonar, 171
PDA filter (PDAF), 120, 129
pdf, 95
pdf of a false measurement, 103
performance evaluation for multiple targets in clutter, 18
performance measure, system level, 264
performance prediction, 403
performance prediction for a single target in clutter, 17
performance prediction for maneuvering targets, 16
performance prediction for nonmaneuvering targets, 15
performance prediction technique, 105

INDEX

- performance prediction, nonsimulation, 280
persistent clutter, 103
persistent interference, 100
pixels, 102
plot, 190
pmf, 102
pmf of the number of false measurements, 338
pmf of the number of new targets, 338
point measurements, 94
Poisson model, 135
polar to Cartesian, 36
polar to Cartesian conversion, 31
posterior probability of a joint association event, 316
precision filtering, 392
precision tracking, 11, 211
precomputed fixed gains, 26
predetection fusion, 22
prediction, 23
preliminary track, 104, 210, 213
preliminary tracking, 210
prewhitening of the measurement noise, 545
primal feasible solution, 511
primal problem, 511
primal-dual approach, 511
prior probability of a joint association event, 315
probabilistic (Bayesian) data association, 6
probabilistic data association (PDA), 120, 129
probability concentration, 95
probability density function (pdf), 95
probability evaluation, 33
probability mass function (pmf), 102
probability of a false alarm (false measurement), 191
probability of target detection, 191
probability region, 444
processing configurations for multiple sensor data fusion, 21
processing configurations for multiple sensors, 20
processor sizing, 110
proximity distance, 571
pseudo sample spatial measurement density, 318
purity, track, 399
radar, over the horizon, 44
random clutter, 103
random field, 574
range, 36
range resolution cell, 74
rank of the measurement, 190
receiver operating characteristic (ROC), 404
reduction of the uncertainty in fusion, 444
registration, 7
relative signal strength, 362
relative strength, 356
relaxation algorithm, 511
remote sensors, 3
report, 7
residual clutter, 99
resolution, 308, 355
resolution cell, 356
resolution cell volume, 290
resolution cells, 94, 102, 405, 472
resolution indicator, 358
resolution, finite, 100
resolved measurement, 360
resource allocation in a multisensor environment, 22
return, 7
revisit, 76
RMS (root mean square), 59
RMS errors (RMSE), 78
robustness, 217
ROC, 404
root mean square, 59
same target hypothesis, 441
sample spatial density, 135
scan, 7, 104
scan to scan association, 7
scanning sensor, 7, 386
scanning sensors, 374
scheduling, sensor, 16
SCR, 192
SD, 61
secondary errors, 36
selected neighbor modified filter, 169
selection of the best design, 65
sensor, 7
sensor scheduling, 16
sensors, scanning, 374
sensors, staring, 374
sequence of measurements, 120
sequences of measurements, 334
sequential state update, 227
sequential updating, 88
sequential updating procedure, 556
set packing, 509
signal detection, 7
signal processing, 7
signal to clutter ratio (SCR), 190, 192
signal to noise ratio (SNR), 404

INDEX

- signal to noise ratio in a cell, 182
single frame (scan) thresholding, 14
single frame association, 17
single frame signal processing, 13
single hit false alarm probability, 190
single maneuvering target in clutter, 17
single nonmaneuvering target in clutter, 17
single passive sensor, 385
single scan misassociation probability, 185
skin returns, 470
slant range, 36
sliding window, 210, 226, 258, 391
small image, 94, 529
SNR, 404
SNR in a cell, 182
SNR, frame, 534
SNSF, 119, 123
SOC, 404, 419
sojourn time, 58
sonar, passive, 171
sparse scenarios, 509
spatial density of the false alarms, 102
splitting target in clutter, 19
spread of the innovations, 156
spread of the association probabilities, 277
spread of the innovations, 132
squawk, 470
standard coordinate conversion, 38
staring sensor, 7, 386
staring sensors, 374
state, 5
state augmentation, 544
static association, 432
static intersensor association, 436, 500
stealthy target, 215
stochastic Riccati equation, 285
straight and level, 57
strongest neighbor standard filter (SNSF), 119, 123
supercluster, 340
supermeasurements, 500
surveillance systems, 3
Swerling I detection model, 416
Swerling I fluctuation model, 276
synchronized, 436
synchronized sensors, 87, 447, 462, 466
system capacity, 422
system level performance measure, 264
system operating characteristic (SOC), 404, 419
system requirements, 403
system target capacity, 403
tail probability, 126
target features, 320
target acquisition, 7, 11
target association indicator, 358
target attribute data fusion across sensors, 21
target centroid variance with the effect of noise background, 572
target cluster centroid variance, 566
target detection, 171, 191
target detection indicator, 313
target detection probability, 105, 250
target detection probability per look, 403
target extent, 533
target feature, 247
target identification number, 470
target intensity band, 569
target maneuvering index, 26, 58, 112
target motion analysis (TMA), 171
target separation, 422
target single-hit detection probability, 190
target state observability, 385
target to clutter amplitude likelihood ratio, 250
target to clutter feature likelihood ratio, 152, 191
target track, 418
target track detection, 253
target track detection based on time difference of arrival, 20
target track detection performance, 260
target track detection probability, 259, 403, 404, 407
target track probability in the presence of false alarms, 418
target-oriented, 334
TDOA, 321
tentative track, 104
termination of a track, 210
the size of the problem, 507
threshold exceedance, 7
time difference of arrival (TDOA), 321
time stamp, 7
time tag, 374
total target-related intensity, 533
track, 6, 335
track confirmation, 210
track life, 299
track termination, 210
track acceptance, 109, 259
track age, 210
track assembly, 7
track before declare, 210
track before detect, 14, 210

INDEX

- track continuation, 7
track formation, 7, 11, 104, 120, 210, 212, 247, 253, 403
track formation with ideal sensor, 13
track formation with realistic sensor, 13
track fusion, 429, 434
track indicator vector, 381
track initiation, 210, 403
track life, 280
track likelihood function, 125
track loss, 299
track loss probability, 295
track maintenance, 11, 119, 211, 247
track purity, 399
track split, 119, 124, 389
track start-up, 210
track to track association, 6
track to track association, 429, 434, 441, 449
track to track association and fusion, 432, 440, 447
track, candidate, 378
tracker operating characteristic (TOC), 280, 291
tracking, 1
tracking accuracy, 285, 291
tracking algorithms, 23
tracking error minimization, 291
tracking filter, 7
tracking in the presence of multipath propagation, 16
tracking instability region, 291
tracking performance, 279
tracking window, 529, 533
tracking, precision, 11
traveling salesman's problem, 507
true bearing, 38
true measurement, 122
true range, 38
true target probability (TTP), 209, 213, 258
true track acceptance probability, 259
true track detection (acceptance), 108
true track detection probability, 105
turning, 57
type I configuration, 432, 433
type II configuration, 432
type IIa configuration, 434, 440, 447, 462
type IIb configuration, 434, 466
type III configuration, 432, 436, 500
type IV configuration, 432, 437

undetectable target, 209
uniform motion, 57
uniform-Gaussian mixture, 174, 201

unresolved measurement, 100, 355, 360
update, 23
updating, 7

validated measurements, 98, 174
validation matrix, 312, 358
validation region, 94, 95, 211
validation region volume, 130
validity limit of the standard conversion, 44
variable sampling interval, 71
variable sampling interval estimator design procedure, 72
volume, 96
volume of the validation region, 130

wake, 120
wake phenomenon, 168
wide band correlation, 321

zero-one programming, 382

