Probabilistic Association and Fusion for Multi-sensor Tracking Applications

Mark L Krieg

Thesis submitted for the degree of

Doctor of Philosophy

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

January 98

Contents

Abstract

The use of state space techniques to track targets using measurements from multiple sensors is considered. In particular, the operation of the asynchronous fused Kalman filter is investigated and evaluated, using real data collected from a collocated tracking radar and optical tracking system. An analysis of the effect of additional sensors on the filter's sensitivity to model mismatch is carried out.

The performance of the tracking filter is unacceptable in multi-target and/or cluttered environments. This poor performance is attributed to the filter treating all measurements as if they originated from the target of interest. This is often not the case in real environments; therefore some form of data association is required. Two algorithms are developed to overcome this inadequacy, the multi-sensor Probabilistic Multi-Hypothesis Tracking (msPMHT) algorithm and the multi-sensor Probabilistic Least Squares Tracking (msPLST) algorithm. Both these algorithms estimate the measurement to target assignments and the target states simultaneously, the msPMHT using maximum likelihood techniques and the msPLST utilising least squares.

Similarities and differences between the linear Gaussian msPMHT and the msPLST algorithms are discussed. The characteristics and performance of both algorithms are compared using simulated and real data.

A general msPMHT algorithm is introduced with multiple measurement models for each physical sensor. Measurement to sensor assignments, associating individual measurements with selected sensor models, are estimated along with the measurement to target assignments and target states. This allows the algorithm to adapt to varying sensor parameters by changing sensor models.

Declaration

This work contains no material that has been accepted for the award of any other degree or diploma in any university or other teritary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

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

Date 23 January 98

Acknowledgements

I wish to thank the following people for their assistance and support during my candidature.

My academic supervisor, Professor Doug Gray of the Electrical and Electronic Engineering Department, University of Adelaide, provided technical guidance and encouraged me to develop sound research skills.

The executive of the Tactical Surveillance Systems Division (formerly Microwave Radar Division) of the Defence Science and Technology Organisation, Australia for the opportunity to pursue this work as part of its research programme. The data collected using the division's hardware and staff provided a significant contribution to this work.

The Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP) for the use of their facilities and the opportunity to meet and converse with various local and international researchers.

Dr Roy Streit, Evangelos Giannopoulos and Tod Luginbuhl of the Naval Undersea Warfare Center, Newport, Rhode Island in the United States of America for the interesting and productive discussions we had through our common interest in PMHT.

My wife, Karen, for her continued encouragement and support.

List of Figures

List of Tables

Abbreviations

AFKF	asynchronous fused Kalman filter
AVT	automatic video tracker
CSSIP	Cooperative Research Centre for Sensor Signal and Information Processing
DSTO	Defence Science and Technology Organisation
EM	Expectation-Maximisation
FG	fusion gain
FOV	field of view
GPDR	Generic Pulse Doppler Radar
IMM	interacting multiple model
IR	infrared
JPDA	joint probabilistic data association
JPDAF	joint probabilistic data association filter
MAP	maximum *a posteriori*
MF	mismatch factor
MHT	multiple hypothesis tracking
MMSE	minimum mean square error
MRD	Microwave Radar Division (now TSSD)
msPLST	multi-sensor Probabilistic Least Squares Tracking
msPMHT	multi-sensor Probabilistic Multi-Hypothesis Tracking
PDA	probabilistic data association
PDAF	probabilistic data association filter
pdf	probability density function
pmf	probability mass function

PMHT	Probabilistic Multi-Hypothesis Tracking
SNR	signal to noise ratio
TEC	track error covariance
TSSD	Tactical Surveillance Systems Division

Symbols

$\mathbf{0}$	Matrix of zeros.
$\mathbf{1}$	Identity matrix.
$\mathrm{E}[\cdot]$	Expectation or expected value.
$\mathbf{F}_{t_i}$	State transition matrix at time t_i.
$\mathbf{F}_{t_i}^{(m)}$	State transition matrix for target m at time t_i.
$\mathbf{H}_{t_i}$	Measurement matrix at time t_i.
$\mathbf{H}_{t_i}^{(m,s)}$	Measurement matrix mapping the state of target m onto the measurement space of sensor s at time t_i.
$\mathbf{H}_{t_i}^{(s)}$	Measurement matrix for sensor s at time t_i.
$\bar{\mathbf{H}}_{t_i}^{(m,s)}$	Measurement transformation matrix for sensor s and target model m at time t_i.
$\tilde{\mathbf{H}}_{t_i}^{(m)}$	Composite measurement matrix for target m at time t_i.
J	Cost function for the msPLST.
$J_m^{(\mathbf{X})}$	msPLST cost sub-function for target m.
$\mathbf{K}^T$	Complete collection of measurement to target assignments.
$\mathbf{K}_{t_i}$	Kalman filter gain at time t_i.
$\mathbf{L}^T$	Complete set of measurement to sensor assignments.
$\mathcal{L}_{t_i}^{(\Pi)}$	msPMHT dual auxiliary function incorporating the target measurement probability constraints using Lagrangian multipliers.
$\mathcal{L}_{t_i}^{(\Xi)}$	General msPMHT dual auxiliary function incorporating the sensor measurement probability constraints using Lagrangian multipliers.
$\mathcal{L}^{(\mathbf{A})}$	msPLST dual cost sub-function for the target assignment weights.
M	Total number of targets.
$\mathcal{N}(\mu, \Sigma)$	Gaussian or normal distribution with mean μ and covariance Σ.

$N_X^{(m)}$	Dimension of state variables for target m.
$N_Z^{(s)}$	Dimension of the measurements from sensor s.
$\mathbf{O}^T$	Complete batch observer.
$\hat{\mathbf{O}}^T$	Estimate of the batch observer.
$\mathbf{O}_{t_i}$	Observer state for the measurement scan at time t_i.
$\mathbf{P}_{t_i\|t_j}$	State estimate covariance at time t_i given all measurements up to and including time t_j.
Q	Auxiliary function of the EM algorithm.
$\mathbf{Q}_{t_i}$	Process noise covariance at time t_i.
$\mathbf{Q}_{t_i}^{(m)}$	Process noise covariance for target m at time t_i.
$Q_{t_i}^{(\mathbf{\Pi})}$	msPMHT auxiliary sub-function for the target measurement probabilities.
$Q_{t_i}^{(\boldsymbol{X})}$	msPMHT auxiliary sub-function for the target states.
$Q_{t_i}^{(\mathbf{\Xi})}$	msPMHT auxiliary sub-function for the sensor measurement probabilities.
R	Range.
$\hat{R}$	Range estimate.
$\dot{R}$	Range rate.
$\hat{\dot{R}}$	Range rate estimate.
$\mathbf{R}_{t_i}$	Measurement noise covariance at time t_i.
$R_{t_i}^{(r)}$	The r^{th} range measurement at time t_i.
$\dot{R}_{t_i}^{(r)}$	The r^{th} Doppler velocity measurement at time t_i.
$\mathbf{R}_{t_i}^{(m,s)}$	Measurement noise covariance for sensor s and target m at time t_i.
$\mathbf{R}_{t_i}^{(s)}$	Measurement noise covariance for sensor s at time t_i.
$\tilde{\mathbf{R}}_{t_i}^{(m)}$	Noise covariance of the composite measurement for target m at time t_i.
S	Total number of sensors or sensor models.
$\mathcal{S}^{(s)}$	Set of sensor models associated with physical sensor s.
T	Number of measurement scans in the batch of measurements.
$^\mathsf{T}$	Matrix transpose.
$\mathbf{X}^{(m)}$	Set of state vectors for target m.
$\mathbf{X}^T$	Complete collection of target states.

$\mathbf{X}_{t_i}$	Set of state variables for all targets at time t_i.
$\hat{\mathbf{X}}^{(m)}$	Estimate of all target states for target m.
Z^{-1}	Delay of one update interval.
$\mathbf{Z}^T$	Complete batch of measurements.
$\mathcal{Z}_{t_i}^{(s)}$	Set of measurement numbers or indices representing the measurements produced by the physical sensor s at time t_i.
$f^{(m)}(\cdot)$	General process or dynamic model for target m.
$h^{(m,s)}(\cdot)$	General measurement model for sensor s and target m.
$\mathbf{k}_{t_i}$	Set of measurement to target assignments for the measurement scan at time t_i.
$k_{t_i}^{(r)}$	Measurement to target assignment for the r^{th} measurement in the scan at time t_i.
$\mathbf{l}_{t_i}$	Set of measurement to sensor assignments for the measurement scan at time t_i.
$l_{t_i}^{(r)}$	Measurement to sensor assignment for the r^{th} measurement in the scan at time t_i.
$n_{\mathcal{S}}$	Number of sensor sets $\mathcal{S}^{(s)}$ representing the physical sensors.
n_{t_i}	Number of measurements in the measurement scan at time t_i.
$n_{\mathcal{Z}_{t_i}^{(s)}}$	Number of measurements from the physical sensor s at time t_i.
$\mathrm{p}(\cdot)$	Probability.
q^2	Continuous time process noise covariance.
q_ε^2	Continuous time elevation process noise covariance.
q_η^2	Continuous time azimuth process noise covariance.
q_R^2	Continuous time range process noise covariance.
t_0	Initial time prior to the arrival of any measurements.
t_i	Arbitrary time where $t_{i-1} \leqslant t_i \leqslant t_{i+1}$.
$\mathbf{v}_{t_i}$	Measurement noise at time t_i.
$\mathbf{v}_{t_i}^{(m,s)}$	Measurement noise at time t_i for sensor s and target m.
$\mathbf{v}_{t_i}^{(s)}$	Measurement noise of sensor s at time t_i.
$\tilde{\mathbf{v}}_{t_i}^{(m)}$	Measurement noise of the composite measurement for target m at time t_i.
$\mathbf{w}_{t_i}$	Process noise at time t_i.

$\mathbf{w}_{t_i}^{(m)}$	Process noise for target m at time t_i.
$\mathbf{x}_{t_i}$	State vector of a target at time t_i.
$\mathbf{x}_{t_i}^{(m)}$	State vector of target m at time t_i.
$\bar{\mathbf{x}}_{t_0}^{(m)}$	Initial *a priori* state vector for target m.
$\hat{\mathbf{x}}_{t_i\|t_j}$	State estimate at time t_i given all measurements up to and including time t_j.
$\hat{\mathbf{y}}_{t_i\|t_j}^{(m)}$	Kalman smoother intermediate (filtered) state estimate at time t_i for target m, given all measurements up to and including time t_j.
$\mathbf{z}_{t_i}$	Measurement scan at time t_i. May also represent a single measurement vector if each scan contains only one measurement.
$\mathbf{z}_{t_i}^{(r)}$	The r^{th} measurement vector in the measurement scan at time t_i.
$\tilde{\mathbf{z}}_{t_i}^{(m)}$	Composite measurement for target m at time t_i.
$\Delta\varepsilon$	Elevation boresight error.
$\Delta\eta$	Azimuth boresight error.
$\mathbf{\Pi}^T$	Complete set of target measurement probabilities for all targets at all measurement times.
$\bar{\mathbf{\Sigma}}_{t_0}^{(m)}$	Covariance of initial *a priori* state vector for target m.
$\mathbf{\Xi}^T$	Complete set of sensor measurement probabilities for all sensors at all measurement times.
$\Psi^{(K)}$	Probability mass function of the target assignment probabilities.
$\Psi^{(L)}$	Probability mass function of the sensor assignment probabilities.
$\Psi^{(X)}$	Probability density function of the target states for all target models.
$\alpha_{t_i}^{(m,r)}$	Target assignment weight for target m and measurement r at time t_i.
$\hat{\alpha}_{t_i}^{(m,r)}$	Estimated target assignment weight for measurement r and target m at time t_i.
ε	Elevation.
$\hat{\varepsilon}$	Elevation estimate.
$\dot{\varepsilon}$	Elevation rate.
$\hat{\dot{\varepsilon}}$	Elevation rate estimate.
$\boldsymbol{\epsilon}_{t_i}^{(m)}$	Error between the actual and predicted states of target m at time t_i.
$\varepsilon_{t_i}^{(r)}$	The r^{th} elevation measurement at time t_i.

$\boldsymbol{\epsilon}_{t_i}^{(m,r)}$	Error between the r^{th} measurement at time t_i and the state of target m at that time.
$\zeta^{(m,s)}$	Conditional probability density function of measurements from the s^{th} sensor at time t_i, given that they originate from target m.
η	Azimuth.
$\hat{\eta}$	Azimuth estimate.
$\dot{\eta}$	Azimuth rate.
$\hat{\dot{\eta}}$	Azimuth rate estimate.
$\eta_{t_i}^{(r)}$	The r^{th} azimuth measurement at time t_i.
$\boldsymbol{\pi}_{t_i}$	Set of target measurement probabilities for all targets at time t_i.
$\pi_{t_i}^{(m)}$	Target measurement probability for target m at time t_i.
$\hat{\pi}_{t_i}^{(m)}$	Estimate of the target measurement probability for target m at time t_i.
$\sigma_{\varepsilon}^{(s)\,2}$	Noise covariance of the elevation position measurements from sensor s.
$\sigma_{\eta}^{(s)\,2}$	Noise covariance of the azimuth position measurements from sensor s.
$\sigma_{R}^{(s)\,2}$	Noise covariance of the range position measurements from sensor s.
$\sigma_{\dot{R}}^{(s)\,2}$	Noise covariance of the Doppler velocity position measurements from sensor s.
$\boldsymbol{\xi}_{t_i}$	Set of sensor measurement probabilities for all sensor models at time t_i.
$\xi_{t_i}^{(p)}$	Sensor measurement probability for sensor model p at time t_i.
$\psi^{(m)}$	Probability density function of the m^{th} target's states.
$\omega_{t_i}^{(m,p,r)}$	Assignment probability for measurement $\mathbf{z}_{t_i}^{(r)}$, sensor model p and target m.
$\omega_{t_i}^{(m,r)}$	Target assignment probability for measurement $\mathbf{z}_{t_i}^{(r)}$ and target m.
$\bar{\omega}_{t_i}^{(p,r)}$	Sensor assignment probability for measurement $\mathbf{z}_{t_i}^{(r)}$ and sensor model p.

Glossary

a posteriori Reasoning or processing that arrives at causes from effects, i.e., knowledge gained from processing.

a priori Denotes knowledge gained independently of processing, i.e., known before processing commences.

active sensors Sensors that locate targets by emitting energy and detecting any resulting reflections.

actual FG The fusion gain calculated or estimated taking into account any filter model errors.

actual TEC The track error covariance obtained when errors in the filter's models are taken into account.

additional sensor The sensor that is added to a single sensor system to produce a dual sensor system.

assignment probability The probability of a particular measurement being produced by a specified sensor model and target.

assumed FG The fusion gain calculated when all filter parameters are assumed to be matched to the data.

assumed TEC The track error covariance calculated when it is assumed that the filter models are matched to the data.

asynchronous fused Kalman filter A variable update rate Kalman filter that uses measurements from multiple dissimilar sensors.

asynchronous fusion *see* asynchronous sensor fusion

asynchronous sensor fusion Combining measurements, from multiple sensors, that occur at different times, at different rates and, possibly, intermittently.

asynchronous sensors Two or more sensors that do not consistently produce simultaneous measurements.

automatic video tracker A device that detects targets in a video image and provides a measure of their positions.

clear sky An environment devoid of other targets and significant clutter.

composite measurement A single synthetic measurement produced by compressing all measurements in a measurement scan into a single entity.

composite measurement model A single measurement model representing a composite or compressed measurement.

continuous time process noise The process noise in the continuous time domain from which the discrete time process noise is derived.

data association The process of determining the measurements or information produced by each target or source.

data compression The combining of multiple measurements into a single composite measurement.

data fusion The combination of information from multiple sources into a single entity.

dissimilar sensors Sensors that differ in one or more of measurement type, measurement function, noise statistics or physical measurement characteristics.

Doppler velocity The velocity of a target in the direction directly toward the sensor, i.e., the negative of the range rate.

dynamic model *see* process model

Expectation-Maximisation An iterative algorithm for estimating the parameters describing the probability distributions of unknown random variables from incomplete data.

fused Kalman filter A Kalman filter operating on measurements from multiple sensors.

fusion gain The ratio of single sensor track error covariance to multi-sensor track error covariance.

gating Limiting the measurements to be processed to those that meet some criteria, e.g., those within some specified distance from the predicted target position.

general multi-sensor Probabilistic Multi-Hypothesis Tracking An extension of the msPMHT algorithm that allows multiple sensor models for each physical sensor.

Kalman filter A state space tracking filter that estimates the state of a system from past and current measurements, using a model of the expected target dynamics.

Kalman gain Gain or weight applied to the measurement innovation by the Kalman filter to correct its state estimate.

Kalman smoother A state space tracking algorithm that estimates each state of a system using all measurements within the batch.

least squares A technique that produces state estimates by minimising the squared errors between the measurements and the target states.

maximum *a posteriori* A technique that produces state estimates by maximising the conditional probability of the states, given the measurements.

maximum likelihood A technique that produces state estimates by maximising the conditional probability of the measurements, given the states.

measurement level fusion The combining or fusing of measurements from multiple sensors into a single composite target track.

measurement matrix A matrix describing the relationship between a measurement and the state of a particular target.

measurement model A model representing the characteristics of the sensor and the relationship between the sensor's outputs and the target state.

measurement noise The part of the measurement model that represents the uncertainty in the model, plus any noise originating from the sensor, target or environment.

measurement rate The frequency at which measurements arrive from a sensor or sensors, sometimes referred to as the sample rate or measurement frequency.

measurement scan A set of simultaneous measurements from one or more sensors, i.e., all the measurements occurring at time t_i.

measurement to sensor assignment Contains a number identifying the sensor that provided the corresponding measurement.

measurement to target assignment Contains a number identifying the target that produced the corresponding measurement.

minimum mean square error A technique that produces state estimates by minimising the expectation of the squared errors between the state estimates and the true states, given the measurements.

mismatch factor The ratio of actual FG to assumed FG, giving a measure of the sensitivity to model errors of a multi-sensor tracking filter relative to that of a single sensor tracking filter.

mismatch ratio The ratio of the assumed value of a filter parameter to the actual value representing the data.

multi-sensor Probabilistic Least Squares Tracking An algorithm for multi-sensor multi-target tracking that estimates soft target assignments and target states from a batch of measurements using least squares techniques.

multi-sensor Probabilistic Multi-Hypothesis Tracking An algorithm for multi-sensor multi-target tracking that estimates soft target assignments and target states from a batch of measurements using maximum likelihood techniques.

observer A system that estimates the state of another system using the other system's inputs and outputs.

original sensor The sensor that is common to both a multiple sensor system and a single sensor system.

passive sensors Sensors that locate targets by detecting energy emitted from the target.

positional fusion *see* sensor registration

Probabilistic Multi-Hypothesis Tracking A multi-target tracking algorithm that estimates soft target assignments and target states from a batch of measurements.

process model A model representing the expected or assumed dynamic behaviour of a target.

process noise The part of the process model that represents the uncertainty in that model.

sensor assignment *see* measurement to sensor assignment

sensor assignment probability The probability that a particular measurement was produced by a specified sensor model.

sensor fusion The combination of measurements from multiple sensors into a single entity.

sensor measurement probability The probability that a measurement is produced by a particular sensor model, i.e., the fraction of total measurements produced by a sensor model at a particular time.

sensor models Virtual sensors used to represent the characteristics of a physical sensor under specific operating conditions.

sensor registration The spatial and temporal alignment of sensors to ensure that each target appears at the same position and time in each sensor.

sensor sets A group of sensor models common to a particular physical sensor.

similar sensors Sensors that produce the same type of measurement and have the same measurement functions, statistically equivalent noise and similar physical measurement characteristics.

state estimate Estimated value of the target's state.

state estimate error The error between an estimate of a target's state and its true value.

state transition matrix A matrix describing how a target's dynamic state propagates over time.

synchronous sensors Two or more sensors that consistently produce simultaneous measurements.

target assignment *see* measurement to target assignment

target assignment probability The probability of a measurement originating from a particular target.

target assignment weight The weighting factor applied to a measurement by a particular target model.

target association probability *see* target assignment probability

target measurement probability The probability of a measurement originating from a particular target, i.e., the fraction of total measurements at a particular time originating from the specified target.

target models A representation of a target containing a dynamic model and appropriate measurement models.

target state A vector containing the current dynamic state or condition of a target.

target track The complete set of estimated states of a target over a time interval of interest. Also referred to as the track history.

track error The error between the estimated target state and the true state of the target.

track error covariance Covariance of the error between the estimated target states and their true values.

track level fusion The combining or fusing of target tracks from multiple sensors into a single composite track.

tracker *see* tracking filter

tracking filter An algorithm that estimates the state of a target or system from noisy measurements.

validated measurements Those measurements that are located within a validation region, and may be used for tracking.

validation region A region in the immediate vicinity of the predicted target position whose space is defined by some appropriate criteria. Only those measurements located within the validation region are considered for tracking purposes.

variable update rate Kalman filter A Kalman filter operating on measurements irregularly spaced over time.

weighted least squares A least squares algorithm where the individual error terms are weighted according to some criteria.

Publications

1. M L Krieg and D A Gray. Track fusion in the presence of an interference. In *Proc. 4th Int. Symp. on Signal Processing and its Applications (ISSPA-96)*, volume 1, pages 192–5, Gold Coast, Qld, Australia, Aug 1996.

2. M L Krieg and D A Gray. Multi-sensor probabilistic multi-hypothesis tracking. In *Proc. 1st Australian Data Fusion Symposium (ADFS-96)*, pages 153–8, Adelaide, SA, Australia, Nov 1996.

3. M L Krieg and D A Gray. Performance of state space multi-sensor track fusion with model mismatch. In *Proc. 1st Australian Data Fusion Symposium (ADFS-96)*, pages 1–6, Adelaide, SA, Australia, Nov 1996.

4. M L Krieg and D A Gray. Radar and optical track fusion using real data. In *Proc. 1st Australian Data Fusion Symposium (ADFS-96)*, pages 25–30, Adelaide, SA, Australia, Nov 1996.

5. M L Krieg and D A Gray. Comparison of probabilistic least squares and probabilistic multi-hypothesis tracking algorithms for multi-sensor tracking. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP-97)*, volume 1, pages 515–8, Munich, Germany, Apr 1997.

6. M L Krieg and D A Gray. Multi-sensor probabilistic multi-hypothesis tracking using dissimilar sensors. In *Acquisition, Tracking and Pointing XI*, volume 3086 of *Proceedings of the SPIE*, pages 129–138, Orlando, FL, USA, Apr 1997.

7. M L Krieg. Asynchronous single platform sensor fusion. Technical Note DSTO-TN-0084, Defence Science and Technology Organisation, Australia, Salisbury, SA, Australia, Jun 1997.

8. M L Krieg and D A Gray. Comparisons of pmh and pls trackers on real and simulated data. Accepted *EURASIP Signal Processing*, Nov 1997.

CHAPTER 1

Introduction

In surveillance applications, target tracking techniques are used to obtain an estimated target trajectory or track history for each target in a region of interest. Measurements from one or more sensors, such as radar or infrared, provide the information necessary to estimate these tracks. Applications that use target tracking include air traffic control, maritime surveillance and building security.

Automatic target tracking has traditionally been restricted to using measurements from a single sensor. Although multi-sensor tracking has only received significant publicity in recent years, it has been in use for much longer. Single sensor target tracks from different sensors have traditionally been combined or fused by human operators. These operators effectively performed the central processing function for multi-sensor systems. For example, the pilot of a fighter jet monitors the radar and infrared displays and uses the information from both to determine the position and possible identity of other aircraft. The ability of humans to successfully perform this task is not surprising, as the human brain continuously combines information from the body's senses.

Recent interest in multi-sensor systems has centred on automating the fusion process. This reduces the workload of the operator and, through suitable processing and display techniques, can provide the operator with more information in a format that is readily absorbed.

A key problem when using multiple sensors is determining which measurements to use for tracking a particular target. This problem, known as the *data association* problem, is not unique to multi-sensor tracking, but appears whenever measurements may have been produced by objects or phenomena other than the target of interest. When tracking single or widely spaced targets in the absence of significant clutter or other noise, it may often be assumed that all measurements originate from the target of interest. In this case, the measurement volume of the sensor and/or restrictions in the tracking filter algorithm, e.g., gating, limit the available measurements to those near the expected target location, effectively eliminating the need for explicit data association. As the environment becomes more dense through the introduction of other targets and clutter, significant

numbers of measurements within the sensor's measurement volume or tracking gate may have originated from other targets or clutter sources. To successfully track the target (or targets) of interest, it is essential to determine which of the available measurements belong to the target (or targets) and are suitable for tracking.

The data association problem is compounded by using measurements from multiple sensors, particularly when the sensors are of different types. Different sensors obtain measurements through different physical processes, and therefore one sensor may see a particular target that remains undetected by another sensor. For example, a radar may not see an approaching low altitude super-sonic missile because of its low radar cross-section at this aspect and the clutter that is evident at low elevation angles. However, it may be detected by an infrared sensor because of its high skin temperature caused by friction with the atmosphere. The failure of a particular sensor to detect a target may also be caused by the target being outside the detection range of that sensor. A further problem relates to the different dimensionality of the measurement volume for different sensors. This is particularly evident when using both active (e.g., radar) and passive (e.g., infrared) sensors. The active sensors are often able to discriminate in angle (azimuth and elevation) and range, providing a restricted three dimensional measurement region. However, passive sensors are usually only able to provide angular discrimination and, because they can't separate targets in range, all targets that fall inside the angular field of view and the detection range will potentially produce measurements. This makes it very difficult to determine which of the multiple targets detected by the passive sensor is actually the one observed by the active sensor.

1.1 Motivation

The Tactical Surveillance Systems Division (TSSD) of the Defence Science and Technology Organisation (DSTO) Australia is interested in improving target tracking by combining radar measurements with measurements from other sensors. Therefore it has fitted its Generic Pulse Doppler Radar (GPDR) with an optical video camera and tracking system to collect multi-sensor data. This led to the development of the asynchronous fused Kalman filter (AFKF) for tracking targets using asynchronous measurements from the radar and optical tracker.

When designing the AFKF, the choice of models and their parameters proved to be important. Real physical objects behave differently to the approximations represented by these models. The effect of these differences on tracking performance deserves consideration when selecting the models and their parameters. This provided the incentive to investigate the effect additional sensors have on a system's sensitivity to model errors.

Evaluating the AFKF with real data revealed that not all of the available measurements should be used for tracking, especially when other targets or clutter are present. In particular, the optical sensor, which has no range discrimination, was often seduced

from the target of interest by other targets and clutter. The AFKF weighted the optical angular measurements significantly higher than those from the radar, because the optical sensor had better angular resolution and lower noise than the radar. Therefore the AFKF was also seduced from the target of interest and, because the sensors followed the new target, the original target eventually left the measurement region of both sensors, causing track loss. This problem provided the motivation to develop multi-sensor multi-target association and tracking algorithms.

1.2 Overview of Sensor Fusion

Data fusion is the combination of information from multiple sources into a single entity. This information may take various forms, including physical measurements, audio messages, computer records, images and written documents. Some possible sources of information are physical sensors, magnetic tape, databases and people.

Data fusion can be viewed as a hierarchical structure, as illustrated in the proposal by the Data Fusion Sub-panel (DFS) for the Joint Directors of Laboratories (JDL) in the United States Department of Defense (figure 1.1). This widely accepted proposal introduces three successively higher levels of fusion, with sensor management providing feedback for sensor control.

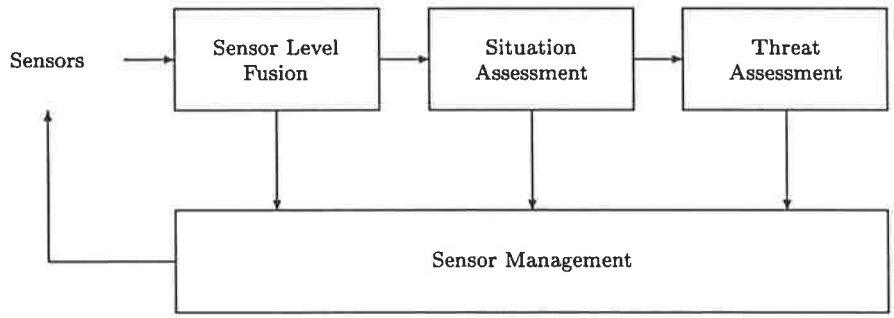


Figure 1.1: US-JDL data fusion levels

Sensor level fusion (level 1) refers to the combining or fusing of information at the sensor or source level. This level deals directly with sensor measurements and information from other sources. Detection, parameter estimation, tracking and identification are among the tasks found at this level.

Situation assessment (level 2) occurs when outputs from level 1 fusion, e.g., target tracks, are processed. Instead of measurements and raw information, objects such as aircraft, vehicles and people are introduced, and the relationships and interactions between these objects are analysed. This information is used to build a picture or model of the environment and determine what the objects are doing and how they are doing it.

In *threat assessment* (level 3), the behaviour of the objects from level 2 is analysed.

This level is concerned with the reasons for, and the significance of, the behaviour of the objects. This information may be used to make decisions and initiate responses.

Sensor management uses the outputs from the above levels to control and configure the individual sensors and information sources. This optimisation aims to maximise the system's performance for a specific application and operating environment. Sensor management tasks include emission control, sensor cuing and scheduling.

Sensor fusion is the combination of measurements from multiple sensors into a single entity. From this definition, it is obvious that sensor fusion is a restricted case of the more general data fusion, where the available information is limited to measurements from physical sensors. Therefore sensor fusion tasks fit into level one fusion because this is the only level that deals directly with measurements. However, level 1 fusion is not restricted solely to sensor fusion, because information other than sensor measurements, e.g., intelligence reports, may be processed at this level.

Sensor fusion is used in many applications, including tracking, identification, manufacture and assembly, surveillance and robotics. It is not a technology in its own right but an extension of the existing technologies used by these applications, e.g., filtering, artificial intelligence, neural networks, estimation theory and wavelet theory.

Similar sensors produce the same types of measurements (e.g., bearing only, bearing and range, etc.) and have similar characteristics (i.e., noise and measurement models). Fusing measurements from similar sensors does not provide any new or different information, instead it provides additional measurements of the same type. These additional measurements improve system performance by increasing track accuracy, and they improve overall system reliability through sensor redundancy. *Dissimilar sensors* provide additional information through different measurement types or different characteristics such as measurement accuracy and resolution. For example, a radar may provide accurate range and low resolution angle measurements, and an infrared sensor may provide high resolution angle measurements. Fusing these two sensors provides accurate range and high resolution angle measurements, more information than either sensor is capable of providing in isolation.

A suite of sensors can operate either synchronously or asynchronously. Synchronous sensors have the same measurement update rates and are synchronised so that their measurements occur at the same times. Sensors in an asynchronous system may have different update rates or just operate with independent timing. In most real applications, dissimilar sensors operate asynchronously. In this case, it is not really the individual measurements that are fused but the data streams as a whole.

Fusion systems may be *centralised* or *distributed.* In centralised systems, all fusion is performed at a central location using information received from every sensor node. A distributed system has no central node; the fusion takes place at every sensor node, using information received from every other node. The centralised system is subject to total system failure if the fusion centre fails, whereas the distributed system can still

operate, albeit at reduced capability, if one node is lost. However the distributed system requires substantial processing resources at all nodes, not just at a central node, and the communication channels must provide bi-directional data transfers between every node. In practice, hybrid schemes between these two extremes may be employed.

The actual fusion may take place at either the *track* level or the *measurement* level. In track level fusion (figure 1.2), each individual sensor produces target tracks using only its own measurements. It is these tracks from each sensor that are fused at either the central node or each of the sensor nodes, depending on whether the fusion is centralised or distributed. When fusing at the measurement level, the measurements are sent to

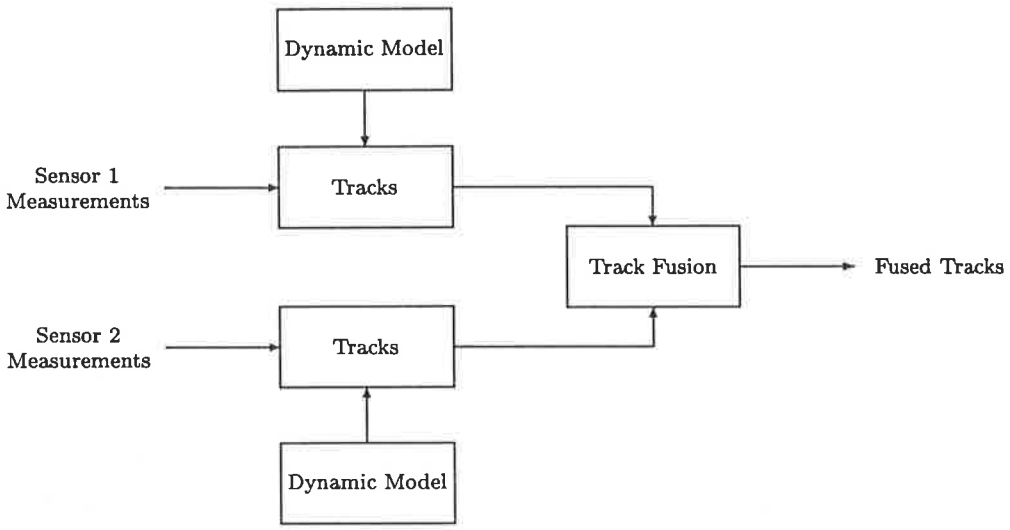


Figure 1.2: Track level fusion

either the central node or every sensor node for processing. As shown in figure 1.3, the multi-sensor measurements are used together with a single dynamic model to produce the fused tracks.

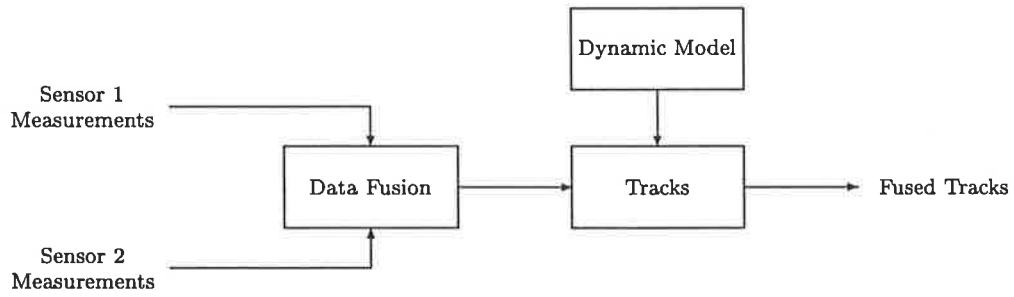


Figure 1.3: Measurement level fusion

The multi-sensor tracking task may be separated into three distinct phases, sensor registration, data association and data combination (figure 1.4). In this context, the data may represent either tracks for track level fusion or measurements for measurement level fusion.

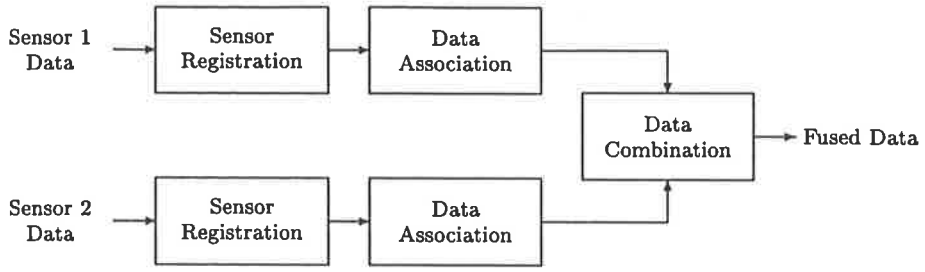


Figure 1.4: Fusing data from two sensors

Sensor registration, also known as *positional fusion*, aligns the data from each sensor both spatially and temporally. It ensures that all sensors see the same target at the same point in space and time. This task is generally simpler for single platform configurations, because the relative positions and measurement latencies of all sensors are known.

Data association determines which measurements originate from a particular target and therefore may be used for tracking that target. As incorrect measurement association can easily result in loss of track, this is probably the most important (and difficult) part of the fusion process.

Data combination is the combination of the data into a single entity. In tracking applications, the measurements from all sensors are combined into target tracks.

Multi-sensor tracking provides a number of benefits. These include improved track accuracy from the additional measurements supplied from the additional sensors and greater system reliability through the redundancy provided by multiple sensors. The use of dissimilar sensors improves system robustness to clutter and countermeasures. Active emissions may be reduced by tracking with passive sensors and using the active sensors for short periods to update the range estimates. However multi-sensor tracking requires additional sensors and increased processing capability. This increases the system complexity, resulting in increased development and hardware costs and increased maintenance.

1.3 Thesis Outline and Contributions

This thesis is primarily concerned with multi-sensor tracking. A variable update rate Kalman filter is used to track a single target using asynchronous measurements from two sensors. The operation and performance of the filter is investigated using simulated and real data.

> **The first key contribution of this thesis is the application of an asynchronous Kalman filter to the fusion of real data from a radar and optical sensor.**

The above investigations highlighted the need for data association and thus led to a

multi-sensor multi-target problem formulation for solving both the data association and state estimation problems. Two multi-sensor multi-target algorithms were developed as potential solutions, and their respective performances were evaluated and compared.

The second key contribution of this thesis is the development and evaluation of two multi-sensor multi-target association and tracking algorithms, the multi-sensor Probabilistic Multi-Hypothesis Tracking and multi-sensor Probabilistic Least Squares Tracking algorithms.

The content of this thesis is described below. The numbers in parentheses refer to publications as listed on page xxxi.

Relevant background information immediately follows this section in chapter 2. It provides an overview of sensor fusion, as it is related to tracking applications, and an extensive overview of the difficult problem of data association. Various techniques for the association of measurements to targets are discussed in a tracking context.

Chapter 3 provides an overview of the Kalman filter and introduces a multi-sensor variable update rate variation known as the asynchronous fused Kalman filter (4). The results obtained from evaluating this algorithm on both real and simulated data are presented. Tests on real radar and optical data highlighted the problem of track seduction by clutter and other targets and reinforced the need for data association.

Contribution: *The formulation and evaluation of a multi-sensor single target Kalman filter tracking algorithm using real data containing radar and optical measurements from multiple targets and clutter.*

The effect of mismatch between the Kalman filter's models and the data, i.e., sensitivity analysis, is well known. Chapter 4 looks at the effect that adding a second sensor to a single sensor Kalman filter tracking system has on the system's sensitivity to model errors. The effect of mismatch in process noise covariance and measurement noise covariance is investigated through simulations and theoretical calculations (3).

Contribution: *Analysing the effect of adding an additional sensor to a Kalman filter's sensitivity to errors in its process and measurement noise covariances.*

The multi-sensor multi-target tracking problem is formulated in chapter 5 for both a simple restricted problem, where each sensor produces exactly one measurement in each measurement scan, and a more general asynchronous tracking problem. This is based on extending the formulation used in the Probabilistic Multi-Hypothesis Tracking algorithm

for multi-sensor tracking. This chapter also contains a more general formulation that provides each physical sensor with multiple sensor models to facilitate tracking under changing operating conditions.

The derivation, for both the restricted synchronous and general asynchronous problems, of the multi-sensor Probabilistic Multi-Hypothesis Tracking (msPMHT) algorithm is presented in chapter 6. This algorithm estimates both the target states and the measurement to target assignments for multiple targets using maximum likelihood techniques (1, 2, 6). It was derived by extending the Probabilistic Multi-Hypothesis Tracking (PMHT) algorithm for multi-sensor tracking.

Contribution: *The development of the multi-sensor Probabilistic Multi-Hypothesis Tracking (msPMHT) algorithm by generalising the PMHT algorithm for tracking multiple targets using measurements from multiple sensors.*

Chapter 6 also contains the derivation of the general msPMHT algorithm (2). This algorithm is an extension of the msPMHT that associates measurements to sensor models, allowing the selection of the sensor model most appropriate for the operating conditions at that time.

Contribution: *The development of the general msPMHT algorithm for multi-sensor multi-target tracking with multiple sensor models to reflect changing operating conditions.*

A similar algorithm, the multi-sensor Probabilistic Least Squares Tracking (msPLST) algorithm, is introduced in chapter 7 to solve the synchronous and asynchronous multi-sensor tracking problems (5). The msPLST is a new algorithm that uses the same problem formulation as the msPMHT, but it uses least squares to estimate the target states and measurement to target assignments.

Contribution: *A new algorithm, the multi-sensor Probabilistic Least Squares Tracking (msPLST) algorithm, for multi-sensor multi-target tracking using least squares techniques.*

In chapter 8, the performance of both the msPMHT and msPLST algorithms are evaluated using simulated and real data (5, 8). The structure of the two algorithms are compared, and their respective performances are evaluated and compared against each other and the fixed interval Kalman smoother.

Contribution: *The evaluation and comparison of the msPMHT and msPLST algorithms and their respective performances. Comparison of both algorithms with the fixed interval Kalman smoother for tracking single targets in multi-target environments.*

The main body of the thesis is completed with the conclusions in chapter 9. Additional information is contained in the appendices as listed below.

1. The sensor fusion testbed used to collect the radar and optical data is described in appendix A.

2. Appendix B contains the derivation of the fixed interval Kalman smoother as the solution to the set of tri-diagonal equations used to obtain the target state estimates in the linear Gaussian msPMHT and the msPLST algorithms.

CHAPTER 2

Background

Tracking is the process of estimating the state or parameters of an object or system over a time interval of interest, using the available information or measurements. This thesis is primarily concerned with the tracking of moving or dynamic objects such as aircraft in air traffic control or maritime vessels in harbour surveillance. In these applications, estimates of the dynamic state (position, velocity, etc.) of each object or target are recorded over time. The result is a *target track*, also known as a *target trajectory* or a *track history*. The state estimator is referred to as a *tracking filter* or simply as a *tracker*.

Various tracking filters are available, but most interest in recent times has centred on the *Kalman filter* for single sensor tracking. Many single target trackers reduce to a Kalman filter when restricted to measurements from only the target of interest, e.g., the probabilistic data association filter (section 2.2.3). Other more complex algorithms contain embedded Kalman filters, in particular for estimating the actual target states, e.g., multiple hypothesis tracking (section 2.2.6).

A brief description of the Kalman filter algorithm may be found in section 3.1. Bar-Shalom and Fortmann (1988), Gelb (1992) and Jazwinski (1970) provide a few of the many available detailed descriptions and derivations of the Kalman filter and the underlying estimation theory.

2.1 Fusion of Multi-sensor Track Information

The fusion or combination of track information from multiple sensors may take place at the *track level* or the *measurement level* (*see* section 1.2). The method selected for a particular application depends on a variety of factors such as system hardware, communication bandwidths, processing power at each sensor and the central fusion processor (*fusion centre*) and the physical layout and separation of the sensors and fusion centre.

This section is primarily concerned with tracking a single target, or tracking widely separated targets individually on a target by target basis. However, the general concepts introduced may be applicable to tracking closely spaced multiple targets.

2.1.1 Track Level Fusion

When fusing at the track level, individual tracks are independently formed from the measurements received from each sensor (*see* figure 1.2 on page 5). This process usually occurs locally at the sensor and is often performed by a Kalman filter. These single sensor tracks are combined into a single track at a fusion centre. The tracks may be viewed as processed or compressed data and, as such, require less communication bandwidth. However some of the original information is lost.

Early attempts at track level fusion assumed that the state estimate errors (state errors) were uncorrelated between sensor tracks, i.e., the individual tracks were independent (Bar-Shalom and Fortmann 1988, section 10.2). The combined track was a weighted sum of the individual tracks, where the weights were determined from the inverse of the state error covariances. For example, two tracks $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ with state error covariances $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ were combined into a single track $\mathbf{X}^{(f)}$ as

$$\mathbf{X}^{(f)} = \left(\mathbf{P}^{(1)\,-1} + \mathbf{P}^{(2)\,-1}\right)^{-1} \left(\mathbf{P}^{(1)\,-1}\mathbf{X}^{(1)} + \mathbf{P}^{(2)\,-1}\mathbf{X}^{(2)}\right).$$

Bar Shalom recognised that each track for a particular target shared the same process noise, and therefore such tracks are not independent. He formulated an optimal technique for combining these correlated tracks in which the correlation is accounted for by cross-covariance matrices (Bar-Shalom 1981, Bar-Shalom and Campo 1986). Saha (1994) derived the necessary constraints to ensure that the cross-covariance matrices are positive definite.

2.1.2 Measurement Level Fusion

Measurement level fusion involves sending all the measurements from every sensor to the fusion centre that then combines all the measurements into a single track (*see* figure 1.3 on page 5). The use of a single tracking filter overcomes the correlation problem because a single dynamic or process model is assumed. This method is computationally more demanding on the fusion centre and requires greater communication bandwidth because more information is transmitted from the sensors. However, the availability of this extra information improves tracking performance in regard to minimising the state error covariance, although the level of improvement depends on the ratio of the measurement noise from each sensor (Roecker and McGillem 1988). The combination of simultaneous multiple measurements at any time may be performed through parallel processing of measurements (the measurements are stacked into a single measurement vector), data compression (the measurements are combined into a single composite measurement), or sequential processing (the measurements are processed individually, separated by a zero prediction time interval) (Willner, Chang and Dunn 1976). Using the data compression method, measurements from dissimilar sensors may be combined by creating *dummy* mea-

surements with very large noise covariances to give the measurements from each sensor the same dimensionality. For example, if a radar produces range and bearing measurements and an optical sensor produces only bearing measurements, the measurements from both sensors can be combined by creating a dummy optical range measurement. The value of this dummy range measurement is typically equal to the predicted range, producing a measurement error of zero. The noise covariance of the dummy measurement is made much larger than the covariance of the radar range measurement, indicating very little confidence in the dummy measurement. The contribution of each measurement to the compressed measurement is inversely proportional to its (normalised) noise covariance, therefore the contribution of the dummy optical range measurement will be insignificant.

It is measurement level fusion that is addressed in the work covered by this thesis.

2.1.3 Dissimilar Sensors

Dissimilar sensors may differ in the types of measurements that they provide, e.g., a radar may provide bearing and range measurements and an infrared (IR) sensor may provide bearing and elevation measurements. Generally they also differ in the assumed noise models, e.g., the IR sensor has a much greater angular resolution than the radar and hence a potentially lower measurement noise covariance.

Haimovich, Yosko, Greenberg, Parisi and Becker (1993) considered the problem of fusing dissimilar sensors, i.e., radar and IR. They found that as the difference in sensor accuracy increased, the fused system approached the single sensor operation of the more accurate sensor. Consequently the benefit gained by fusing at the measurement level over fusing at the track level decreased.

They also considered the problem of tracking clusters of targets, where only one of two sensors is able to resolve the individual targets within the cluster. They suggest correlating all high resolution tracks within a gate to the low resolution track, using the resulting centroid as the (single) fused track. Alternatively, each target within the cluster may be individually tracked, with the low resolution track contributing to each.

Measurements from different sensors may be offset from one another when targets are at close range and occupy a significant portion of the sensors' fields of view. For example, radars provide measurements of the *centre of reflection* which is extremely aspect dependent and may not always be on the target. However, imaging sensors provide *centre of mass* measurements based on the centroid of the target (Romine, Kamen and Sastry 1994). Therefore, although the same range may be assumed, angular measurements from a radar and imaging sensor may differ even though they correspond to the same target. Fusion can be advantageous in this situation, particularly as the radar's angular offsets become observable, and may be estimated, when the the imaging sensor is included.

2.1.4 Asynchronous Sensor Fusion

Most theoretical sensor fusion work has centred on combining simultaneous measurements from multiple sensors. However different types of sensors have different measurement update rates, e.g., IR sensors may have measurement rates that are an order of magnitude greater than those of radars. Therefore simultaneous measurements are unlikely if dissimilar sensors are used.

Blair, Rice, Alouani and Xia (1991) fused the measurements from both a radar and an optical sensor. The optical sensor's measurement update rate was much higher than the radar's. The radar was capable of tracking multiple targets and therefore did not always provide periodic measurements. Using a sub-optimal approach, they combined a block of optical measurements into a single *optical datum* that was used to update the tracking filter. If a radar measurement was available, it was fused with the optical datum before updating the tracking filter. The combination of measurements was based on a weighted sum, where the weight for each measurement was inversely proportional to the covariance of that measurement. It was necessary to compress the optical data to reduce the computational load because of the optical sensor's high measurement update rate.

2.2 Data Association

Data association is any process used to determine which measurements to use for tracking a particular target. It is the most difficult problem in sensor fusion, and it is critical to the success of any tracking application. The problem is not unique to sensor fusion; it is also important for tracking in the presence of clutter and countermeasures, and in multi-target tracking in general. In fact, the sensor fusion problem is really only a further complication in these other applications.

Many techniques have been used with varying degrees of success. Usually a particular technique is successful under certain conditions but is out performed by other methods when the conditions change. Often such a technique is designed for a specific application, e.g., tracking a single target in clutter or tracking multiple targets in a dense environment. Methods for associating measurements in both single and multi-target environments are investigated in this section.

Many of the approaches listed below incorporate track estimation with the data association, often iterating between the two in an attempt to obtain an appropriate solution. The list below is by no means exhaustive, but it does represent a reasonable cross-section of the available techniques.

2.2.1 Nearest Neighbour Data Association

The *nearest neighbour* algorithm (Bar-Shalom and Fortmann 1988, section 6.2) is the simplest form of data association. The measurement that is closest in some sense to

the predicted state of a particular target is used by the tracking filter to estimate that target's state. A distance metric is usually chosen to determine which measurement is closest. Gating, where only those measurements within a *validation region*[1] about the predicted location of the target are considered, may be used to restrict the problem to those measurements in the immediate vicinity of the target. This allows for the occurrence of missed detections, i.e., no measurements from the target at that time.

No provision is included for incorrect measurement assignments in the filtered error covariance, therefore track loss is likely to occur.

2.2.2 Optimal Bayesian Approach

The optimal Bayesian approach associates complete sequences of measurements, up to the current time, to a single target track (Bar-Shalom and Fortmann 1988, section 6.5). The probability of each possible measurement sequence is calculated, and each measurement within a sequence is weighted by the probability for that sequence. At each measurement time, a combined measurement is produced by summing and normalising each weighted measurement at that time, and this combined measurement is used by a Kalman filter to produce a state estimate for the target at that time. This concept may be extended to multiple targets, where the probabilities of groups containing one or more non-overlapping sequences of measurements are considered.

This algorithm grows exponentially over time, making it infeasible, in practice, for any reasonable length batch of measurements. Therefore sub-optimal algorithms of reduced complexity must be considered. Usually these algorithms only operate on the latest N measurement times (N-scan back), where track histories that are identical during this time are merged.

2.2.3 Probabilistic Data Association

Probabilistic data association (PDA) is a sub-optimal Bayesian approach that considers only the *validated measurements*[2] in the current *measurement scan*, i.e., only the measurements occurring at the current time (a 1-scan back algorithm). The measurement to target association probability, i.e., the probability that each single measurement (and not the complete measurement sequence, as in the optimal approach) originates from the target of interest, is calculated for each measurement (Bar-Shalom and Fortmann 1988, section 6.4). These probabilities form the basis for the *probabilistic data association filter* (PDAF). This tracking filter tracks a single target in the presence of false alarms or clutter and, in simple terms, is a Kalman filter that uses a combined measurement (or

[1] A validation region is a region in the immediate vicinity of the predicted target position that is defined by some relevant criteria, e.g., elliptical or elliptical with missing or illegal sub-regions.

[2] Those measurements within the validation region.

measurement innovation). This combined measurement is a weighted sum of the validated measurements, where each weight is the appropriate measurement to target association probability. These probabilities may be based on either a parametric (Poisson) or a non-parametric (diffuse prior distribution) model of the clutter.

The interacting multiple model PDAF (Bar-Shalom 1990, Dufour and Mariton 1991a, Dufour and Mariton 1991b) combines the *Probabilistic Data Association* (PDA) and *Interacting Multiple Model* (IMM) tracking filters for fusing measurements from multiple passive sensors. The IMM filter contains a number of target models, each representing different dynamic scenarios, e.g., constant velocity, a slow clockwise turn, etc. Each model has a probability that is continually updated using Markovian *a priori* transition probabilities. The new state estimate for each model, and its covariance, are formed using the state estimates and covariances from all models, hence the term interacting multiple models. The association is performed by the PDA that combines all validated measurements within a scan according to their measurement to target probabilities.

The problems of track creation, confirmation and termination have been addressed by Musicki and Evans (1992). They treated track existence as an event with a probability that can be calculated. Their *Integrated* PDAF (IPDAF) uses this additional information when determining the association probabilities. These probabilities are applied to a PDAF in the same way as the standard PDAF.

Houles and Bar-Shalom (1989) have developed a multi-sensor PDAF (MSPDAF) algorithm using a non-parametric PDA. Simultaneous measurements from the sensors are applied to the filter sequentially, i.e., the measurements from one sensor are used to correct the predicted target state, and this new state estimate is used as the predicted measurement for the next sensor's measurements. Pulford and Evans (1996) have also developed a multi-sensor PDAF for tracking a single target in clutter using measurements from more than one sensor.

The PDAF state estimate error covariance contains a term representing the square of the innovations. This component grows during manoeuvres and may be used to determine when more information, or additional measurements, would be beneficial. Using non-uniform sampling, more measurements may be obtained during manoeuvres, and the measurement rate can be reduced during periods of constant velocity when the target behaviour is predictable (Ahmeda, Harrison and Woolfson 1996).

2.2.4 Joint Probabilistic Data Association

When multiple targets are sufficiently separated so that their validation regions or gates do not overlap, a separate PDAF may be used to track each target. However, if any of the targets have overlapping validation gates, then these targets may be tracked using the joint probabilistic data association filter (JPDAF) (Bar-Shalom and Fortmann 1988, section 9.3). The JPDAF is a generalisation of the PDAF for multiple targets, and differs

from the PDAF in the way that the association weights are determined. In the JPDAF, these probabilities are calculated jointly across the targets using the whole surveillance region, i.e., no gating. To reduce the computational complexity, validation gates are used to select only the feasible joint events; the low probability events are ignored. As in the PDA, both parametric and non-parametric versions of the algorithm exist.

As the number of targets increases, the JPDA becomes computationally expensive. To avoid this problem, the *cheap JPDA* (Bar-Shalom 1990, section 1.2.1) has been developed. It involves using a simple *ad hoc* formula to approximate the association probabilities. In another attempt to reduce complexity, Zhou and Bose (1995) efficiently compute the *a posteriori* probabilities of the measurement origins by decomposing the process into two parts, one of which is trivial.

The concept of a *N-scan back* JPDA algorithm has been introduced by Korona and Kokar (1995). This computationally expensive algorithm uses the last N measurement scans, instead of just the most recent. Using a sliding window of N scans, it is a combination of the multiple hypothesis tracking (section 2.2.6) and JPDA algorithms.

Bar-Shalom, Chang and Blom (Bar-Shalom 1992, chapter 4) use an interacting multiple model JPDA algorithm to track splitting targets in cluttered environments. The interacting multiple models provide the mechanism for tracking the manoeuvres of aircraft as they break formation, or tracking missiles as they are launched from an airborne platform.

2.2.5 The Track Splitting Filter

The optimal Bayesian approach (section 2.2.2) considered the whole batch of measurements up to the current time, partitioning it into likely sequences representing target tracks. An alternative is to generate the possible measurement sequences by splitting each existing sequence at each measurement scan into new sequences, one for each measurement in the scan, i.e., each new sequence consists of the existing sequence and a single measurement from the current scan. The probability that each hypothesised measurement sequence represents an actual target track is calculated and, in practice, this may be used to determine the most likely target tracks.

One such approach is the track splitting filter where, at each measurement time, each track is split into separate hypothesis tracks, one for each measurement in the validation region centred on the predicted measurement location (Bar-Shalom and Fortmann 1988, section 6.3). Each of these hypothesis tracks are then propagated forward, a new validation region is established, and the process is repeated. As each track is considered individually, a single measurement may be assigned to more than one track.

To overcome the exponential growth of this algorithm, low probability tracks are removed, and a sliding window is often used to prevent old measurements dominating the hypothesised track probabilities.

2.2.6 Multiple Hypothesis Tracking

The track splitting filter considers the probability that a measurement is assigned to a particular existing track, i.e., it is track oriented. Multiple hypothesis tracking (MHT) is measurement oriented, i.e., it evaluates the probability that an existing or new target produced the measurement. Therefore each measurement is seen as a false alarm or a potential candidate for updating any existing track or initiating a new track. Every possible combination of measurement to target assignment is considered (Reid 1979, Bar-Shalom 1990), and each such combination is hypothesised as a potential partition of the measurements into target tracks. This method is suited to both single and multiple target tracking.

The probability of each track is calculated, and hypothesised tracks with low probabilities are pruned to control the exponential growth in complexity. The complexity may also be reduced by employing a N-scan back approach, where only the N most recent measurement scans are considered. Hypotheses that have the same measurement to target associations during this period are combined. Similar hypotheses that have the same number of targets and similar state estimates may also be combined to reduce the complexity.

The MHT algorithm has been used for tracking single targets, or well separated multiple targets, in clutter using a phased array radar (van Keuk 1995).

Werthmann (1992) uses three concentric gates, centred on the predicted state of each hypothesised track, to restrict the choice of hypotheses and therefore reduce the complexity. The available choices vary depending upon where within the gates a measurement falls. For example, a measurement appearing inside the inner most gate is always assigned to that track, and measurements outside all the gates are assumed to be a new track. In other cases, various combinations of options are available, including initiate a new track, update the existing track or do not update the track. Even with this limited choice of hypotheses, track pruning is still required.

2.2.7 Maximum Likelihood

The combined data association and track estimation problem can be viewed as an incomplete data problem. To solve this problem, a complete set of measurements and measurement to target assignments is required. However the measurement to target assignments are missing, and the incomplete data set, containing just the measurements, is all that is available.

An unknown set of parameters defining the probability density function of the complete data is obtained by maximising the joint probability (likelihood) of the measurements and measurement to target assignments over all measurement scans with respect to these parameters. Using the estimated parameter set, the target states and measurement to target assignments are estimated by finding the values with largest *a posteriori* probability

(Avitzour 1992).

Gauvrit, Jauffret and le Cadre (1997) formulated the multi-target tracking problem as one of incomplete data and used the Expectation-Maximisation (EM) algorithm to obtain maximum likelihood estimates. The target states are assumed to be deterministic, i.e., they are not treated as random variables, so if the state is known at one time, it can be determined at all other times by using the (noiseless) dynamic model. Therefore the target state needs only be estimated at one (convenient) time. Modelling the target states as random variables with *a priori* distributions and using the same incomplete data formulation with the EM algorithm, they also developed a maximum *a posteriori* (MAP) algorithm that they recognised as being equivalent to the probabilistic multi-hypothesis tracking algorithm of Streit and Luginbuhl (1993).

2.2.8 Probabilistic Multi-Hypothesis Tracking (PMHT)

As described in the previous section, the probabilistic multi-hypothesis tracking (PMHT) algorithm formulates the data association and tracking problem as an incomplete data problem and solves it using the EM algorithm. Using soft or probabilistic measurement to target assignments, each measurement is partially assigned to each of a fixed number of targets (Streit and Luginbuhl 1993, Streit and Luginbuhl 1994, Streit and Luginbuhl 1995).

The concept of gating may be applied to the PMHT by using sub-models with different measurement noise covariances in the measurement model of each target (Rago, Willett and Streit 1995a). Also false alarms may be handled by a similar method, where each target has a second model using the same dynamics but a measurement model with measurement noise covariance equal to that of the innovation at that time (Rago, Willett and Streit 1995b). This second model is ignored when estimating the target states, effectively removing the effect of some measurements from the problem. Theoretical comparisons have been made with the JPDA (Streit and Luginbuhl 1995) and the comparative performance of the PMHT and JPDA has been analysed (Rago et al. 1995a, Rago et al. 1995b).

The PMHT uses pointers to assign each measurement in each scan to a target. An alternative approach is to use pointers to assign each target to a measurement, with the possibility of empty pointers for missed detections (Rago, Willett and Streit 1995c). All measurements that have no targets assigned to them are assumed to be false alarms, e.g., clutter.

The PMHT assumes each measurement belongs to exactly one target, but each target may produce multiple measurements at any time. Therefore it lends itself naturally to sensor fusion (Rago et al. 1995b). Dunham and Hutchins (1997) have evaluated the performance of the PMHT in the presence of significant clutter.

2.2.9 Assignment Algorithms

The general assignment problem is one of matching n people to m objects. In a data association and tracking context, this is a problem of matching measurements to target tracks or, conversely, matching target tracks to measurements. For explanation purposes, the problem of matching measurements to target tracks will be used here, unless otherwise stated.

There is a benefit associated with matching a particular measurement to a specific target track, and the aim is to select the set of measurement to target assignments that maximise the total benefit over the whole set of assignments. In tracking applications, the benefit may be the probability of a measurement originating from a specific target, or the penalty (to be minimised to obtain the maximum benefit) may be the error between a measurement and a target track. This maximisation (or minimisation for penalties) is usually subject to some constraints, such as each measurement being assigned to exactly one target or limiting the choice of measurements that may be associated to a particular target track, e.g., validated measurements. A feasible solution is a set of measurement to target assignments that satisfy the constraints, and the optimal solution is the feasible solution that provides the highest overall benefit.

Assignment algorithms belong to the broader class of minimum cost flow problems that, in addition to the assignment problem, includes transport and maximum flow problems. These problems may be solved using methods incorporating one or more of the three basic algorithmic ideas, namely primal cost improvement, dual cost improvement and auction (Bertsekas 1991).

2.2.9.1 Primal Cost Improvement

Consider a hypothetical measurement scan containing N measurements that are to be assigned to N target tracks, such that each measurement is assigned to a single target and no two or more measurements are assigned to the same target. Each measurement to target assignment has a cost or penalty associated with it, and the primal cost improvement problem becomes

$$\arg\min_{\mathbf{x}} \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}x_{ij} \tag{2.1}$$

subject to

$$\begin{aligned} \sum_{i=1}^{N} x_{ij} &= 1 \qquad j = 1, 2, \ldots, N \\ \sum_{j=1}^{N} x_{ij} &= 1 \qquad i = 1, 2, \ldots, N \end{aligned} \tag{2.2}$$

where a_{ij} represents the cost (penalty) associated with assigning measurement i to target track j and $\mathbf{X} \equiv \{x_{ij} : i = 1, 2, \ldots, N, j = 1, 2, \ldots, N\}$ denotes the assignment variable for measurement i and target track j. x_{ij} has a value of 1 or 0, depending on whether measurement i is assigned to target track j or not.

These algorithms commence with a feasible complete set of assignments, and then the benefit or cost is improved iteratively by swapping assignments to obtain another feasible solution with a greater benefit. The algorithm terminates when the optimal or maximum benefit is achieved, i.e., maximum benefit or minimum penalty.

2.2.9.2 Dual Cost Improvement

Using linear programming theory, a problem can be replaced by its *dual* that operates on *prices* instead of assignments. There is a price for each measurement if the number of target tracks to which the measurement may be assigned is constrained, and a price for each target track if the number of measurements that may be assigned to it is limited. A measurement price represents the benefit obtained by not assigning that measurement to any target tracks, i.e., it is a false alarm, and a target track price represents the benefit obtained by assigning a measurement to that target track, i.e., a detection exists from that target. In linear programming, the prices are known as Lagrange multipliers, and they incorporate the original constraints on the assignments into the dual cost function, i.e., the prices are unconstrained, with the original assignment constraints now part of the cost function.

The dual cost problem may be given as

$$\min_{\mathbf{p}} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \max_{x_{ij}} \left\{ (a_{ij} + p_j - p_i)\, x_{ij} \,|\, x_{ij} \in \{0, 1\} \right\} + \sum_{i=1}^{N} p_i - \sum_{j=1}^{N} p_j \right\} \tag{2.3}$$

where $\mathbf{p} \equiv [p_i, p_j : i, j = 1, 2, \ldots, N]$ is the price vector. The term $a_{ij} + p_j - p_i$ in (2.3) represents the *profit* obtained by assigning the measurement i to the target track j, i.e., it is the benefit of assigning the measurement to the target track j plus the benefit of assigning a measurement to the target track, less the benefit of not assigning the measurement to any target track. The aim is to maximise the profits while minimising the dual cost.

The dual cost improvement algorithms commence with a complete set of *prices*, and attempt to iteratively obtain new prices that improve the dual cost. The price changes at each iteration occur along a particular direction, the steepest in *primal-dual* algorithms and along a direction with a small number of non-zero elements in *relaxation* or *coordinate ascent* algorithms. If the problem is feasible, the algorithm terminates when the assignments, that are related to the prices, become feasible. At this point, the solution is optimal, and the primal and dual costs are equal, i.e., if the primal cost function increases, the corresponding dual cost function will decrease, until eventually both are equal.

2.2.9.3 Auction

Auction algorithms (Bertsekas 1981) reach the optimal solution without necessarily improving the primal and dual costs at each iteration. They commence with a possibly empty partial assignment set, and at each iteration an unassigned measurement is selected and assigned to the target track that provides the maximum benefit. If that target track already has another measurement assigned to it, that measurement is removed from the partial assignment. At each iteration, a feasible partial assignment is produced, and the algorithm terminates when all the measurements have been assigned to a target track. The price (*see* section 2.2.9.2) of the target track to which the measurement has just been assigned is increased by the difference between the benefit of assigning the measurement to it and the benefit of assigning the measurement to next best choice of target track.

A problem with the auction algorithm is that it may cycle between two or more partial assignments and never terminate, i.e., the optimal solution is never reached. To overcome this, the price of each newly assigned target track is increased by the difference between the benefits of assigning the measurement to the best two target tracks *plus* a small positive increment ϵ. This ensures that the prices of continually reassigned target tracks increase until some other target track, with a lower price, is assigned to the measurement. Therefore the algorithm must always terminate because eventually all target tracks (and measurements) will be assigned.

In the *reverse auction* algorithm (Bertsekas, Castanon and Tsaknakis 1993), target tracks compete for measurements by either lowering their prices or increasing the profit of the measurement. Combining the forward and reverse auction algorithms produces the *combined forward and reverse auction* algorithm in which both the prices and profits are updated at every iteration. The algorithm commences with several iterations of the forward auction, and then it switches to the reverse auction for several iterations. This process continues until all the target tracks are assigned and the optimal solution is reached. However, as profits increase, prices decrease (and vice versa), and the algorithm may never terminate. To overcome this potential problem, the number of assigned measurement/target pairs must increase before the algorithms are switched.

When there are more measurements than target tracks (or vice versa), two approaches are possible (Bertsekas et al. 1993). In the *asymmetric assignment problem*, measurements are left unassigned (Castanon 1992), e.g., false alarms, and in the *multi-assignment* problem, multiple measurements may be assigned to a single target track. Both problems are solved by initially running the forward auction algorithm until all the target tracks have exactly one measurement assigned to them. Then, for the asymmetric assignment problem, the reverse auction is run until the price of every unassigned measurement is less than or equal to the price of every assigned measurement. In the multi-assignment problem, the reverse auction is run until all the measurements are assigned. If an unassigned measurement is assigned to a target track, the measurement previously assigned

to that target track is not removed if the target track's profit has reached a maximum limit (equal to the highest profit at the commencement of the reverse auction).

The Jonker-Volgenant-Castanon (JVC) algorithm was originally developed for *sparse cost matrices.* This algorithm consists of two phases, an *auction* algorithm phase followed by a modified *Munkres* algorithm (Munkres 1957), utilising the rapid initial convergence of the auction algorithm and the fast convergence of the Munkres algorithm in the final stages (Malkoff 1997).

2.2.9.4 Relaxation Algorithms

Algorithms, such as the auction, are able to solve the 2-D assignment problem in polynomial time. However, the 3-D and higher dimension assignment problems have exponential complexity. The general K-D assignment problem may be solved by relaxing one of the K constraints and incorporating it into the cost function using Lagrange multipliers. The problem then becomes one of solving the dual $(K-1)$-D assignment problem (Deb, Mallubhatla, Pattipati and Bar-Shalom 1990, Pattipati, Deb, Bar-Shalom and Washburn 1992, Deb, Pattipati and Bar-Shalom 1993, Poore and Rijavec 1991, Poore and Rijavec 1993). If the solution of the $(K-1)$-D dual problem is known, a solution for the primal K-D problem can be obtained, where the difference between the primal and dual cost, known as the *duality gap*, provides an indication of how close the solution is to optimal (if the solutions are equal, they are optimal). If the duality gap is too large (typically greater than 1%), the process is repeated using the Lagrange multipliers obtained from the previous iteration. This process may be recursively applied until the 3-D problem is reached. Then the 2-D dual problem may be solved by an algorithm such as the auction to obtain a solution to the 3-D problem (iterating until the duality gap is acceptable). Then each successively higher dimension problem is solved, until the solution to the original problem is obtained.

An alternative approach is to successively relax all but two of the constraints, putting each relaxed constraint into the objective function using Lagrange multipliers (Deb, Yeddanapudi, Pattipati and Bar-Shalom 1997, Poore, Robertson III and Shea 1995). The resulting 2-D problem is solved optimally, using the auction or similar algorithm, and then each of the relaxed constraints are enforced sequentially to obtain a feasible solution. If the duality gap is too large, the whole process is repeated, using the Lagrange multipliers from the previous iteration.

2.2.10 Other Approaches

Various other approaches to data association have been attempted, some of which are listed below.

Neural networks have been used to associate and fuse tracks from different sensors, e.g.,

fusing two radar tracks (Filippidis and Bogner 1992), and to associate individual measurements to target tracks (Wang, Litva, Lo and Bosse 1996).

Hidden Markov models and dynamic programming have been used extensively for frequency line tracking. This requires a discrete state space, and therefore it does not lend itself readily to target tracking in general. However, Martinerie and Forster (1992) have used this technique to track targets using measurements from three sensors that measure either range, or range and range rate.

Clustering techniques group measurements into clusters, and the centres of these clusters become pseudo measurements that are assigned to target tracks (Thompson, Parra-Loera and Tao 1991).

Fuzzy logic that compares measurements and targets is used to replace the correction phase of the Kalman filter (i.e., *fuzzy return processor*) (Horton and Jones 1995). This approach may be useful for tracking in cutter.

CHAPTER 3

Asynchronous Fused Kalman Filter

When tracking targets in real environments using measurements from multiple sensors, i.e., measurement level fusion (section 2.1.2), it is unlikely that the measurements will arrive simultaneously from each sensor. Each sensor in the system will probably have a different *measurement rate* with little or no synchronisation between sensors. Missed detections, caused by faint fluctuating targets or deliberate emission control, may cause further complications. Therefore the system's measurement rate is unlikely to be constant, and the number of measurements occurring simultaneously will vary over time. I will refer to sensor fusion under these conditions as *asynchronous sensor fusion* or, more simply, as *asynchronous fusion.*

3.1 Kalman Filter

The *Kalman filter* (Bar-Shalom and Fortmann 1988) is a state space algorithm for estimating the track history of a target from all past and current measurements. It uses a *process model* to describe the expected dynamics of the target, particularly between measurements when no other information is available. In tracking applications, the process model is also known as the *dynamic model.*

Assuming linear dynamics, the state of the target at time t_{i+1} may be modelled by

$$\mathbf{x}_{t_{i+1}} = \mathbf{F}_{t_i}\mathbf{x}_{t_i} + \mathbf{w}_{t_i} \tag{3.1}$$

where, at time t_i, $\mathbf{x}_{t_i}$ is the *target state* and $\mathbf{F}_{t_i}$ is the *state transition matrix*[1] describing the target dynamics from time t_i to t_{i+1}. The transition matrix represents a stationary process because it is dependent on the time difference between t_i and t_{i+1}, not on the absolute time. $\mathbf{w}_{t_i}$ is a zero mean uncorrelated Gaussian *process noise* with known covariance $\mathbf{Q}_{t_i}$.[2] This process noise models the deviations of the target from its expected dynamics, e.g.,

[1]The state transition matrix, as it depends only on time difference, should strictly be denoted $\mathbf{F}_{t_{i+1}-t_i}$.

[2]As for $\mathbf{F}_{t_i}$, $\mathbf{Q}_{t_i}$ is dependent only on time difference and should strictly be denoted $\mathbf{Q}_{t_{i+1}-t_i}$.

manoeuvres. The discrete state process noise is sometimes represented by $\mathbf{G}_{t_i}\tilde{\mathbf{w}}_{t_i}$ where $\tilde{\mathbf{w}}_{t_i}$ is a vector whose elements contain the continuous time process noise for each positional coordinate. The discrete time process noise is obtained by integrating the continuous time noise over the time interval t_i to t_{i+1}, a process represented by the matrix $\mathbf{G}_{t_i}$. Therefore the process noise covariance may be expressed as $\mathbf{Q}_{t_i} \equiv \mathbf{G}_{t_i}\tilde{\mathbf{Q}}\mathbf{G}_{t_i}^{\mathsf{T}}$, where $\tilde{\mathbf{Q}} \equiv \mathrm{E}\left[\tilde{\mathbf{w}}_{t_i}\tilde{\mathbf{w}}_{t_i}^{\mathsf{T}}\right]$ and is assumed to be time invariant. The notation $\mathbf{Q}_{t_i}$ has been adopted for clarity. The time sequence, $[t_1, t_2, t_3, \ldots]$ where $t_i \geqslant t_j$ for $i > j$, represents the measurement times. The time t_0 denotes a time, prior to any measurements, at which the target's initial state may be defined.

At each measurement time t_i, a *measurement model* is defined as

$$\mathbf{z}_{t_i} = \mathbf{H}_{t_i}\mathbf{x}_{t_i} + \mathbf{v}_{t_i} \tag{3.2}$$

where $\mathbf{z}_{t_i}$ is the measurement vector, $\mathbf{H}_{t_i}$ is the *measurement matrix* and $\mathbf{v}_{t_i}$ is a zero mean uncorrelated Gaussian *measurement noise* with covariance $\mathbf{R}_{t_i}$. The measurement matrix represents the relationship between the measurement and the target state. The measurement noise includes sensor, target and environmental noise, and any mismatch between the measurement model and the real situation. The measurement noise and process noise are assumed to be independent.

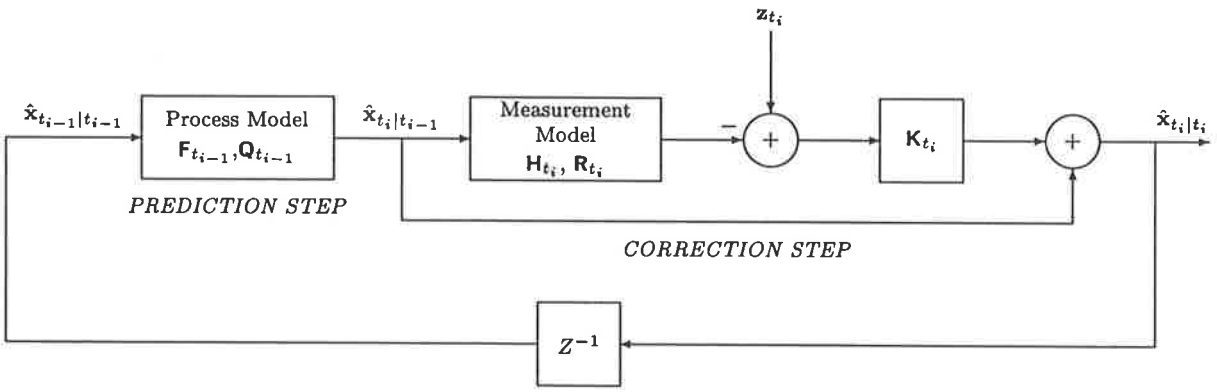


Figure 3.1: Kalman filter functional diagram[3]

The filter estimates the target state at each measurement time using the current and all previous measurements. The state estimate at time t_i, given all measurements up to and including time t_i, is denoted $\hat{\mathbf{x}}_{t_i|t_i}$, and it is obtained using the two step process shown in figure 3.1. The first step is prediction, where the previous state estimate $\hat{\mathbf{x}}_{t_{i-1}|t_{i-1}}$ and the process model are used to predict a new state estimate $\hat{\mathbf{x}}_{t_i|t_{i-1}}$ for the target at time t_i. This new estimate is conditional only on past measurements, i.e., it is the estimate at

[3]Strictly speaking, $\mathbf{Q}_{t_{i-1}}$ is not used to update the predicted state estimates, but to update the predicted state error covariance. This covariance is used to determine the Kalman gain $\mathbf{K}_{t_i}$ that is used to obtain the corrected state estimates.

time t_i given the measurements up to and including time t_{i-1}, and is determined, along with its error covariance $\mathbf{P}_{t_i|t_{i-1}}$, from

$$\begin{aligned} \hat{\mathbf{x}}_{t_i|t_{i-1}} &= \mathbf{F}_{t_{i-1}} \hat{\mathbf{x}}_{t_{i-1}|t_{i-1}} \\ \mathbf{P}_{t_i|t_{i-1}} &= \mathbf{F}_{t_{i-1}} \mathbf{P}_{t_{i-1}|t_{i-1}} \mathbf{F}_{t_{i-1}}^{\mathsf{T}} + \mathbf{Q}_{t_{i-1}} \end{aligned} \tag{3.3}$$

where $^{\mathsf{T}}$ denotes the matrix (and vector) transpose. The second step is correction, where the measurement received at time t_i is used to correct the predicted state estimate $\hat{\mathbf{x}}_{t_i|t_{i-1}}$. The corrected estimate $\hat{\mathbf{x}}_{t_i|t_i}$ and its covariance $\mathbf{P}_{t_i|t_i}$ are obtained from

$$\begin{aligned} \mathbf{K}_{t_i} &= \mathbf{P}_{t_i|t_{i-1}} \mathbf{H}_{t_i}^{\mathsf{T}} \left(\mathbf{H}_{t_i} \mathbf{P}_{t_i|t_{i-1}} \mathbf{H}_{t_i}^{\mathsf{T}} + \mathbf{R}_{t_i} \right)^{-1} \\ \hat{\mathbf{x}}_{t_i|t_i} &= \hat{\mathbf{x}}_{t_i|t_{i-1}} + \mathbf{K}_{t_i} \left(\mathbf{z}_{t_i} - \mathbf{H}_{t_i} \hat{\mathbf{x}}_{t_i|t_{i-1}} \right) \\ \mathbf{P}_{t_i|t_i} &= \left(\mathbf{1} - \mathbf{K}_{t_i} \mathbf{H}_{t_i} \right) \mathbf{P}_{t_i|t_{i-1}} \end{aligned} \tag{3.4}$$

where $\mathbf{K}_{t_i}$ is referred to as the *Kalman gain.* The resulting state estimate error, i.e., the difference between the estimate and the true state, is assumed to be independent of the process noise and the measurement noise. Note that the contribution of all past measurements is contained in the previous state estimate $\hat{\mathbf{x}}_{t_{i-1}|t_{i-1}}$, and is therefore present in the predicted state estimate (3.3).

Many sensors provide measurements at regular time intervals, and in these cases the Kalman filter operates with a fixed measurement rate. This usually results in time invariant dynamic and measurement models for linear systems. However, when the time difference between successive measurements varies, the dynamic and measurement models are no longer time invariant, and the filter is referred to as a *variable update rate Kalman filter.* Jazwinski (1970) gives a detailed description and derivation of both the fixed and variable rate Kalman filters.

3.2 Asynchronous Fused Kalman Filter

The *asynchronous fused Kalman filter* (AFKF) is a Kalman filter that estimates a target's track history using measurements from multiple, and possibly dissimilar, sensors (*see* measurement level fusion, section 2.1.2). A single state space dynamic model is used to model the dynamics of the target of interest and, because the sensors are likely to have different measurement rates, a variable update rate Kalman filter is used to estimate the target states.

Consider a two sensor system in which the measurement rates of each sensor are different. At any measurement time, it is possible to have a single measurement from either of the two sensors, or two simultaneous measurements, one from each sensor. Therefore the measurement model (3.2) at any time will depend on which of the above conditions

occur at that time, i.e.,

$$\mathbf{z}_{t_i} \equiv \begin{cases} \mathbf{z}_{t_i}^{(1)} = \mathbf{H}_{t_i}^{(1)} \mathbf{x}_{t_i} + \mathbf{v}_{t_i}^{(1)} & \text{single measurement from sensor 1} \\ \mathbf{z}_{t_i}^{(2)} = \mathbf{H}_{t_i}^{(2)} \mathbf{x}_{t_i} + \mathbf{v}_{t_i}^{(2)} & \text{single measurement from sensor 2} \\ \begin{bmatrix} \mathbf{z}_{t_i}^{(1)} \\ \mathbf{z}_{t_i}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{t_i}^{(1)} \\ \mathbf{H}_{t_i}^{(2)} \end{bmatrix} \mathbf{x}_{t_i} + \begin{bmatrix} \mathbf{v}_{t_i}^{(1)} \\ \mathbf{v}_{t_i}^{(2)} \end{bmatrix} & \text{simultaneous measurements} \end{cases} \tag{3.5}$$

where, at time t_i, $\mathbf{z}_{t_i}^{(s)}$ denotes the measurement from sensor s, $\mathbf{H}_{t_i}^{(s)}$ the measurement matrix for sensor s and $\mathbf{v}_{t_i}^{(s)}$ the measurement noise from sensor s.

The measurement noise covariance will also depend on the origin and number of measurements, and it is denoted

$$\mathbf{R}_{t_i} \equiv \begin{cases} \mathbf{R}_{t_i}^{(1)} & \text{single measurement from sensor 1} \\ \mathbf{R}_{t_i}^{(2)} & \text{single measurement from sensor 2} \\ \begin{bmatrix} \mathbf{R}_{t_i}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{t_i}^{(2)} \end{bmatrix} & \text{simultaneous measurements} \end{cases} \tag{3.6}$$

where $\mathbf{R}_{t_i}^{(s)}$ denotes the measurement noise covariance for sensor s at time t_i and $\mathbf{0}$ is a zero matrix of the appropriate dimensions. Alternatively, the simultaneous measurements may be processed sequentially using a zero time difference between the two measurements, i.e., the target's dynamic state does not change between simultaneous measurements. This alternative technique is particularly attractive when the occurrence of simultaneous measurements is rare.

This formulation assumes that the measurement noise is uncorrelated between sensors and, in theory, may be extended to any number of sensors.

3.3 Algorithm Evaluation

The AFKF is initially evaluated using simulated data to analyse its tracking behaviour with measurements from two sensors. The AFKF's process and measurement models are matched to the data; the problem of model mismatch is addressed in chapter 4. Finally, the operation of the AFKF is observed using real data collected from a radar and optical tracking system.

When comparing the AFKF to a single sensor Kalman filter, the term *original sensor* is reserved for the sensor that is present in both the AFKF and the single sensor Kalman filter. Additional sensors are those occurring only in the AFKF.

3.3.1 Performance Indicators

The performance of the Kalman filter and AFKF can be quantified by the estimated state error covariance or *track error covariance* (TEC), i.e., the lower the TEC, the better the tracking performance. To compare the performance of the AFKF over the single sensor Kalman filter, the concept of a *fusion gain* is introduced. This value, denoted as FG, is defined as the ratio of TEC of a single sensor tracking filter to the TEC of a multiple sensor tracking filter (in this case the AFKF). Therefore the FG is proportional to the tracking performance of the AFKF relative to the single sensor Kalman filter's performance. Values of FG greater than unity imply better performance from the multi-sensor tracker, values less than unity indicate that the performance of the single sensor tracker is better. In these simulations, the FG is used to compare a dual sensor AFKF to a single sensor Kalman filter.

3.3.2 Simulated Data

The evaluation using simulated data was carried out with artificial measurements of one dimension. Calculated results in this section were taken when the AFKF had reached its steady state operation, i.e., after the filter gains had stabilised to their steady state values. Estimated results where taken as the average over 900 measurement times during steady state operation for 100 different data ensembles.

The simulated data was produced by first generating a random target trajectory using Gaussian process noise to determine the target's manoeuvres. Gaussian measurement noise was then added to the target trajectory to simulate the measurements from each sensor. Figure 3.2 shows the target positions for two example trajectories generated with different values of process noise covariance.

The AFKF was implemented using (3.3) and (3.4), and the multiple sensor measure-

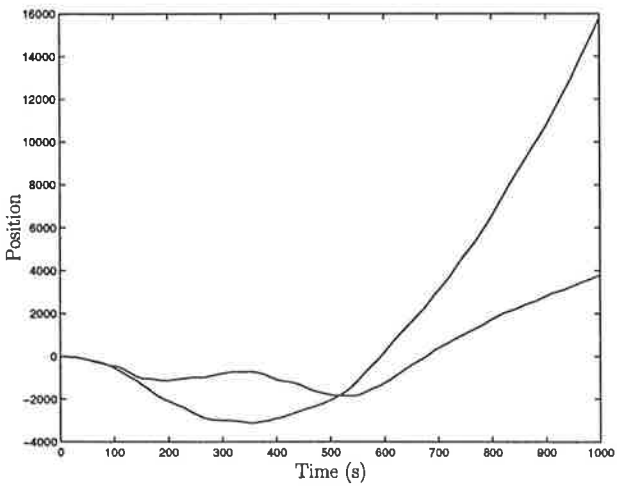


Figure 3.2: Examples of simulated target trajectories

ment model in (3.5) and (3.6). A second order dynamic model, i.e., a constant velocity model, was used, with the transition and process noise covariance matrices given as

$$\begin{aligned} \mathbf{F}_{t_i} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \\ \mathbf{Q}_{t_i} &= \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} q^2 \end{aligned} \tag{3.7}$$

where $T \equiv T(t_{i+1}, t_i) = t_{i+1} - t_i$ is the update interval between times t_i and t_{i+1}. q^2 is the covariance of the continuous time process noise representing random accelerations, i.e., the maximum change in velocity during one update interval is of the order of $\sqrt{q^2 T}$. The measurement matrices for the one dimensional position measurements from each sensor are given as

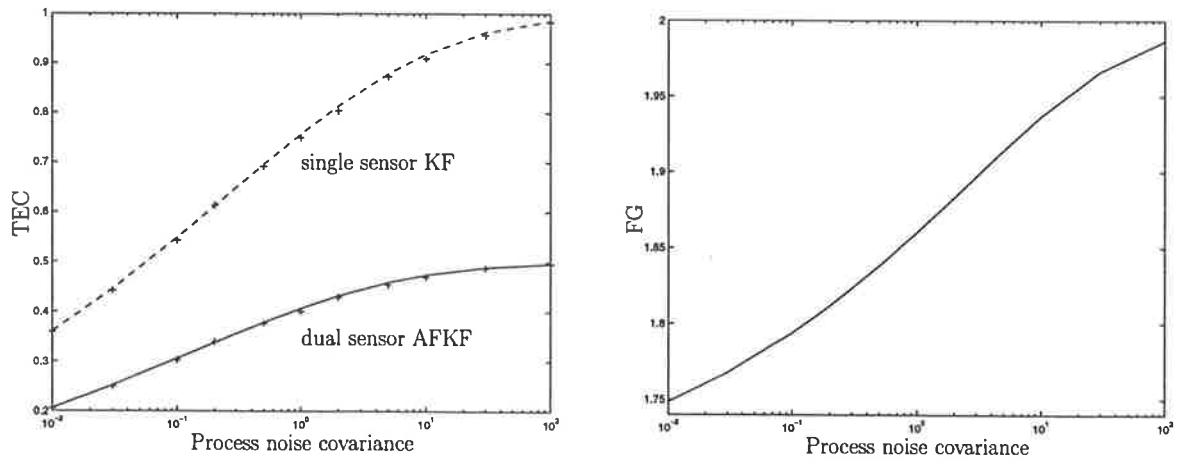
$$\mathbf{H}_{t_i}^{(s)} = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad s = 1, 2. \tag{3.8}$$

Unless stated otherwise, simultaneous measurements from each sensor are assumed.

3.3.2.1 Variations in Process Noise

Simultaneous measurements at one second intervals with unity measurement noise covariance were generated for each sensor. These measurements were processed by the AFKF for values of process noise covariance from 0.01 to 100. The track error covariances derived from (3.3) and (3.4) for the single sensor Kalman filter and the dual sensor AFKF are shown in figure 3.3(a). The estimated values obtained from the simulations are indicated by +'s.

As the process noise covariance increases, the track error covariance approaches unity



(a) Track error covariance

(b) Fusion gain

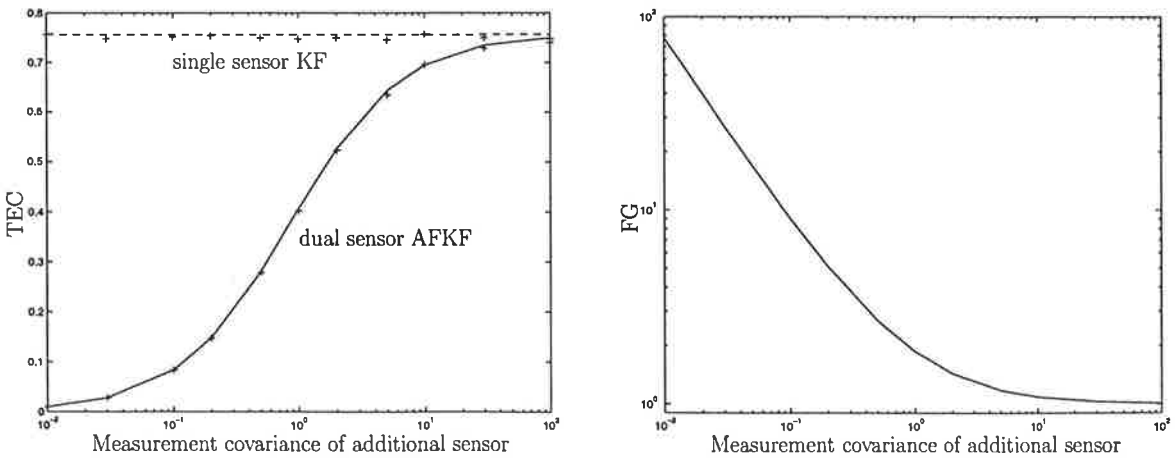
Figure 3.3: AFKF performance over a range of process noise covariances

for the single sensor and 0.5 for the AFKF. These values correspond to the noise covariance of a single sensor measurement and the combined dual sensor measurement respectively. This behaviour is expected, because the filter increases the contribution of the measurements as the uncertainty in the process model increases. As the process noise covariance decreases, the track error covariance also decreases, as does the fusion gain (figure 3.3(b)). This reduction in track error covariance is caused by the process model starting to dominate the filter as its uncertainty decreases relative to the measurement noise. The process model is common to both the single sensor Kalman filter and the AFKF, so the behaviours of both become similar as the influence of the measurements is reduced. Note however that there is still a significant fusion gain of 1.75 when the process noise covariance is 0.01, i.e., one hundredth of the measurement noise covariance.

3.3.2.2 Variations in the Measurement Noise of the Additional Sensor

Using a measurement update interval of one second, and setting the process noise covariance and the measurement noise covariance of the original sensor to unity, the behaviour of the AFKF was evaluated with different values of measurement noise covariance for the additional sensor. The track error covariance (TEC) and FG are shown in figure 3.4.

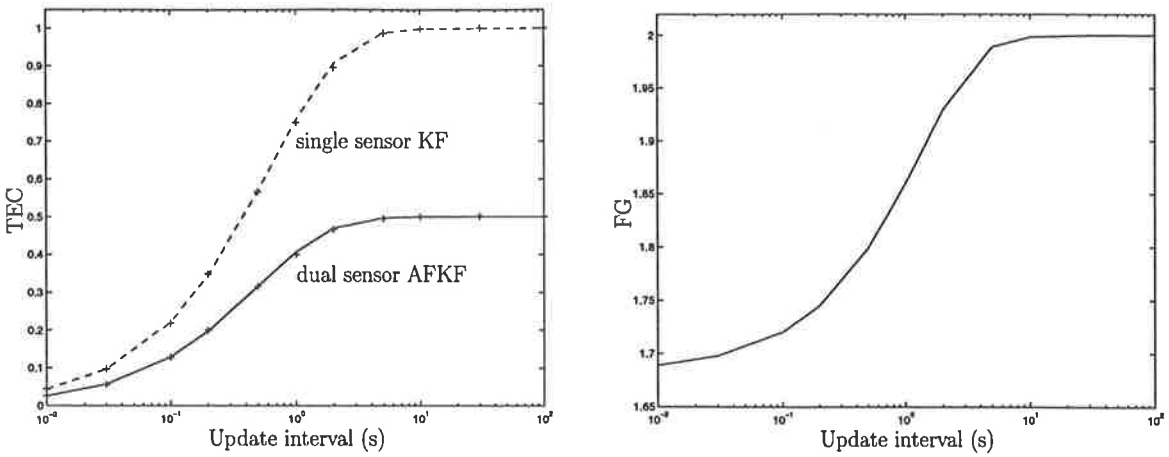
The results show that as the measurement noise covariance of the additional sensor increases, its effect on the AFKF reduces and the operation of the AFKF approximates that of the single sensor Kalman filter, as indicated by the FG approaching unity. However, as its measurement noise covariance decreases, the additional sensor begins to dominate the behaviour of the AFKF, causing it to behave as a single sensor Kalman filter with a much lower measurement noise covariance than that of the actual single sensor Kalman filter. Therefore the AFKF's TEC decreases, causing a corresponding rise in the FG.



(a) Track error covariance

(b) Fusion gain

Figure 3.4: AFKF performance over a range of additional sensor measurement noise covariances



(a) Track error covariance

(b) Fusion gain

Figure 3.5: AFKF performance over a range of update intervals

3.3.2.3 Variations in Measurement Update Rate

These simulations were performed with unity process noise covariance and measurement noise covariances. The performance of the AFKF for different measurement update times, i.e., different time intervals between measurements, is illustrated in figure 3.5.

The results obtained show similar trends to those for variations in process noise. This is because as the time between measurements increases, the uncertainty in the process model also increases. As the effective process noise has increased, the Kalman filter weights the measurements more, resulting in an increase in FG.

3.3.2.4 Offset Measurements

The simulations so far have involved simultaneous measurements from both sensors. If the sensors are able to be synchronised, do simultaneous measurements provide the best tracking performance? Simulations where each measurement from sensor 2 occurs some fraction of the update interval after the corresponding measurement from sensor 1 are considered. The results are shown in figure 3.6, where the offset denotes the fraction of the update interval that separates the measurements from sensor 2 from the corresponding measurements from sensor 1.

Processing each measurement sequentially in time order, the solid line in figure 3.6 represents the TEC immediately after each measurement from sensor 2 is processed. The best tracking performance occurs when the measurements arrive simultaneously, i.e., zero offset. The TEC rises as the offset increases because of the increased uncertainty, due to the increased filter extrapolation time, in the predicted state estimate at the measurement times of sensor 2. The dashed line represents the TEC obtained immediately after each measurement from sensor 1 is processed. This shows a decrease in TEC as the offset increases because, in this case, the time interval between the current measurement from

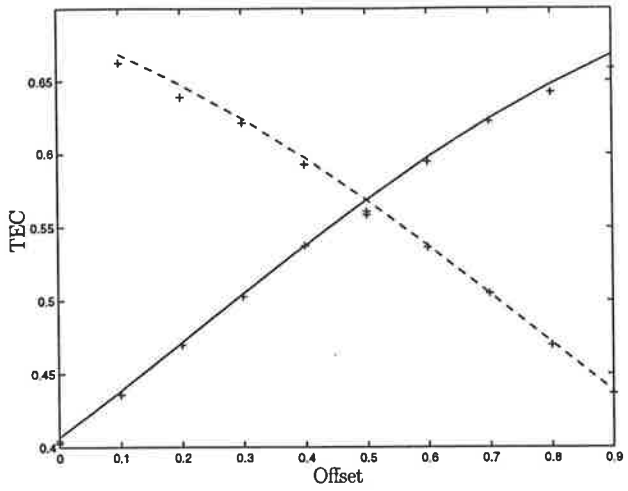


Figure 3.6: Track error covariance over a range of measurement time offsets

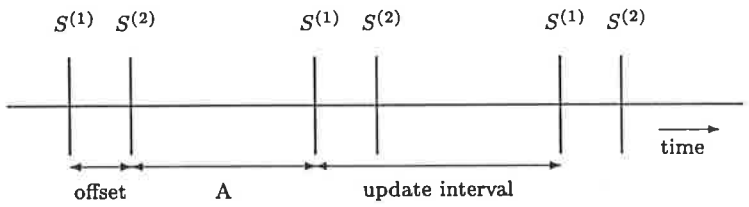


Figure 3.7: Timing of offset measurements from sensor 1 ($S^{(1)}$) and sensor 2 ($S^{(2)}$)

sensor 1 and the previous measurement from sensor 2 (denoted 'A' in figure 3.7) decreases as the offset increases.

As can be seen from figure 3.6, the combined plots are symmetrical about an offset of 0.5 update intervals, because an offset of say 0.3 between one measurement and the next corresponds to an offset of 0.7 between that same measurement and the previous. Therefore, if the state estimates may be taken at the measurement times of either sensor, the worst case performance occurs when the measurement separation is maximum, i.e., half the update interval.

As the ratio of process noise covariance to measurement noise covariance decreases, the measurements have less effect on the tracking performance, and the effect of the measurement offset on the tracking performance becomes less pronounced.

3.3.3 Real Data

The AFKF was evaluated using real radar and optical data (Krieg and Gray 1996c). The radar and optical sensors were not synchronised and had different measurement update rates, i.e., approximately 30 Hz and 50 Hz respectively. The radar measures target position (in radar coordinates, i.e., azimuth, elevation and range) and Doppler velocity,[4] and the

[4]Doppler velocity is the negative of range rate, i.e., the velocity towards the sensor.

optical system measures azimuth and elevation. Therefore the dynamics are non-linear with complex dependencies between the positional coordinates and their derivatives. The targets used in this evaluation were relatively benign, exhibiting only gentle manoeuvres. Therefore a second order linear dynamic model with additional process noise to account for the manoeuvres and non-linearities was implemented. The resulting state vector included each positional coordinate and its first derivative, i.e.,

$$\begin{bmatrix} \eta & \dot{\eta} & \varepsilon & \dot{\varepsilon} & R & \dot{R} \end{bmatrix}^{\mathsf{T}} \tag{3.9}$$

where η represents the azimuth or bearing, ε the elevation and R the range, with $\dot{a}$ indicating the derivative of a. The corresponding transition and process noise covariance matrices were written

$$\begin{aligned} \mathbf{F} &= \begin{bmatrix} \mathbf{F}^{(1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}^{(1)} \end{bmatrix} & \mathbf{F}^{(1)} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \\ \mathbf{Q} &= \begin{bmatrix} \mathbf{Q}^{(1)} q_\eta^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{(1)} q_\varepsilon^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}^{(1)} q_R^2 \end{bmatrix} & \mathbf{Q}^{(1)} &= \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \end{aligned} \tag{3.10}$$

where q_η^2, q_ε^2 and q_R^2 denote the continuous time process noise covariances for azimuth, elevation and range respectively. As the targets are tracked in radar coordinates, the actual target dynamics are non-linear and the coordinates (η, ε and R) are not independent. For simplicity, linear dynamics are assumed, and the dependencies between the coordinates are ignored in the state transition matrix $\mathbf{F}$. The errors introduced by these assumptions are modelled by additional process noise. This, in effect, decouples the coordinates, and therefore each may be filtered (tracked) independently of the others.

For all i, let the measurement vectors $\mathbf{z}_{t_i}^{(1)} = \begin{bmatrix} \eta_{t_i}^{(1)} & \varepsilon_{t_i}^{(1)} & R_{t_i}^{(1)} & -\dot{R}_{t_i}^{(1)} \end{bmatrix}^{\mathsf{T}}$ and $\mathbf{z}_{t_i}^{(2)} = \begin{bmatrix} \eta_{t_i}^{(2)} & \varepsilon_{t_i}^{(2)} \end{bmatrix}^{\mathsf{T}}$ represent the measurements from the radar and optical sensor respectively. Then the measurement and covariance matrices for the AFKF (*see* (3.5) and (3.6)) are given as

$$\begin{aligned} \mathbf{H}_{t_i}^{(1)} \equiv \mathbf{H}^{(1)} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \\ \mathbf{R}_{t_i}^{(1)} \equiv \mathbf{R}^{(1)} &= \begin{bmatrix} \sigma_\eta^{(1)\,2} & 0 & 0 & 0 \\ 0 & \sigma_\varepsilon^{(1)\,2} & 0 & 0 \\ 0 & 0 & \sigma_R^{(1)\,2} & 0 \\ 0 & 0 & 0 & \sigma_{\dot{R}}^{(1)\,2} \end{bmatrix} \end{aligned} \tag{3.11}$$

$$\mathbf{H}_{t_i}^{(2)} \equiv \mathbf{H}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathbf{R}_{t_i}^{(2)} \equiv \mathbf{R}^{(2)} = \begin{bmatrix} \sigma_\eta^{(2)\,2} & 0 \\ 0 & \sigma_\varepsilon^{(2)\,2} \end{bmatrix} \tag{3.12}$$

where $\sigma_\eta^{(s)\,2}$, $\sigma_\varepsilon^{(s)\,2}$, $\sigma_R^{(s)\,2}$ and $\sigma_{\dot{R}}^{(s)\,2}$ denote the measurement noise covariance of the azimuth, elevation, range and Doppler velocity measurements from sensor s.

The single sensor Kalman filter using only radar measurements, i.e., the radar Kalman filter, and the single sensor Kalman filter using only optical measurements, i.e., the optical Kalman filter, use the same dynamic model as the AFKF and the appropriate measurement and noise covariance matrices from (3.11) and (3.12). This section concentrates on the angular measurements, as it is these that are common to both sensors.

3.3.3.1 Data Collection

The radar and optical data was collected using a sensor fusion testbed (appendix A) located at an elevated site approximately 3 km from, and 140 metres above, a light aircraft airfield situated in the northern suburbs of Adelaide, South Australia. This site also provided a clear view of the airspace above the Adelaide International Airport, some 25 km away.

The collected data was arranged into data sets containing the radar and optical measurements collected while attempting to track a single target with the radar. Many of these data sets also include measurements from other aircraft, and radar and optical clutter.

The measurement noise covariances for each sensor were estimated from the data, and the values $\sigma_\eta^{(1)\,2} = \sigma_\varepsilon^{(1)\,2} = 1\,\text{mrad}^2$, $\sigma_R^{(1)\,2} = 40\,\text{m}^2$, $\sigma_{\dot{R}}^{(1)\,2} = 4\,\text{m}^2\text{s}^{-2}$ and $\sigma_\eta^{(2)\,2} = \sigma_\varepsilon^{(2)\,2} = 0.1\,\text{mrad}^2$ were substituted into (3.11) and (3.12).

The true target position is required to determine the tracking errors. As ground truth was not available, the target position was estimated by applying polynomial fitting techniques to the data. This smoothed the measurements and removed short time constant variations, such as those produced by air turbulence. This had the effect of artificially increasing the measured track error covariance by an unknown amount.

During data collection, the sensor mount was controlled solely by the radar. This provided the opportunity for the optical system to re-acquire lost tracks provided the radar maintained track. If the radar lost track, the optical sensor eventually lost the target also.

3.3.3.2 Clear Sky Tracking

Under *clear sky* conditions, i.e., in the absence of significant clutter and other targets, the AFKF and the radar and optical Kalman filters all successfully maintained track on the

Tracker	Azimuth	Elevation
Radar	0.083	0.18
Optical	0.037	0.083
Fused	0.037	0.072

Table 3.1: Track error covariance ($mrad^2$) of a light aircraft

Tracker	Azimuth	Elevation
Radar	0.082	0.050
Optical	0.053	0.019
Fused	0.050	0.017

Table 3.2: Track error covariance ($mrad^2$) of a commercial aircraft

target of interest. Typical track error covariances for light and commercial aircraft are shown in tables 3.1 and 3.2.

It is noted that these results are approximate because, as it is estimated from the data, the actual target position is not known with any certainty. Generally this will increase the estimated track error and give lower values of fusion gain. The assumed measurement noise covariances that were estimated from the data are subject to change with target type, target position and environmental factors. This may result in a mismatch between the data and filter models, also causing an increase in track error. Therefore it is likely that the values of FG obtained from these tables will be lower than those actually being achieved. However these results still show an increase in performance of the AFKF over the radar Kalman filter, and equal or better performance over the optical Kalman filter. It appears that, for these examples, little or no benefit is gained from using the radar's angular measurements.

The Kalman, or filter, gains provide a useful insight into the operation of the tracking filters. They are the weights that are applied to the measurement innovations, i.e., the errors between the measurements and the predicted target state, to correct the state estimates. Figure 3.8 shows the steady state gains that are applied to the azimuth measurement innovations to correct the azimuth position estimates. The gain for the radar Kalman filter is labelled 'A', the optical Kalman filter gain is shown as 'B', and the AFKF gains for the optical and radar measurements are denoted 'C' and 'D' respectively.

The radar Kalman filter gain 'A' contains a number of peaks, following missed detections, where the higher gain is used to increase the influence of the measurements when they do arrive. When detections are missed, the filter has to propagate the target dynamics for a longer period of time, increasing the uncertainty in the state estimates. Therefore a larger correction is required from the measurements. Conversely, the optical Kalman filter gain 'B' is lower than that of the radar Kalman filter because the measurement update interval of the optical sensor is less than that of the radar's. The shorter the measurement update interval (i.e., the higher the update rate), the lower the predicted state estimate covariance and the lower the gain.

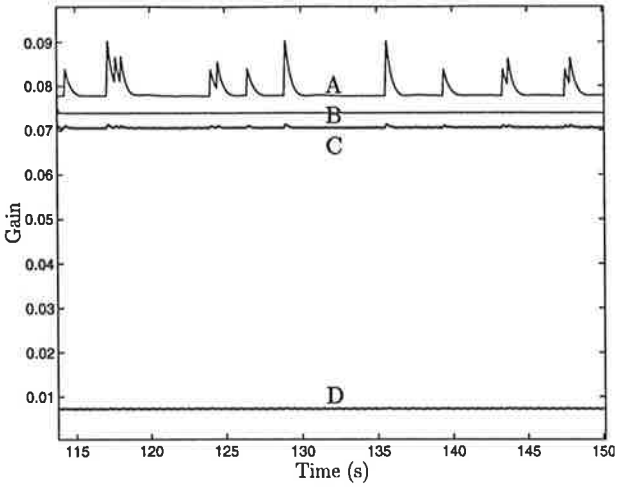


Figure 3.8: Kalman gains for single and fused sensor trackers

The ratio of the optical and radar gains for the AFKF, 'C' and 'D' respectively, is determined by the inverse of the sensor measurement noise covariances and the relative measurement update rates. In this case, the ratio is determined predominantly by the covariances, and the higher optical gain ensures that the lower noise optical measurements have a greater influence on the state estimation. The overall lower gains of the AFKF are caused by the increase in the average measurement update rate resulting from the increased number of measurements. The AFKF gains are also influenced by the missing radar measurements, but to a much lesser extent than for the radar Kalman filter gain because of the presence of the optical measurements. This variation in the AFKF radar gain is not obvious from figure 3.8 because of the small average value of this gain; the percentage variation of the AFKF radar gain is similar to that of the AFKF optical gain.

3.3.3.3 Tracking Targets at Close Range

When a target occupies a significant region in a sensor's field of view, the sensor may not be able to maintain a consistent reference point on the target. The position of a radar measurement is determined by the geometry of the significant electro-magnetic reflectors on the target, and this generally changes with target aspect. The optical tracker relies upon image processing to locate the target's centre of mass or centroid, and this may also change with target aspect. This problem increases with target size and decreases with target range.

The optical sensor on the testbed had difficulty maintaining a consistent centroid position in such scenarios, the effect of which is shown in table 3.3. The track error covariance of the optical Kalman filter is greater than that of the radar Kalman filter. The additional *noise*, caused by the uncertainty in target position, was not included in the measurement model, therefore the optical Kalman filter assumes that the optical noise is less than its actual value and weights the optical measurements higher than it should. This effectively transfers additional noise to the target track and reduces tracking performance.

Tracker	Azimuth	Elevation
Radar	0.10	0.021
Optical	0.34	0.044
Fused	0.31	0.040

Table 3.3: Track error covariance ($mrad^2$) of a light aircraft at close range

The AFKF treats the optical measurements in the same way, and its performance is degraded to such an extent that the radar Kalman filter out-performs it.

3.3.3.4 Loss of Measurements From One Sensor

Earlier clear sky results indicated that the radar's angular measurements added little to tracking performance in terms of track error covariance. However, if the optical sensor loses track, the AFKF will continue to track using the angular measurements from the radar, albeit with increased TEC. An example of this is illustrated in figure 3.9 where, at 'A', the target moves out of optical range and the optical measurements cease.

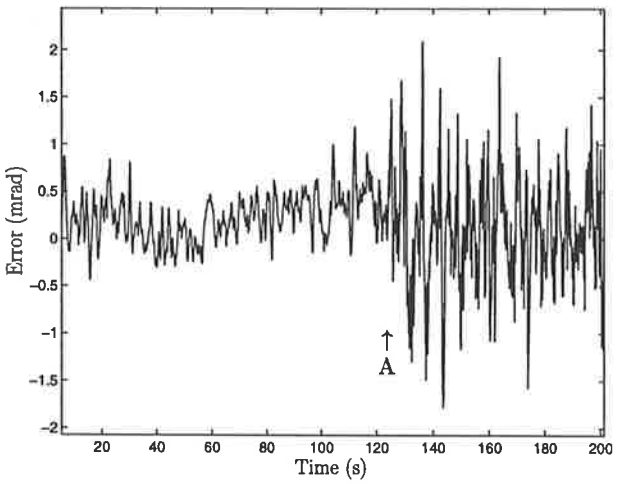


Figure 3.9: Azimuth tracking error increase due to loss of optical measurements

3.3.3.5 Tracking in Clutter

The elevated site provided a good opportunity to evaluate tracking performance in clutter, as the targets were often tracked at negative elevations against an urban background. The optical tracker had great difficulty maintaining track under these conditions, frequently acquiring objects in the background. The radar suffered a similar fate in the presence of large reflectors, such as industrial buildings.

Figure 3.10 shows the azimuth tracking errors obtained from a target that was tracked in clutter. The dotted line represents the tracking errors obtained from the optical Kalman filter, the dashed line the radar Kalman filter, and the solid line the AFKF.

Initially the clutter level was insufficient to influence the trackers. However, radar clutter introduced at 'A' severely degraded the performance of the radar Kalman filter,

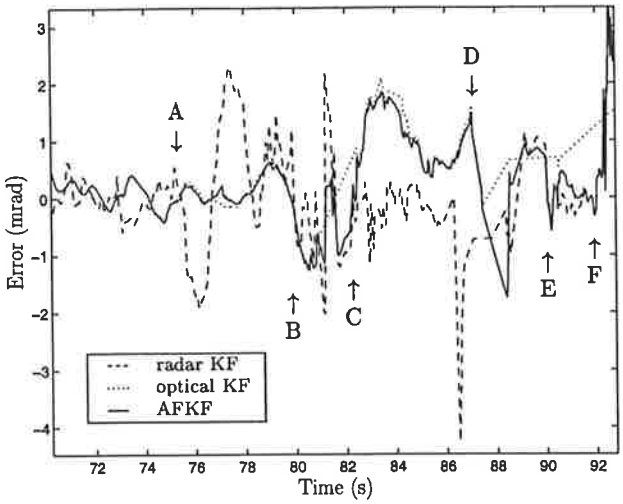


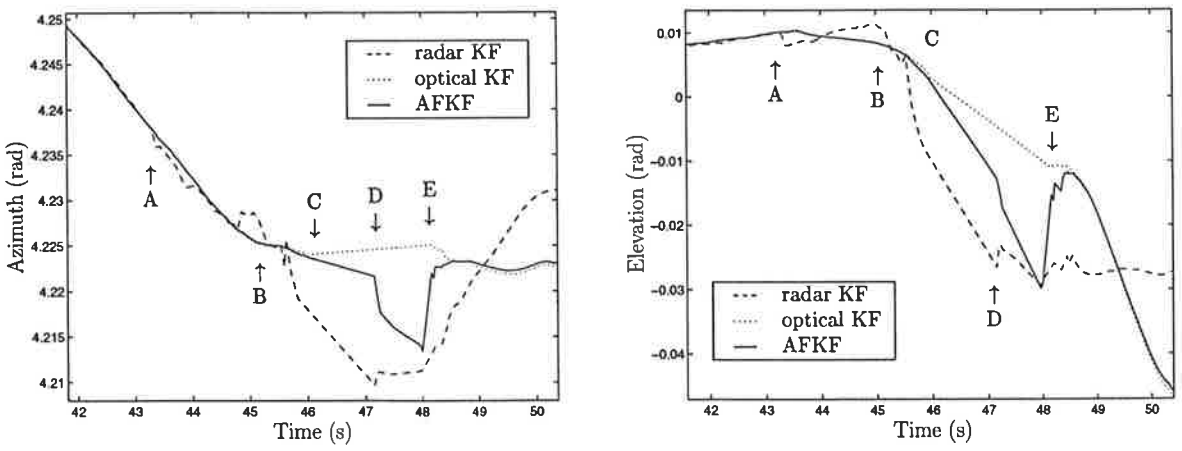
Figure 3.10: Azimuth tracking errors in clutter

causing it to lose track. The AFKF was not significantly affected because the optical measurements dominate its operation. At 'B', the optical sensor lost the target due to clutter, and then the optical measurements ceased. The AFKF was then driven by the radar clutter, until the optical sensor detected another object in the clutter at 'C'. The AFKF followed this object until it was lost at 'D'. After a few spurious measurements, the radar acquired another object in clutter at 'E', and the AFKF tracked it until it was lost a short time later at 'F'.

This example illustrates the importance of determining which measurements to use for tracking a particular target. The AFKF and Kalman filter do not perform data association, and therefore they perform very poorly in clutter.

This problem is further illustrated by a second example, for which the azimuth and elevation tracks are shown in figure 3.11.

Radar clutter appeared at 'A' and caused the radar Kalman filter to deviate from the



(a) Azimuth

(b) Elevation

Figure 3.11: Tracking in clutter

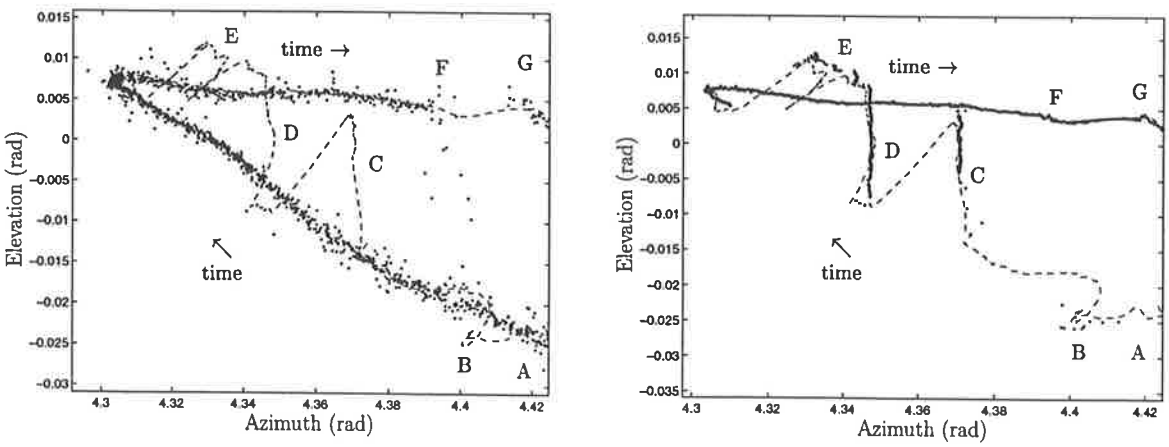
true target. The AFKF maintained track using the optical measurements until, at 'B', the radar was completely seduced by clutter. The radar subsequently moved the mount and the sensors away from the target until all measurements ceased at 'C'. The propagated state estimates were influenced by spurious radar measurements at 'D', and the target fortuitously re-entered the field of view at 'E', was re-acquired by the optical sensor and tracked by both the AFKF and optical Kalman filter.

It was more common for the optical sensor to be effected by clutter because it is only two-dimensional. Having no range or depth discrimination, it can receive optical measurements from any object within its field of view and over its entire detection range. Therefore the AFKF performed poorly in optical clutter because the optical measurements have the greatest influence on its operation. When optical measurements were available, radar clutter had little effect on the angular tracking performance of the AFKF.

3.3.3.6 Interfering Targets

Other targets in the optical sensor's field of view often seduced the optical sensor in the same way as optical clutter. This is illustrated by the example elevation tracks shown in figure 3.12 where, in each plot, the measurements are shown as dots and the AFKF track as a dashed line. The radar measurements are shown in figure 3.12(a) and the optical measurements are shown in figure 3.12(b).

The target track commenced at 'A' and proceeded in a clockwise direction. The AFKF was seduced from the target by optical clutter at 'B' and by other optical targets at 'C' and 'D' (until they were lost as they left the optical field of view). After losing the target near the top of the field of view at 'D', the optical tracker was seduced by several optical interferences ('E') before it re-acquired the target. Normal dual sensor tracking continued until the radar lost the target ('F'), at which point the AFKF continued to track the



(a) With radar measurements

(b) With optical measurements

Figure 3.12: Azimuth-elevation track from the AFKF with interfering targets

target using only the optical measurements until dual sensor tracking recommenced at 'G', with the re-introduction of radar measurements.

In real time operation, track loss will occur once the optical tracker has been seduced, because the optical measurements from the interfering source will move the AFKF track, and therefore the mount, away from the original target.

Again the problems associated with the absence of data association are highlighted, with both clutter and other targets likely to degrade tracking performance.

3.3.3.7 Sensor Registration Errors

The problems associated with sensor registration errors are illustrated by introducing an artificial 2 milli-radian azimuth offset between the radar and optical data in one of the collected data sets. The resulting tracks are shown in figure 3.13, where the dotted track represents the optical Kalman filter, the dashed the radar Kalman filter, and the solid the AFKF. The AFKF track falls between the two sets of measurements, its actual position determined by the relationship between the noise covariance of the two sensors. The AFKF track will be closer to the sensor with the lowest noise covariance; therefore the error is relatively small if the optical sensor is correctly aligned. However, if the radar is the correctly aligned sensor, the error is much larger. Therefore it is more important to align the higher resolution sensors accurately or to include the registration errors in the AFKF.

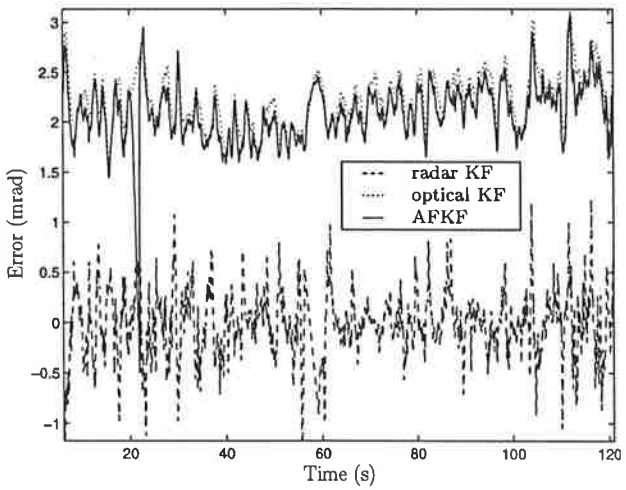


Figure 3.13: Azimuth tracking error with sensor misalignment

3.3.3.8 Summary of Real Data Evaluation of AFKF

The performance of the AFKF has been evaluated using real radar and optical data. Under near ideal conditions, the AFKF performed as expected, providing state estimates with lower covariance (and therefore higher confidence) than those obtained from either

the radar or optical Kalman filters. This improvement was only marginal over the optical Kalman filter because the optical sensor has a significantly higher resolution than the radar.

At close range, the optical sensor had difficulty maintaining a consistent centroid position on the target's body. The resulting *jitter* added extra noise to the optical measurements that was not accounted for in its measurement model. The resulting tracking performance from both the optical Kalman filter and the AFKF under these conditions was inferior to that of the radar Kalman filter.

The optical sensor has no range measurement, therefore it cannot discriminate between targets that are separated only in range. As a result, it was often seduced by clutter and other targets and, because the Kalman filter and the AFKF do not perform any data association, the trackers were seduced from the target of interest by the high resolution optical measurements from these other sources.

The angular measurements from the radar contributed little to the AFKF during normal operation. However, if the optical sensor failed, the AFKF continued tracking using only the measurements from the radar. Therefore lower resolution measurements may make a significant contribution to system performance in the event of sensor failure.

The higher the resolution of a particular sensor, the more critical it becomes to correctly align that sensor. In practice, the highest resolution sensor should provide the datum to which the other sensors are aligned.

CHAPTER 4

Sensitivity to Model Errors in Fused Trackers

A number of assumptions are prerequisites for the successful tracking of a target's dynamic behaviour. These assumptions determine how the expected dynamics of a target and the characteristics of the sensors providing the measurements are represented, and this information is generally expressed as dynamic and measurement models respectively. Of particular interest to the tracking filter designer is the effect of discrepancies, or *mismatch*, between the assumed models and the physical phenomena they represent, particularly as physical characteristics change with time and the environment. The study of this robustness to model errors is referred to as *sensitivity analysis*, and it has been covered extensively for single sensor systems, in particular the Kalman filter (Gelb 1992).

The sensitivity of a dual sensor *fused Kalman filter* to model errors is compared to that of a *single sensor Kalman filter* that uses one of the two sensors used by the fused Kalman filter.

4.1 Sources of Model Mismatch

Model mismatch may occur in either the dynamic (process) model or the measurement model (Gelb 1992). In the process model of a linear Gaussian Kalman filter (3.1), the likely candidates of model mismatch are the state transition matrix ($\mathbf{F}$) and the additive process noise. Model errors in the state transition matrix arise from non-linearities and higher order dynamics. These linearities may be compensated by adding additional process noise. The physical processes that the additive Gaussian process noise represent are generally not Gaussian distributed. Also the processes may not be uncorrelated over time, introducing bias into the noise. Although both these contravene the assumptions of the linear Gaussian Kalman filter, their effects will not be considered here. Assuming zero mean Gaussian process noise, the only parameter left in the process noise model is its covariance ($\mathbf{Q}$). This is a critical design parameter of the Kalman filter because it

directly affects the gain of the filter. If the filter's assumed or design value is higher than that of the data, the filtered state estimates will contain excessive noise. A value that is too low will ultimately cause the filter to diverge from the true track.

In the measurement model (3.2), mismatch may appear in the measurement matrix ($\mathbf{H}$) or the additive measurement noise. Errors in the measurement matrix are generally caused by non-linearities in the transfer function between the target state and the measurement vector. These may be approximated by additional measurement noise and are not considered here. As for the process noise, the measurement noise is assumed to be zero mean and Gaussian. Although this may not accurately represent the physical behaviour, these effects are beyond the scope of this analysis. The measurement noise covariance ($\mathbf{R}$) is another critical design parameter; it too determines the filter gain. An excessive design value of measurement noise covariance will reduce the Kalman gain and increase the possibility of track divergence, while a smaller value will increase the gain and introduce additional sensor noise into the state estimate.

The sensitivity analysis presented here will centre on errors in the process noise covariance and the measurement noise covariance.

4.2 Performance measures

The *track error covariance* (TEC)[1] provides a quantitative measurement of tracking performance. Adding another sensor under matched conditions improves the track error covariance and, to produce meaningful results, this effect must be quantified. The *fusion gain*, introduced in section 3.3.1, provides a quantitative measure of this tracking performance improvement and, in the context of this chapter, will be referred to as the *assumed FG,* i.e., the value obtained under the assumption of matched conditions.

A similar parameter can be used to specify the performance improvement of the fused Kalman filter over the single sensor filter under conditions of model mismatch. This new parameter, the *actual FG,* is defined as the ratio of single sensor Kalman filter track error covariance to the fused Kalman filter track error covariance under identical mismatch conditions for the process noise covariance and the measurement noise covariance of the sensor common to both filters.

Comparing the assumed FG with the actual FG provides an indication of the component of performance improvement (or degradation) contributed by the model errors. The *mismatch factor* (MF) is introduced as the ratio of actual FG to assumed FG. The MF provides a measure of the performance improvement (or degradation) of the fused Kalman filter caused by model mismatch relative to that of the single sensor Kalman filter. As the minimum track error covariance occurs under matched conditions, any increase in actual

[1] Under zero average track error conditions, the track error covariance is defined as the average squared track error.

FG is caused by a smaller rise in the track error covariance of the fused system than for the single sensor system. Therefore the fused system is affected less by the mismatch and is thus more robust. If the relative increases in the single and fused systems' track error covariance are the same, the actual FG maintains its assumed value. Therefore, under these conditions, where the sensitivity of both systems is the same, the MF is unity. An increase in actual FG relative to the assumed FG causes a rise in the MF, where values of MF above unity indicate that the fused system is less sensitive, and values of MF below unity indicate a greater sensitivity in the fused system.

4.3 Filter Models

The Kalman filter described in section 3.1 was used for both the single sensor and fused Kalman filters. Each used a second order (constant velocity) target state model to represent the target's dynamic behaviour. The state transition matrix and process noise covariance matrix are given in (3.7). For a fixed measurement update rate, the only degree of freedom in the process noise covariance matrix is the continuous time process noise covariance q^2 (section 3.3.2). It is this parameter that is used to vary the process noise covariance. Both Kalman filters operate with a uniform update rate, therefore the time dependence in the notation has been dropped for convenience.

The one dimensional measurement from each sensor is modelled using the measurement matrix in (3.8) with a scalar measurement noise covariance $\sigma^{(s)\,2}$. The measurement matrix and measurement noise covariance for the single sensor Kalman filter is given as

$$\mathbf{H} \equiv \mathbf{H}^{(1)} \qquad\qquad \mathbf{R} \equiv \sigma^{(1)\,2}, \tag{4.1}$$

and for the fused Kalman filter with simultaneous measurements,

$$\mathbf{H} \equiv \begin{bmatrix} \mathbf{H}^{(1)} \\ \mathbf{H}^{(2)} \end{bmatrix} \qquad\qquad \mathbf{R} \equiv \begin{bmatrix} \sigma^{(1)\,2} & 0 \\ 0 & \sigma^{(2)\,2} \end{bmatrix}. \tag{4.2}$$

Sensor 1 is common to both the single sensor and fused Kalman filters, and as such is labelled as the *original* sensor. Sensor 2 has been added to the single sensor Kalman filter to form the fused Kalman filter, and it is referred to as the *additional* sensor.

4.4 Theoretical Track Error Covariance

The theoretical steady state track error covariance of a Kalman filter may be determined by solving the steady state Riccati equation, i.e.,

$$\mathbf{P}_{ss} = (\mathbf{1} - \mathbf{K}_{ss}\mathbf{H})\left(\mathbf{F}\mathbf{P}_{ss}\mathbf{F}^{\mathsf{T}} + \mathbf{Q}\right)(\mathbf{1} - \mathbf{K}_{ss}\mathbf{H})^{\mathsf{T}} + \mathbf{K}_{ss}\mathbf{R}\mathbf{K}_{ss}^{\mathsf{T}} \tag{4.3}$$

where $\mathbf{P}_{ss}$ denotes the *actual* steady state track error covariance, $\mathbf{K}_{ss}$ the steady state Kalman gain and $\mathbf{Q}$ and $\mathbf{R}$ the actual process noise covariance and actual measurement noise covariance of the data. The equation in this form does not rely upon optimal operation of the Kalman filter, and it is therefore suitable to use under the non-optimal conditions of model mismatch.

The Riccati equation cannot be solved explicitly and therefore must be solved iteratively. Let $\mathbf{Q}_D$ and $\mathbf{R}_D$ denote the filter design or *assumed* values of the process noise covariance and measurement noise covariance respectively, with assumed parameters q_D^2, and $\sigma_D^{(1)2}$ and $\sigma_D^{(2)2}$. Design (assumed) parameters and variables calculated from the assumed design parameters, i.e., data independent parameters and variables, are identified by the subscript $_D$, and the superscript $^{(i)}$ denotes the value of that variable at the i^{th} iteration.

Denoting $\mathbf{M}_D^{(i)}$ as the assumed predicted state error covariance and $\mathbf{P}_D^{(i)}$ as the assumed track error covariance at the i^{th} iteration, and commencing with some large value of initial state error covariance $\mathbf{P}^{(0)} = \mathbf{P}_D^{(0)}$, the assumed and actual track error covariances are calculated using

$$\begin{aligned}
\mathbf{M}_D^{(i)} &= \mathbf{F}\mathbf{P}_D^{(i-1)}\mathbf{F}^{\mathsf{T}} + \mathbf{Q}_D \\
\mathbf{K}_D^{(i)} &= \mathbf{M}_D^{(i)}\mathbf{H}^{\mathsf{T}}\left(\mathbf{H}\mathbf{M}_D^{(i)}\mathbf{H}^{\mathsf{T}} + \mathbf{R}_D\right)^{-1} \\
\mathbf{P}_D^{(i)} &= \left(\mathbf{1} - \mathbf{K}_D^{(i)}\mathbf{H}\right)\mathbf{M}_D^{(i)} \qquad (4.4)
\end{aligned}$$

and

$$\mathbf{P}^{(i)} = \left(\mathbf{1} - \mathbf{K}_D^{(i)}\mathbf{H}\right)\left(\mathbf{F}\mathbf{P}^{(i-1)}\mathbf{F}^{\mathsf{T}} + \mathbf{Q}\right)\left(\mathbf{1} - \mathbf{K}_D^{(i)}\mathbf{H}\right)^{\mathsf{T}} + \mathbf{K}_D^{(i)}\mathbf{R}\mathbf{K}_D^{(i)\mathsf{T}}.$$

The equations in (4.4) are repeated until $\mathbf{P}^{(i)}$ and $\mathbf{P}_D^{(i)}$ both converge, i.e., $\mathbf{P}_{ss} \triangleq \mathbf{P}^{(i)} \approx \mathbf{P}^{(i-1)}$ and $\mathbf{P}_{Dss} \triangleq \mathbf{P}_D^{(i)} \approx \mathbf{P}_D^{(i-1)}$ (Gelb 1992). Note that the first three equations are simply the covariance update equations of the Kalman filter in (3.3) and (3.4), and the fourth is the steady-state Riccati equation of (4.3).

4.5 Methodology

A selection of single sensor and fused Kalman filters with various design values of process noise covariance and measurement noise covariance were applied to two hundred sets of statistically equivalent simulated data. The average steady state covariance of the *track error*[2] was estimated for each filter from these simulations.

This actual *track error covariance* $\mathbf{P}_{ss}$, calculated using (4.4), depends on the actual values of covariance for the data, i.e., it takes into account any mismatch in the filter models and the data. The Kalman filter calculates the track error covariance assuming

[2]Difference between the estimated target state and the target's true state.

that its models are matched to the data, and this assumed track error covariance $\mathbf{P}_{Dss}$ is also obtained from (4.4). If the filter is matched to the data, $\mathbf{P}_{ss}$ and $\mathbf{P}_{Dss}$ are identical.

The effect of process model mismatch was evaluated by varying the design value of the filter process noise covariance while matching the measurement noise covariance to the data. Similarly, the effect of measurement noise mismatch was evaluated by varying the filter design value of the measurement noise of each sensor while maintaining matched process noise.

Variations in covariances over three orders of magnitude either side of the matched values, i.e., 0.001–1000 times the matched value, were considered. In the process noise mismatch results, the process noise covariances and measurement noise covariances are all normalised to the process noise covariance of the data. For the measurement noise mismatch, these same variables are normalised to the original sensor's measurement noise covariance. Provided all measurement and process noise covariances are scaled by the same value, the fusion gains and mismatch factors are not affected because they are ratios of the TEC's. The absolute values of track error covariance are not important, as only the trends and relationships between such parameters are of interest. Therefore, for convenience, these are scaled by the single sensor track error covariance corresponding to matched conditions. The results obtained from the simulations were in close agreement with the calculated values, therefore only the calculated results are presented.

The ratios of assumed covariance to actual covariance are referred to as *mismatch ratios* for both process and measurement noise mismatch. Increases in mismatch, as opposed to increases in mismatch ratio, refer to movements of the mismatch ratio away from unity, i.e., matched conditions, in either direction.

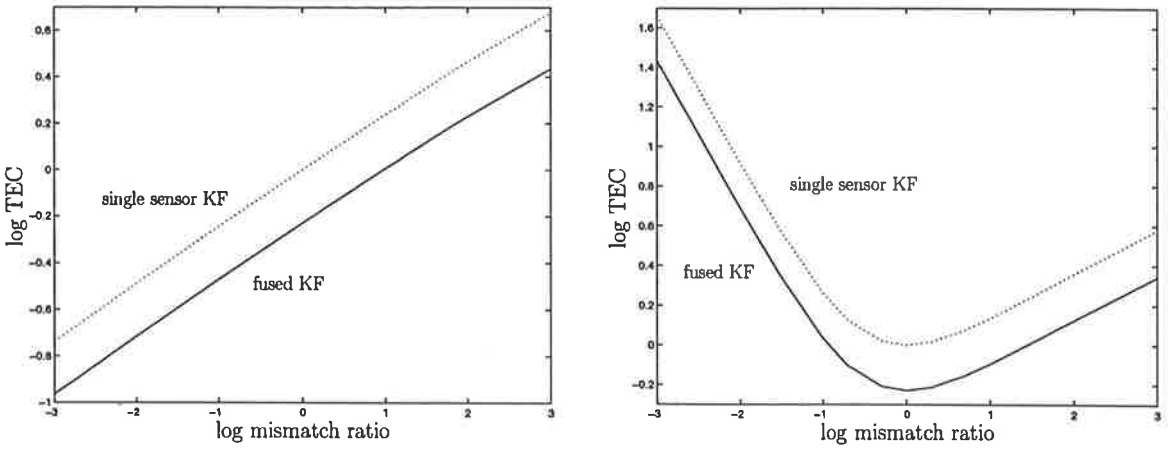
4.6 Results

This section contains the results obtained for mismatch in both process noise covariance and measurement noise covariance.

The horizontal axes in the plots that follow, generally labelled *log mismatch ratio*, represent the logarithm (base 10) of the ratio of assumed covariance to the actual covariance, e.g., $\log_{10} \frac{q_D^2}{q^2}$ and $\log_{10} \frac{\sigma_D^{(1)\,2}}{\sigma^{(1)\,2}}$. The use of logarithmic scales not only provides a clear presentation of the results over several orders of magnitude, but the differences between these logarithmic plots represent the logarithms of the ratio parameters, e.g., the FG and MF.

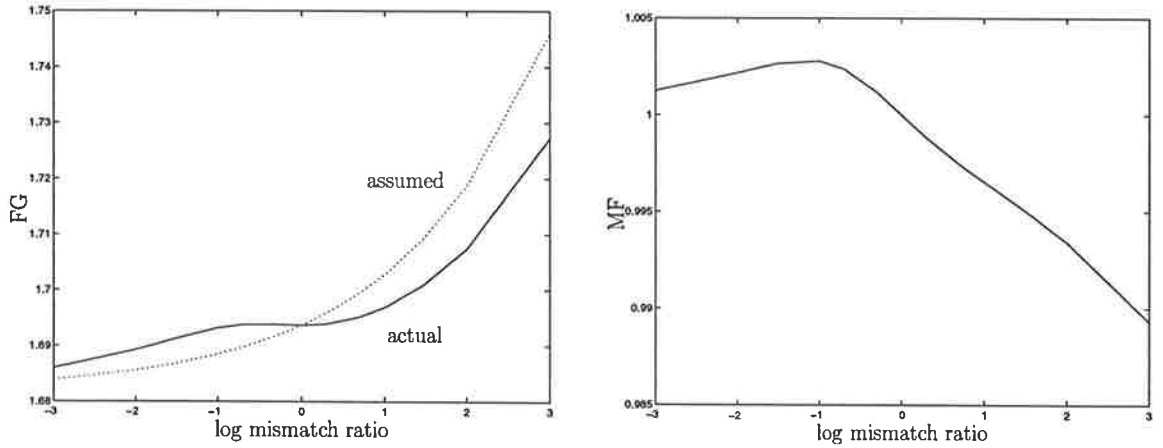
4.6.1 Process Noise Mismatch

The assumed track error covariance (TEC) produced by both filters, as shown in figure 4.1(a), increases as the mismatch ratio increases, i.e., it increases as the ratio of the assumed process noise covariance to the (constant) actual process noise covariance



(a) Assumed (b) Actual

Figure 4.1: Track error covariance with process noise mismatch



(a) Fusion gain (b) Mismatch Factor

Figure 4.2: FG and MF for mismatch in process noise covariance

increases. Close inspection of figure 4.1(a) reveals that as the assumed process noise covariance increases, the separation between the single and fused Kalman filter TEC's, representing the logarithm of the assumed FG, increases slightly. An explanation of this behaviour is provided in section 3.3.2.1.

Figure 4.1(b) shows that the actual TEC is at a minimum under matched conditions, where it corresponds to the assumed TEC, and it increases as the level of mismatch increases in either direction. As the mismatch ratio decreases from unity, the gains of both filters are reduced, and the measurements have less influence on the filter output. Therefore the influence of the process noise increases and, as the actual process noise is larger than that assumed by the filter, some of the extra actual process noise is transfered to the track error. Treating the dual sensor measurements in the fused Kalman filter as a single compressed measurement (Willner et al. 1976), the gain of the fused Kalman filter

is larger than the gain of the single sensor filter because the covariance of the compressed measurement is less than the covariance of either single measurement. Therefore the fused filter relies more on the measurements than the single sensor filter, and less of the extra actual process noise appears in its track error. Therefore, if the mismatch ratio is around unity, this lower contribution of process noise keeps the ratio of single sensor filter to dual sensor filter TEC, i.e., the actual FG, approximately constant, as shown in figure 4.2(a). As the mismatch falls significantly below unity, the measurements of both filters have less effect, so the difference in the contribution of the process noise to the TEC between the filters becomes less significant, and the reducing gap between their TEC's produces a decrease in the actual FG, eventually approaching the assumed FG from above, as shown in figure 4.2(a).

As the mismatch ratio rises significantly above unity, the filter gains increase and less process noise is transfered to the track error. However, the filter gains are becoming more dependent on the measurement noise covariance and, because this covariance is lower for the fused Kalman filter, the gain of the fused filter is increasing relative to that of the single sensor filter. Therefore the difference in the process noise contribution between the single and dual sensor filters is increasing and the dual sensor filter's TEC does not increase as rapidly as the TEC of the single sensor filter. This results in an increase in the actual FG, as shown in figure 4.2(a). Under these mismatched conditions, the covariance of the actual process noise is less than its assumed value and subsequently the TEC is also lower than its assumed value. The increase in actual TEC as the mismatch ratio increases is caused by the higher than optimal gains that transfer increasingly greater amounts of additional measurement noise to the TEC as the mismatch ratio rises.

The MF is the ratio of actual to assumed FG, and figure 4.2(b) shows how it decreases as the mismatch ratio rises above unity. These values of MF below unity indicate that the fused filter is more sensitive to errors in process noise covariance than the single sensor filter. As the mismatch ratio falls below unity, the MF initially rises above unity and then gradually falls back toward unity, indicating that the fused filter is less sensitive than the single sensor filter under these conditions. However, the MF is within just over 1% of unity for the process noise covariance errors of three orders of magnitude considered here, and may therefore be assumed to be unity over this range, i.e., the Kalman filter's sensitivity to errors in process noise covariance is not significantly affected by adding another sensor.

These results are for sensors with equal measurement noise covariances. If these covariances differ, the effects observed here will be reduced because the behaviour of the fused filter will be closer to that of a single sensor filter.

4.6.2 Measurement Noise Mismatch

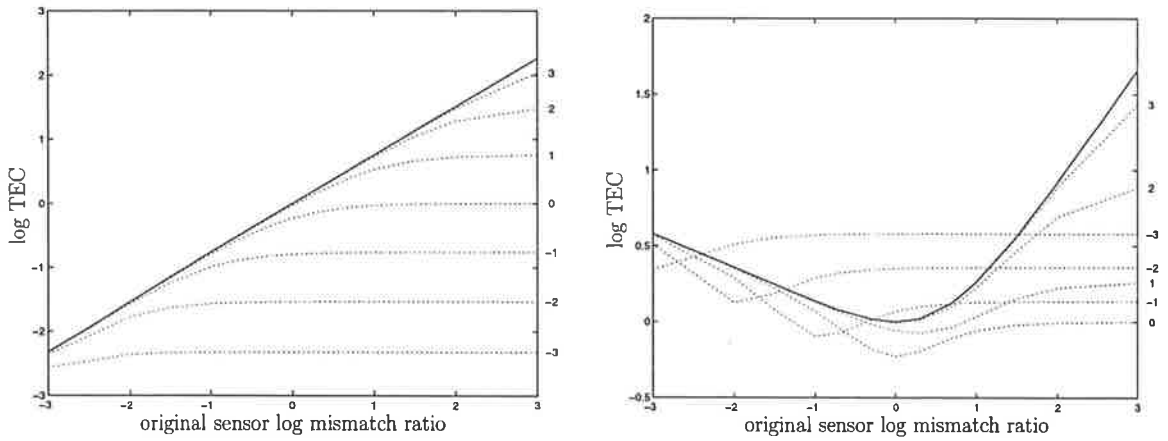
The measurement noise mismatch was considered for three different cases, equal measurement noise covariance in each sensor, an additional sensor measurement noise covariance

ten times that of the original sensor, and an additional sensor measurement noise covariance one tenth of the original sensor.

4.6.2.1 Equal Measurement Noise Covariances

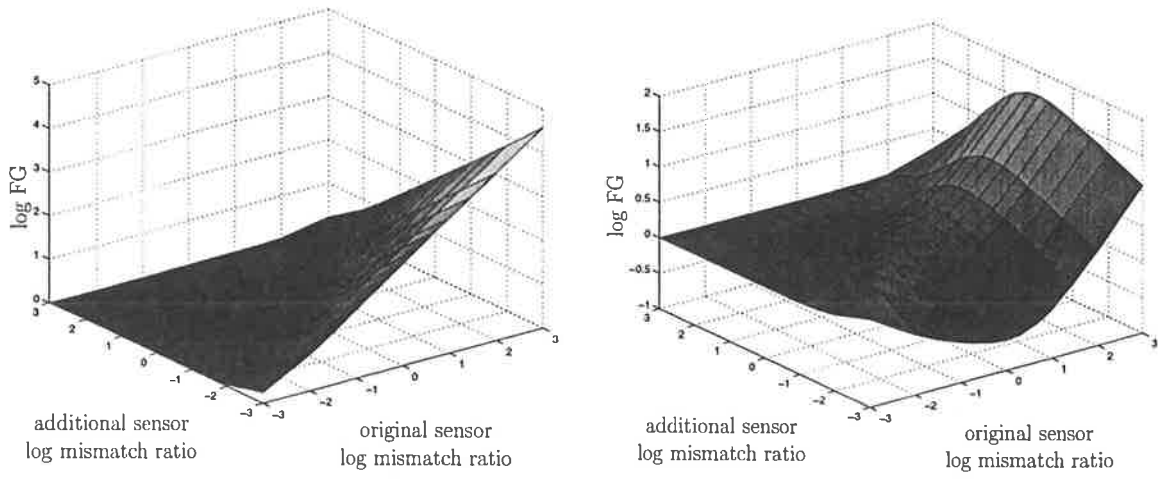
In figure 4.3, the single sensor Kalman filter track error covariance is shown as a solid line, the dotted lines indicating the TEC obtained from the fused filter. The ratio of the assumed measurement noise covariance to the actual measurement noise covariance of the data for the original sensor appears along the horizontal axis as a logarithmic scale. The same ratio for the additional sensor appears (as log values) along the right hand side of the plots, where each label indicates the appropriate (dotted) curve for that value.

Where the mismatch ratio for the original sensor is significantly less than that of the additional, the fused filter's assumed and actual TEC both approach the respective values



(a) Assumed (b) Actual

Figure 4.3: Track error covariance with measurement noise mismatch for similar sensors



(a) Assumed (b) Actual

Figure 4.4: Fusion gain with measurement noise mismatch for similar sensors

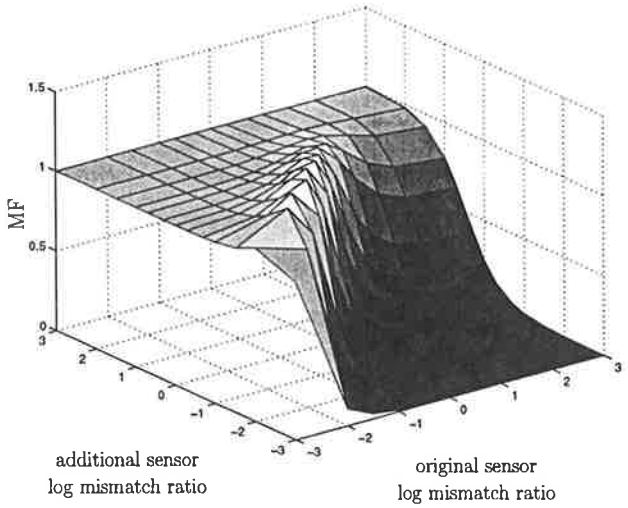


Figure 4.5: Mismatch factor with measurement noise mismatch for similar sensors

of the single sensor filter (figure 4.3). Therefore the assumed and actual FG's (figure 4.4) both approach unity, and the corresponding MF (figure 4.5) also approaches unity. This implies that, under these conditions, the fused Kalman filter has the same sensitivity to model errors as the single sensor Kalman filter. This is not surprising because the fused Kalman filter effectively operates as a single sensor filter and the additional sensor contributes little to its performance.

As the mismatch ratios of both sensors become similar, the additional sensor begins to influence the operation of the fused Kalman filter and, as illustrated in figure 4.3(a), its assumed TEC begins to flatten out. However, the single sensor filter's assumed TEC continues to rise, and therefore the assumed FG (figure 4.4(a)) gradually begins to increase. Also the ratio of the weights for each measurement in the compressed measurement of the fused filter approaches its optimal value, i.e., the compressed measurement has equal contributions from the measurements of both sensors. Therefore the covariance of the compressed measurement in the fused filter falls relative to the covariance of the single sensor measurement, and the actual TEC of the fused filter also falls relative to that of the single sensor (figure 4.3(b)), increasing the actual FG (figure 4.4(b)). This increase is greater than that of the assumed FG, and therefore the MF rises above unity (figure 4.5), indicating that the fused Kalman filter is less sensitive to measurement model error under these conditions than the single sensor Kalman filter. As the mismatch ratio in both sensors increases, this effect reduces because the gains of both filters fall and the measurement noise has less effect.

Continued increases in the mismatch ratio of the original sensor relative to the additional causes the covariance of the compressed measurement to approach the covariance of the measurement from the additional sensor, and the actual TEC of the fused filter rises toward a constant value (figure 4.3(b)). If the mismatch ratio in the original sensor is less than unity, the actual TEC of the single sensor filter is still falling toward its minimum (matched) value, and it actually drops below that of the fused filter, causing the actual

FG to fall below unity (figure 4.4(b)). As the mismatch ratio of the original sensor is increased above unity, the single sensor filter becomes mismatched and its actual TEC begins to rise, giving an increase in the actual FG.

When the mismatch ratio of the additional sensor is significantly less than that of the original sensor, the fused Kalman filter again operates as a single sensor filter, but this time using the additional sensor. Under these conditions, its assumed TEC is largely influenced by the additional sensor, and it becomes independent of the original sensor's mismatch ratio, as indicated by the horizontal dotted lines in figure 4.3(a).

The actual TEC under these conditions is larger than the assumed TEC because additional measurement noise is introduced into the TEC if the mismatch ratio is below unity, or additional process noise is introduced to the TEC if the mismatch ratio of the additional sensor is above unity, because of the measurement noise mismatch. Therefore the actual FG is less than the assumed FG and the MF falls away rapidly, indicating that in this case the fused Kalman filter is much more sensitive to measurement model error than the single sensor filter. This is expected, as the gain in the fused Kalman filter is larger than for the single filter, and therefore any errors in the measurement noise model will have a greater effect.

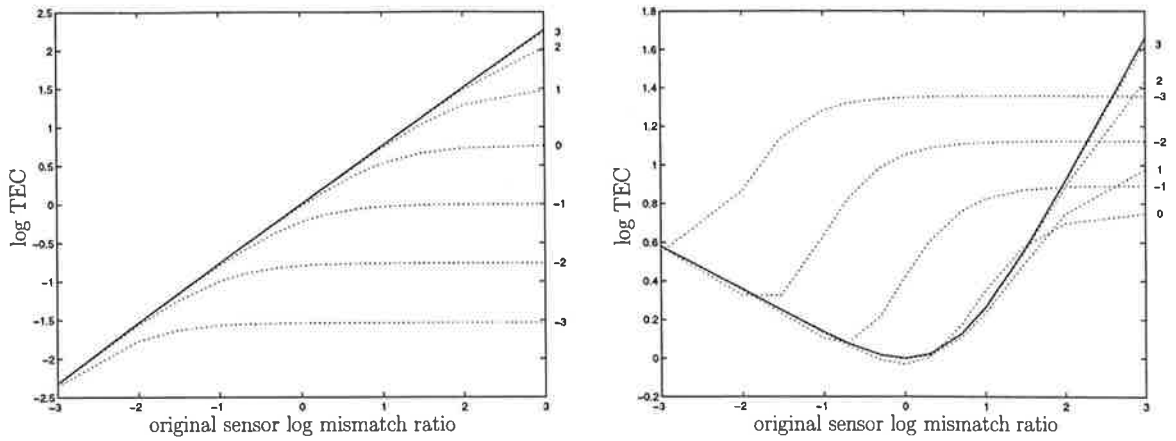
Therefore when the actual measurement noise covariance of both sensors are the same, the fused and single sensor Kalman filters exhibit the same sensitivity to errors in the measurement noise covariance if the mismatch ratio of the original sensor is significantly less than that of the additional sensor. Conversely, if the mismatch ratio of the additional sensor is less than that of the original sensor, the fused Kalman filter is more sensitive to these errors. However if the mismatch ratio of the original sensor is slightly less than or equal to that of the additional sensor, the fused Kalman filter is less sensitive than the single sensor filter to measurement noise covariance errors.

4.6.2.2 Higher Noise Covariance in Additional Sensor

The measurement noise covariance of the additional sensor was increased to ten times that of the original. Note that a unity mismatch ratio still implies matched conditions, although the absolute values of measurement noise covariance are different.

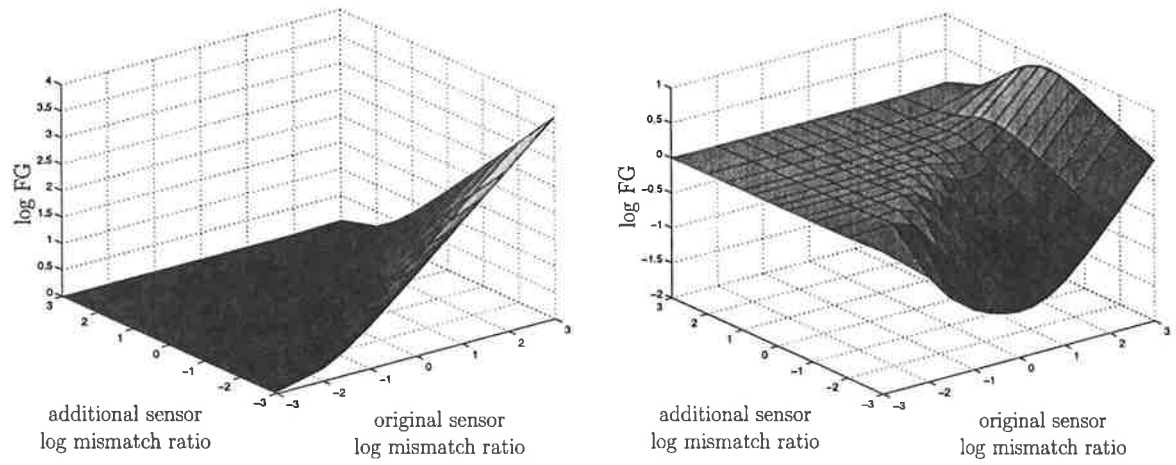
When the mismatch ratio in the original sensor is significantly lower than that of the additional sensor, the fused Kalman filter again operates as a single sensor Kalman filter using the original sensor and, as for the case of equal actual measurement noise covariances, the MF is unity (figure 4.8) and the fused Kalman filter shows the same sensitivity to measurement noise covariance errors as the single sensor Kalman filter.

The additional sensor with larger actual measurement noise covariance now begins to significantly affect the fused filter's assumed TEC when the mismatch ratios of both sensors are almost equal (figure 4.6(a)), not when the original sensor's mismatch ratio is less than the additional sensor's, as is the case for equal measurement noise covariances,



(a) Assumed (b) Actual

Figure 4.6: Track error covariance with measurement noise mismatch where the additional sensor is noisiest



(a) Assumed (b) Actual

Figure 4.7: Fusion gain with measurement noise mismatch where the additional sensor is noisiest

and the corresponding rise in assumed FG also commences at these larger mismatch ratios in the original sensor (figure 4.7(a)). For equal mismatch ratios, the actual TEC only falls slightly below the single sensor filter's actual TEC (figure 4.6(b)) because the additional sensor contributes little to the compressed measurement. The corresponding rise in actual FG is no longer significant (figure 4.7(b)), and therefore the rise in MF above unity also becomes insignificant (figure 4.8). This results in no appreciable change in the fused Kalman filter's sensitivity to measurement noise covariance errors.

As the mismatch ratio of the original sensor continues to increase, and the additional sensor mismatch ratio is less than unity, the fused filter's actual TEC rises rapidly above the still falling single sensor filter's actual TEC (figure 4.6(b)). The resulting trough

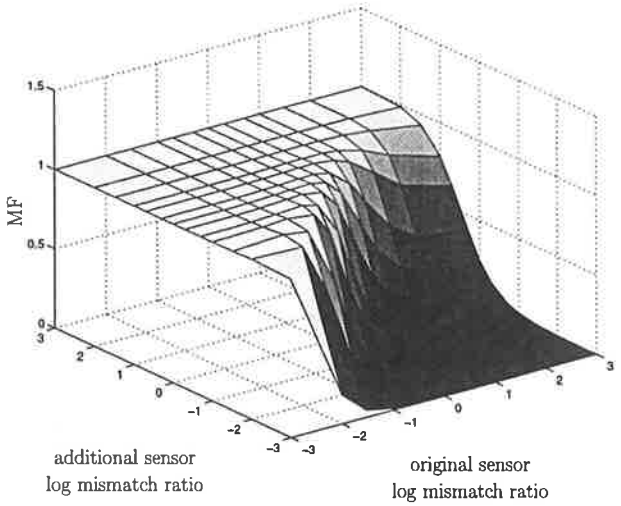


Figure 4.8: Mismatch factor with measurement noise mismatch where the additional sensor is noisiest

in the mismatched FG is now much deeper and more pronounced than for the case of equal measurement noise covariances (figure 4.7(b)). However, the assumed FG is much larger than the actual FG when the additional sensor has a lower mismatch ratio than the original sensor (figure 4.6), and the MF (figure 4.8) again falls off rapidly, indicating that the fused Kalman filter is more sensitive to errors in the measurement noise covariance than the single sensor filter.

Therefore, from a practical viewpoint, increasing the actual measurement noise covariance of the additional sensor only significantly affects the sensitivity of the fused Kalman filter when the mismatch ratio of both sensors are approximately equal. In this case, the reduction in measurement noise covariance error sensitivity becomes less significant as the actual measurement noise covariance of the additional sensor is increased above that of the original sensor.

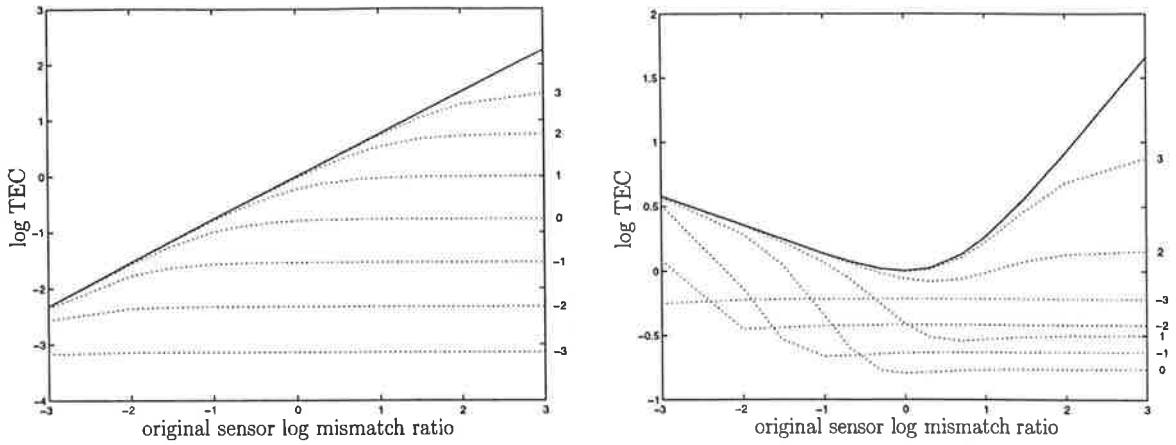
4.6.2.3 Lower Noise Covariance in Additional Sensor

The final results in this section were obtained when the measurements of the additional sensor had a measurement noise covariance one tenth that of the original sensor's.

Again the results were similar to the case of equal actual measurement noise covariance in both sensors when the mismatch ratios in the original sensor were less than those of the additional sensor, with the sensitivity of the two filters being similar.

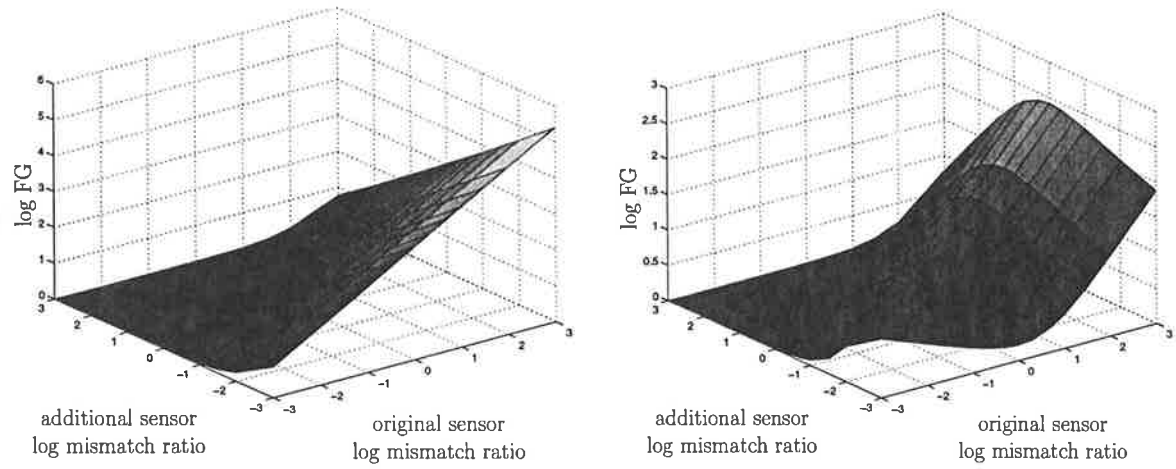
The lower noise covariance of the measurements from the additional sensor causes it to influence the assumed TEC of the fused Kalman filter when the mismatch ratio of the original sensor is much less than that of the additional (figure 4.9(a)). Therefore the assumed FG begins to increase at these lower values of original sensor mismatch (figure 4.10(a)).

As the original sensor's mismatch ratio approaches that of the additional sensor, the actual TEC of the fused filter has already reached its minimum and is beginning to



(a) Assumed (b) Actual

Figure 4.9: Track error covariance with measurement noise mismatch where the original sensor is noisiest



(a) Assumed (b) Actual

Figure 4.10: Fusion gain with measurement noise mismatch where the original sensor is noisiest

increase (figure 4.9(b)). If the mismatch ratio of the additional sensor is less than unity, the additional sensor has significantly reduced the fused filter's actual TEC below that of the single sensor filter. Therefore the actual FG (figure 4.10(b)) rises rapidly to a larger value than obtained for equal measurement noise covariance. As this increase is much greater than the increase in assumed FG, the MF increases substantially above unity (figure 4.11). Therefore reducing the actual measurement noise covariance of the additional sensor decreases the sensitivity of the fused Kalman filter when the mismatch ratios of both sensors are almost equal.

Further increases in original sensor mismatch ratio only slightly increase the fused filter's track error covariance, but the track error covariance of the single sensor filter

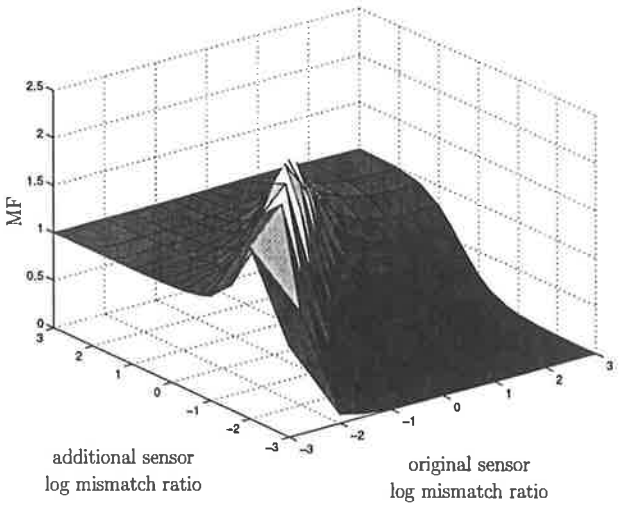


Figure 4.11: Mismatch factor with measurement noise mismatch where the original sensor is noisiest

is still falling. However, the single sensor filter's track error covariance does not drop below that of the fused, and the trough in the mismatch FG is therefore much shallower (figure 4.10(b)). However, the MF rapidly falls below unity for additional sensor mismatch ratios less than those of the original sensor, again showing increased sensitivity to measurement noise covariance errors.

The only significant difference between this result, and that for equal actual measurement noise covariance in both sensors, occurs when the mismatch ratio for original sensor is slightly less than or equal to the mismatch ratio of the additional sensor. Here the sensitivity to measurement noise covariance is reduced to an even greater extent than for sensors with equal measurement noise covariance.

4.7 Summary of Sensitivity Analysis

The fused Kalman filter shows no significant improvement or degradation in sensitivity to errors in the process noise covariance over the single sensor Kalman filter.

The sensitivities of the fused Kalman filter and single sensor Kalman filter to measurement noise covariance errors are similar if the mismatch ratio of the original sensor is smaller than that of the additional sensor. If the mismatch ratio of the additional sensor is less than that of the original sensor, the fused Kalman filter exhibits a greater sensitivity than the single sensor Kalman filter to errors in the measurement noise covariance.

When the mismatch ratio of the original sensor is marginally less than or equal to that of the additional sensor, the fused Kalman filter is less sensitive to these measurement noise covariance errors. This reduction in sensitivity becomes more significant as the actual measurement noise covariance of the additional sensor falls relative to that of the original sensor. It becomes less pronounced when the mismatch ratios of both sensors increase above unity, because the influence of the process noise increases.

CHAPTER 5

Multi-Sensor Multi-target Problem Formulation

The AFKF (chapter 3) is a useful tool for tracking single targets using measurements from multiple sensors. However, it assumes that all these measurements originate from the target being tracked. This is usually not the case in real environments, where measurements may be received from other targets, or may simply be caused by clutter and other sources of noise, i.e., false alarms.

Under these circumstances, it is necessary to determine which of the available measurements should be used to track a particular target. This problem is known as *data association*, and the peril of ignoring it has been demonstrated in section 3.3.3 where an AFKF was used on real data.

This chapter defines the multi-sensor multi-target tracking problem, specifies the structure of the estimator's observer and introduces appropriate target models. It follows Streit and Luginbuhl's formulation (Streit and Luginbuhl 1994, Streit and Luginbuhl 1995) for their Probabilistic Multi-Hypothesis Tracking (PMHT) algorithm. Differences between the single sensor formulation of Streit and Luginbuhl and this multi-sensor version are highlighted. The problem is initially restricted to single simultaneous measurements from each sensor to give a clearer explanation of the problem. This restriction is lifted, giving a formulation for asynchronous sensors, and the chapter then concludes by considering a general problem with multiple sensor models for each physical sensor.

5.1 Problem Definition

Consider a surveillance region that is monitored by $S > 1$ sensors.[1] The sensors report target activity within this region by providing measurements from detected targets. Let $\mathbf{Z}^T$ denote a batch of measurements received from the sensors over the period of time t_1

[1] Although this formulation is also valid for $S = 1$ (single sensor), only the multi-sensor problem is considered here.

to t_T. Assume that the measurements are received at times denoted $[t_1, t_2, \ldots, t_T]$, where $t_i \leqslant t_{i+1}$ for all i. Then the batch of measurements may be denoted

$$\mathbf{Z}^T \equiv [\mathbf{z}_{t_1}, \mathbf{z}_{t_2}, \ldots, \mathbf{z}_{t_T}] \tag{5.1}$$

where $\mathbf{z}_{t_i}$ is the measurement scan at time t_i. Each measurement scan comprises of one or more measurements produced at time t_i, i.e.,

$$\mathbf{z}_{t_i} \equiv \left[\mathbf{z}_{t_i}^{(1)\mathsf{T}}, \mathbf{z}_{t_i}^{(2)\mathsf{T}}, \ldots, \mathbf{z}_{t_i}^{(n_{t_i})\mathsf{T}}\right]^{\mathsf{T}} \tag{5.2}$$

where $\mathbf{z}_{t_i}^{(r)}$ is the r^{th} measurement in the scan at time t_i and n_{t_i} is the total number of measurements in that particular scan. It is assumed that the dimension of all measurements from a particular sensor $N_Z^{(s)}$, $s = 1, 2, \ldots, S$, is constant, but it may differ between sensors. Note that if each scan contains every measurement occurring at that time, then the condition $t_i < t_{i+1}$ holds for all i. The possibility of $t_i = t_{i+1}$, introduced above, allows the batch of measurements to be processed sequentially, or in related groups such as all measurements with the same latency (i.e., from the same sensor).

5.2 Observer Structure

The *observer* estimates the state of the system, i.e., the dynamic state of all targets over the period t_0 to t_T, from the batch of measurements. The structure of the observer is based on a state space representation of the target states, and this forms the framework for the material in the following chapters.

Let $M > 1$ denote the maximum number of targets[2] in the surveillance region during the time interval t_1 to t_T.[3] Let t_0 be an arbitrary time, prior to the batch of measurements, at which the initial *a priori* state for each target may be defined. The assumed *a priori* state vector for the m^{th} target, $m \in \{1, 2, \ldots, M\}$, is denoted $\bar{\mathbf{x}}_{t_0}^{(m)}$, and its covariance $\bar{\mathbf{\Sigma}}_{t_0}^{(m)}$ provides a measure of the uncertainty between this value and the true target state. In practice, these state vectors and their covariances are obtained from a track initialisation algorithm.

The collection of all target states for all times t_0 to t_T, $\mathbf{X}^T$, is denoted

$$\mathbf{X}^T \equiv \left[\mathbf{X}^{(1)\mathsf{T}}, \mathbf{X}^{(2)\mathsf{T}}, \ldots, \mathbf{X}^{(M)\mathsf{T}}\right]^{\mathsf{T}} \qquad \mathbf{X}^{(m)} \equiv \left[\mathbf{x}_{t_0}^{(m)}, \mathbf{x}_{t_1}^{(m)}, \ldots, \mathbf{x}_{t_T}^{(m)}\right] \tag{5.3}$$

where $\mathbf{X}^{(m)}$ denotes the set of state vectors for target m, and $\mathbf{x}_{t_i}^{(m)}$ denotes the state vector for target m at time t_i. The dimension of all state variables for the m^{th} target model, $N_X^{(m)}$,

[2] $M = 1$ corresponds to single target tracking in the absence of false alarms. This problem is trivial as all measurements are, by definition, assigned to the target.

[3] Additional models may be used to represent false alarms caused by clutter or other sources of noise.

is constant. However, this dimension may vary between target models, e.g., in a generally benign environment, the target models may only require position and velocity in their state vectors, but additional models that include acceleration states may be added for occasional highly manoeuvring targets. The collection of target states may be partitioned in time, giving the alternative representation

$$\mathbf{X}^T \equiv [\mathbf{X}_{t_0}, \mathbf{X}_{t_1}, \ldots, \mathbf{X}_{t_T}] \qquad \mathbf{X}_{t_i} \equiv \left[\mathbf{x}_{t_i}^{(1)\mathsf{T}}, \mathbf{x}_{t_i}^{(2)\mathsf{T}}, \ldots, \mathbf{x}_{t_i}^{(M)\mathsf{T}}\right]^{\mathsf{T}} \tag{5.4}$$

where $\mathbf{X}_{t_i}$ is the set of state variables for all targets at time t_i.

The objective is to estimate the state of each target from the batch of measurements. The problem is that the data is not complete, i.e., although the measurements are present, the information indicating which measurements belong to each target is generally missing. This missing information is the set of *measurement to target assignments*, or simply the *target assignments*, and is denoted

$$\mathbf{K}^T \equiv [\mathbf{k}_{t_0}, \mathbf{k}_{t_1}, \ldots, \mathbf{k}_{t_T}] \qquad \mathbf{k}_{t_i} \equiv \left[k_{t_i}^{(1)}, k_{t_i}^{(2)}, \ldots, k_{t_i}^{(n_{t_i})}\right]^{\mathsf{T}} \tag{5.5}$$

where $k_{t_i}^{(r)} = m$, $m \in \{1, 2, \ldots, M\}$, denotes the target that produced the measurement $\mathbf{z}_{t_i}^{(r)}$. $\mathbf{k}_{t_i}$ is the set of all target assignments in the scan at time t_i. Note that as no measurements are available at time t_0, no target assignments are present at this time, i.e., $\mathbf{k}_{t_0}$ is the empty set $\{\phi\}$.

The unknown target assignments are required to estimate the target states and therefore need to be determined. Streit and Luginbuhl (1995) approached this problem by including the target assignments in the observer with the target states. The same technique is used here, and the structure of the observer $\mathbf{O}^T$ is therefore the same as that used by Streit and Luginbuhl, i.e.,

$$\mathbf{O}^T \equiv [\mathbf{O}_{t_0}, \mathbf{O}_{t_1}, \ldots, \mathbf{O}_{t_T}] \qquad \mathbf{O}_{t_i} \equiv [\mathbf{X}_{t_i}, \mathbf{k}_{t_i}] \tag{5.6}$$

where $\mathbf{O}_{t_i}$ is the observer state for the measurement scan at time t_i.

Aside: It is assumed that each measurement is assigned to exactly one target. Measurements that don't belong to any real targets, i.e., false alarms, may be assigned to additional target models representing clutter, noise, etc. In this way, all measurements are assigned to at least one target model. The problem of assigning a single measurement to multiple target models may arise when tracking a cluster (formation) of targets with different resolution sensors. The high resolution sensors may be able to resolve the individual targets within the cluster, but the lower resolution sensor may only see a single target, the entire cluster. In this situation, the measurement from the low resolution sensor should really be assigned to all the targets in the cluster. However, the tracking filter will virtually ignore the low sensor measurements because they are less accurate

than the high resolution measurements. Assigning the low resolution measurements to a target will therefore have little effect on the estimated track for that target, and therefore these measurements may be assigned to any target within the cluster without significantly affecting the track estimates. If the low resolution sensor provides additional information, it may be able to resolve the targets using this information and therefore produce a measurement for each target. For example, radars usually have poor angular resolution but provide additional information in the form of high resolution range measurements. If the targets can't be resolved by this additional information, the single available measurement will be assigned to one of the targets within the cluster. However, the target to which this measurement is assigned will depend on the actual measurement itself, and the current estimated position of each target in the cluster, at the time of the measurement. The target assignments for the measurements from this sensor will vary over time, therefore it is very likely that each target in the cluster will receive this additional information at various times during tracking. Given the above argument, it is considered that the restriction of assigning each measurement to exactly one target does not present a significant limitation.

5.3 Synchronised Sensors

The problem is initially restricted to simultaneous single measurements from each sensor, i.e., each measurement scan contains exactly one measurement from each sensor. This restriction provides a simpler explanation of the problem. In a practical sense, this restriction corresponds to the situation where the sensors all report measurements at the same time, i.e., they are synchronised, and no sensor misses any target detection. This situation is somewhat artificial, because in real scenarios different types of sensors would be deployed to improve overall tracking accuracy and decrease the susceptibility of the system to clutter and intentional counter measures. Different sensors would be unlikely to have the same measurement rates and would occasionally fail to detect targets, particularly if some of the targets are small. However, this restricted problem provides a useful and relatively simple introduction to the problem. This restricted multi-sensor problem differs from that of Streit and Luginbuhl (1995), who assumed a single sensor capable of producing multiple simultaneous measurements.

The restriction fixes the number of measurements in each scan to S, i.e., $n_{t_i} = S$, $i = 1, 2, \ldots, T$. As each measurement scan contains exactly one measurement from each sensor, the measurement vectors and target assignments within each scan may be referenced by their sensor number, i.e., $\mathbf{z}_{t_i}^{(s)}$ and $k_{t_i}^{(s)}$ respectively.

This problem is illustrated by the simple example in figure 5.1. At the top of this figure are two target trajectories, *target 1* and *target 2*. The initial state of each target at time t_0 and the state of each target at the measurement times $t_1, t_2, \ldots, t_6$ are contained in the respective state variables, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. The targets are monitored by two sensors,

sensor 1 and *sensor 2*, each producing a measurement at each measurement time, i.e., $\mathbf{z}_{t_i}^{(1)}$ and $\mathbf{z}_{t_i}^{(2)}$. Sensor 1 receives its first three measurements (denoted as $\times$'s superimposed on partial target trajectories in figure 5.1) from target 1 and the last three from target 2. This is reflected in the appropriate target assignment values of $k_{t_i}^{(1)}$. Sensor 2 sees the other target at each measurement time. To estimate the trajectory of target 1, the measurements corresponding to all $k_{t_i}^{(s)}$'s with values of 1, i.e., $\mathbf{z}_{t_1}^{(1)}$, $\mathbf{z}_{t_2}^{(1)}$, $\mathbf{z}_{t_3}^{(1)}$, $\mathbf{z}_{t_4}^{(2)}$, $\mathbf{z}_{t_5}^{(2)}$ and $\mathbf{z}_{t_6}^{(2)}$, are required. For target 2, on the other hand, measurements with $k_{t_i}^{(s)}$'s of 2, i.e., $\mathbf{z}_{t_1}^{(2)}$, $\mathbf{z}_{t_2}^{(2)}$, $\mathbf{z}_{t_3}^{(2)}$, $\mathbf{z}_{t_4}^{(1)}$, $\mathbf{z}_{t_5}^{(1)}$ and $\mathbf{z}_{t_6}^{(1)}$, are used. Therefore the target assignments act as pointers to the appropriate measurements to use for estimating each target trajectory. Note that, at any time, both sensors could have seen the same target. In this case, both measurements would be assigned to one target and no measurements to the other.

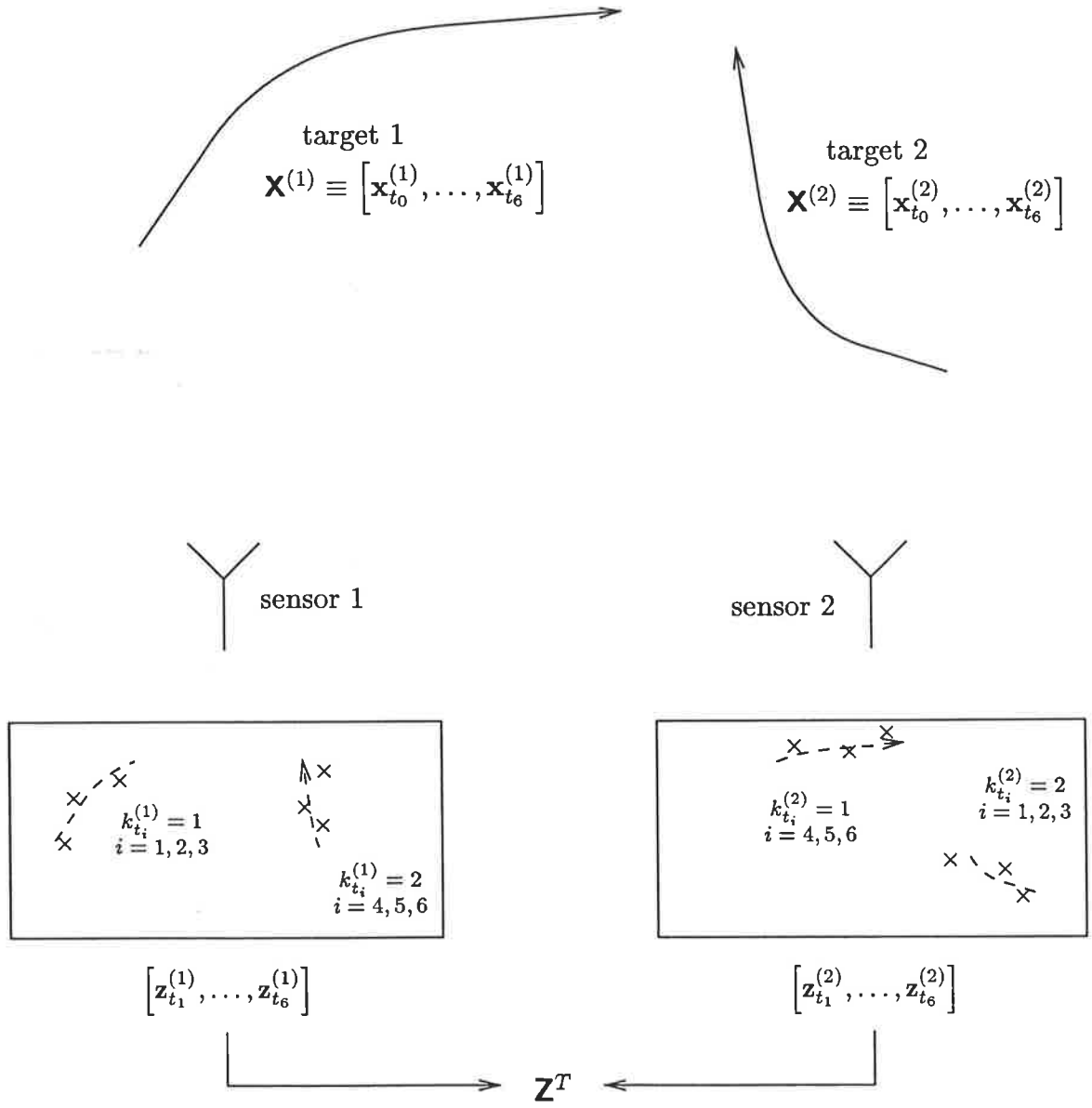


Figure 5.1: Tracking two targets with measurements from two sensors

5.3.1 Models

A separate state space model is used to represent the behaviour of each of the M targets. These *target models* each contain a *process model* to describe the target's dynamic behaviour and a *measurement model* to represent the measurements and their relationship to the target states.

The general form for the process or *dynamic model* for target m is

$$\mathbf{x}_{t_i}^{(m)} = f^{(m)}\left(t_i, \mathbf{x}_{t_{i-1}}^{(m)}, \mathbf{w}_{t_{i-1}}^{(m)}\right) \qquad i = 1, 2, \ldots, T \tag{5.7}$$

where $f^{(m)}(\cdot)$ is the target dynamic function that depends on time, the previous target state and the target's process noise $\mathbf{w}_{t_{i-1}}^{(m)}$. It is assumed that the dynamic models for each target are independent, although this assumption is not strictly correct if the targets are moving in formation. However, over all possible scenarios, no two targets can be assumed to always be in formation, and therefore may be assumed to be statistically independent.

The general measurement model for the measurement from sensor s at time t_i, $\mathbf{z}_{t_i}^{(s)}$, is

$$\mathbf{z}_{t_i}^{(s)} = \begin{cases} h^{(1,s)}\left(t_i, \mathbf{x}_{t_i}^{(1)}, \mathbf{v}_{t_i}^{(1,s)}\right) & \text{measurement from target 1} \\ h^{(2,s)}\left(t_i, \mathbf{x}_{t_i}^{(2)}, \mathbf{v}_{t_i}^{(2,s)}\right) & \text{measurement from target 2} \\ \vdots & \vdots \\ h^{(M,s)}\left(t_i, \mathbf{x}_{t_i}^{(M)}, \mathbf{v}_{t_i}^{(M,s)}\right) & \text{measurement from target M} \end{cases} \tag{5.8}$$

where $h^{(m,s)}(\cdot)$ is the time variant measurement function for target m and sensor s. This function depends on the target state $\mathbf{x}_{t_i}^{(m)}$ and measurement noise $\mathbf{v}_{t_i}^{(m,s)}$. The measurement is assumed to originate from a single target, therefore only one of the alternatives in (5.8) is valid. This implies that the measurement model for this measurement will only appear in one target model, the one pointed to by the value contained in the target assignment $k_{t_i}^{(s)}$. The measurement model may be viewed as a finite mixture of the M measurement functions at time t_i. In this case, all the mixing parameters are zero with the exception of the one corresponding to the value of $k_{t_i}^{(s)}$ which is unity, i.e.,

$$z_{t_i}^{(s)} = \sum_{m=1}^{M} \pi_{t_i}^{(m)} h^{(m,s)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,s)}\right) \tag{5.9}$$

where the mixing parameters $\pi_{t_i}^{(m)}$ are defined as

$$\pi_{t_i}^{(m)} = \begin{cases} 1 & \text{if } k_{t_i}^{(s)} = m \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

Consistent with Streit and Luginbuhl (1995), the measurement model for sensor s in

target model m may be written

$$\mathbf{z}_{t_i}^{(s)} = h^{(m,s)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,s)}\right)\Big|_{m=k_{t_i}^{(s)}} \qquad i = 1, 2, \ldots, T, \quad s = 1, 2, \ldots, S \tag{5.11}$$

which states that the model is valid given that it originated from target m. This may be viewed as a conditional model, i.e., it is only valid if the measurement was produced by the target to which the measurement belongs.

This measurement model differs from that used by Streit and Luginbuhl (1995) because it depends on the sensor that produced the measurement. Streit and Luginbuhl's measurement model depended only on the target and time, a perfectly reasonable assumption given that their work only considered a single sensor with possibly simultaneous measurements. In the formulation presented here, measurements from multiple dissimilar sensors are allowed. The measurement model describes the relationship between the measurement and the appropriate state vector. The measurement vectors from each sensor may vary, e.g., a radar may provide bearing and range measurements and a passive electro-optical sensor may provide only bearing measurements. Also the components and structure of the state vector may vary between target models, e.g., second or third order dynamic models. Therefore this relationship between the measurement and state vector will depend on both the sensor and the target model.

The measurement noise also depends on both the sensor and the target model. Different types of sensors will have different resolutions and contribute different types and levels of measurement noise. Also different types of target will also affect the measurement noise, for example a large target at close range may have several dominant well separated radar reflectors, resulting in more uncertainty in the location of the target's centroid than would be the case for a small distant target. In general, this noise cannot be separated into sensor dependent and target dependent components because the sensor component is affected by the target, and vice versa. For example, a small target will decrease the signal to noise ratio and therefore increase the measurement noise from a sensor, and any target dependent component of the noise will be different for radar and optical measurements because these sensors use different physical processes to produce their measurements. It is worth noting that differences in the process model between target types are often consistent with the measurement model, e.g., any manoeuvring target is likely to be physically small while larger targets are more likely to be benign.

5.3.1.1 Linear Gaussian Models

Often models are simplified by assuming a linear system with additive Gaussian noise. Under these conditions, the dynamic or process noise for target model m at time t_i, denoted as $\mathbf{w}_{t_i}^{(m)}$, is assumed to be Gaussian distributed with zero mean and known covariance $\mathbf{Q}_{t_i}^{(m)}$. $\mathbf{w}_{t_i}^{(m)}$ is also assumed to be uncorrelated over time and independent

between target models, i.e.,

$$\mathrm{E}\left[\mathbf{w}_{t_i}^{(m)}\mathbf{w}_{t_j}^{(\rho)\mathsf{T}}\right] = \begin{cases} \mathbf{Q}_{t_i}^{(m)} & \text{if } m = \rho \text{ and } i = j \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{5.12}$$

The assumption of independent noise between target models is consistent with the independent target assumption in the general form of the model. Under these conditions (5.7) becomes

$$\mathbf{x}_{t_i}^{(m)} = \mathbf{F}_{t_{i-1}}^{(m)}\mathbf{x}_{t_{i-1}}^{(m)} + \mathbf{w}_{t_{i-1}}^{(m)} \qquad i = 1, 2, \ldots, T \tag{5.13}$$

where $\mathbf{F}_{t_{i-1}}^{(m)}$ represents the transition matrix describing the target dynamics from time t_{i-1} to t_i.

As for the general measurement model (5.11), the linear Gaussian measurement model is a function of the sensor, target and time. The measurement noise $\mathbf{v}_{t_i}^{(m,s)}$ is assumed to be Gaussian distributed with zero mean and known covariance $\mathbf{R}_{t_i}^{(m,s)}$. The measurement noise is assumed to be uncorrelated in time and independent between measurement models and sensors, i.e.,

$$\mathrm{E}\left[\mathbf{v}_{t_i}^{(m,s)}\mathbf{v}_{t_j}^{(\rho,\nu)\mathsf{T}}\right] = \begin{cases} \mathbf{R}_{t_i}^{(m,s)} & \text{if } m = \rho,\ s = \nu \text{ and } i = j \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{5.14}$$

The measurement noise $\mathbf{v}_{t_i}^{(m,s)}$ is dependent on both the sensor and the target, and it is reasonable to expect some correlation between the measurement noises when either the target model or the sensor are the same, i.e.,

$$\mathrm{E}\left[\mathbf{v}_{t_i}^{(m,s)}\mathbf{v}_{t_i}^{(m,\nu)\mathsf{T}}\right] \neq 0 \quad \text{or} \quad \mathrm{E}\left[\mathbf{v}_{t_i}^{(m,s)}\mathbf{v}_{t_i}^{(\rho,s)\mathsf{T}}\right] \neq 0.$$

In practice, it is likely that this will be less than the measurement noise covariance $\mathbf{R}_{t_i}^{(m,s)}$. Therefore, for simplicity, it is assumed that (5.14) holds, and (5.11) may be written

$$\mathbf{z}_{t_i}^{(s)} = \mathbf{H}_{t_i}^{(m,s)}\mathbf{x}_{t_i}^{(m)} + \mathbf{v}_{t_i}^{(m,s)}\Big|_{m=k_{t_i}^{(s)}} \qquad i = 1, 2, \ldots, T, \quad s = 1, 2, \ldots, S \tag{5.15}$$

where, at time t_i, $\mathbf{H}_{t_i}^{(m,s)}$ represents the measurement matrix mapping the state space of target m onto the measurement space of the s^{th} sensor.

5.4 Asynchronous Sensors

Section 5.3 defined a multi-sensor multi-target tracking problem in which each measurement scan contained exactly one measurement from each sensor. This type of problem

is unlikely in practice, particularly for systems using dissimilar sensors. Different types of sensors use different physical processes to detect targets, and the accuracy, resolution, detection range and other sensor parameters may all vary between such sensors. Different sensors may also have different measurement rates, depending upon the complexity of its internal processing, mechanical limitations or the type of application for which it is intended. Some sensors, such as surveillance radars, are capable of detecting multiple targets within a single scan, and at times, particularly with weak or fluctuating targets, a sensor may miss detections. Therefore a typical measurement scan may contain a single measurement from some sensors, multiple measurements from others and no measurements from the remaining sensors.

Under these conditions, the sensors are operating asynchronously and may be referred to as *asynchronous sensors*. An appropriate problem formulation and observer structure for asynchronous sensors has been presented in sections 5.1 and 5.2. The primary difference between this formulation and that for synchronous sensors (section 5.3) is the method of indexing the measurements (5.2), and the target assignments (5.5) within a measurement scan.

These changes to measurement and target assignments only affect the measurement models (i.e., (5.11) and (5.15)). The asynchronous measurement model is therefore written

$$\mathbf{z}_{t_i}^{(r)} = h^{(m,s)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,s)}\right)\Big|_{m=k_{t_i}^{(r)}} \qquad i = 1, 2, \ldots, T, \quad r = 1, 2, \ldots, n_{t_i} \tag{5.16}$$

and for a linear Gaussian system, it becomes

$$\mathbf{z}_{t_i}^{(r)} = \mathbf{H}_{t_i}^{(m,s)}\mathbf{x}_{t_i}^{(m)} + \mathbf{v}_{t_i}^{(m,s)}\Big|_{m=k_{t_i}^{(r)}} \qquad i = 1, 2, \ldots, T, \quad r = 1, 2, \ldots, n_{t_i} \tag{5.17}$$

with the covariance of the measurement noise denoted $\mathbf{R}_{t_i}^{(m,s)}$. Note that s still denotes the sensor from which the measurement is received. It is assumed that this is known for all measurements, a reasonable assumption in practice.

5.5 General Problem Definition

The multi-sensor multi-target problem formulation presented thus far has been concerned with estimating unknown target tracks from measurements of unknown origin. It has been assumed that it is known which sensor produced each measurement, and although this information is generally available, it may be desirable to have multiple measurement models for particular sensors. This may be useful where the sensor operating characteristics vary with changes in conditions, or even the state of a target. For example, the measurement noise from a radar may increase if the signal to noise ratio is reduced by cloud cover or precipitation, or the uncertainty in optical measurements can increase if large targets are close to the sensor (*see* section 3.3.3.3). In these cases it would be desirable to select the

most appropriate measurement model for the conditions.

Therefore the problem formulation is generalised to one of estimating the target tracks when both the target and sensor model associations of each measurement are unknown. In this general formulation, S no longer represents the number of physical sensors but the number of virtual sensors or *sensor models.* Each physical sensor may be represented by several different sensor models, each designed for a different set of sensor operating conditions. Therefore these sensor models, $1, 2, \ldots, S$ are partitioned into $n_{\mathcal{S}}$ mutually exclusive *sensor sets* that each represent a physical sensor, i.e., $\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \ldots, \mathcal{S}^{(n_S)}$. Each sensor set contains the number (index) of the sensor models that are included in that set. For example, a radar and optical system may use two sensor models for the radar, one for low and the other for high signal to noise targets, and the optical sensor may use one for normal operation and another specifically for targets at close range. In this system, the sensor models may be numbered $1, 2, 3$ and 4 as presented above. Then the radar sensor set is given by $\mathcal{S}^{(1)} = \{1, 2\}$ and the optical sensor set is $\mathcal{S}^{(2)} = \{3, 4\}$. It is assumed that the sensor set $\mathcal{S}^{(s)}$ for a particular measurement is known, i.e., the physical sensor producing the measurement is known.

The set $\mathcal{Z}_{t_i}^{(s)}$ contains the numbers or indices of the measurements in the scan at time t_i from the physical sensor represented by the sensor set $\mathcal{S}^{(s)}$. The number of measurement indices in this set is denoted $n_{\mathcal{Z}_{t_i}^{(s)}}$. In the above example, if the first and third measurements in the measurement scan at time t_i are from a surveillance radar (sensor set $s = 1$), then $n_{\mathcal{Z}_{t_i}^{(1)}} = 2$ and $\mathcal{Z}_{t_i}^{(1)} = \{1, 3\}$.

To avoid unnecessary repetition, only the differences between the asynchronous sensor and general formulations will be presented.

5.5.1 General Observer Structure

The observer in (5.6) is generalised by adding a set of *measurement to sensor assignments,* or *sensor assignments,* to each scan observer, i.e.,

$$\mathbf{O}^T \equiv [\mathbf{O}_{t_0}, \mathbf{O}_{t_1}, \ldots, \mathbf{O}_{t_T}] \qquad \mathbf{O}_{t_i} \equiv [\mathbf{X}_{t_i}, \mathbf{k}_{t_i}, \mathbf{l}_{t_i}] \tag{5.18}$$

where $\mathbf{l}_{t_i}$ denotes the sensor assignments for the measurement scan at time t_i. The sensor assignment for a particular measurement contains the number or index of the sensor model to which that measurement is assigned, i.e., the sensor model producing the measurement that is most appropriate for the current operating conditions of the sensor. The complete set of sensor assignments is defined as

$$\mathbf{L}^T \equiv [\mathbf{l}_{t_0}, \mathbf{l}_{t_1}, \ldots, \mathbf{l}_{t_T}] \qquad \mathbf{l}_{t_i} \equiv \left[l_{t_i}^{(1)}, l_{t_i}^{(2)}, \ldots, l_{t_i}^{(n_{t_i})}\right]^{\mathsf{T}} \tag{5.19}$$

where $l_{t_i}^{(r)} = p$, with $p \in \mathcal{S}^{(s)}$ and $\mathcal{S}^{(s)} \subset \{1, 2, \ldots, S\}$. Therefore the value of each sensor assignment gives the most appropriate sensor model for the corresponding measurement.

The range of possible values for each sensor assignment is limited to the sensor models available for the sensor providing the measurement, i.e., those contained in the sensor set $\mathcal{S}^{(s)}$ where $s \in \{1, 2, \ldots, n_{\mathcal{S}}\}$.

It is assumed that each measurement is assigned to exactly one sensor model. In a practical sense, this is equivalent to each sensor having only a single set of characteristics for a particular target at the time of the measurement. This is reasonable because at any time, and for a particular target, the sensor will have a specific set of characteristics, e.g., it will have a single value of noise covariance at any particular time.

As for the target assignments, $\mathbf{l}_{t_0} = \{\phi\}$ because no measurements are available at time t_0.

5.5.2 General Models

The general form of the process model introduced in section 5.3.1 for synchronous sensors (5.7) is suitable for the general problem definition. Likewise, (5.13) is appropriate for linear Gaussian dynamic systems.

The measurement model for synchronous sensors (5.8) was described as a finite mixture of the measurement models for each target model. In this general formulation, not only are the measurement models for each target model considered, but also the set of different sensor models for the sensor producing the measurement, i.e.,

$$\mathbf{z}_{t_i}^{(r)} = \begin{cases} h^{(1,s_1)}\left(t_i, \mathbf{x}_{t_i}^{(1)}, \mathbf{v}_{t_i}^{(1,s_1)}\right) & \text{measurement from target 1 and sensor } s_1 \\ \vdots & \vdots \\ h^{(1,s_n)}\left(t_i, \mathbf{x}_{t_i}^{(1)}, \mathbf{v}_{t_i}^{(1,s_n)}\right) & \text{measurement from target 1 and sensor } s_n \\ h^{(2,s_1)}\left(t_i, \mathbf{x}_{t_i}^{(2)}, \mathbf{v}_{t_i}^{(2,s_1)}\right) & \text{measurement from target 2 and sensor } s_1 \\ \vdots & \vdots \\ h^{(M,s_n)}\left(t_i, \mathbf{x}_{t_i}^{(M)}, \mathbf{v}_{t_i}^{(M,s_n)}\right) & \text{measurement from target M and sensor } s_n \end{cases} \tag{5.20}$$

where $s_1, \ldots, s_n$ denote the sensor models in the sensor set $\mathcal{S}^{(s)}$. The finite mixture of measurement models now becomes

$$z_{t_i}^{(r)} = \sum_{m=1}^{M} \sum_{p \in \mathcal{S}^{(s)}} \pi_{t_i}^{(m)} \xi_{t_i}^{(p)} h^{(m,p)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,p)}\right) \tag{5.21}$$

where the mixing parameters are now $\pi_{t_i}^{(m)} \xi_{t_i}^{(p)}$ (cf. (5.9)), with $\pi_{t_i}^{(m)}$ defined as

$$\pi_{t_i}^{(m)} = \begin{cases} 1 & \text{if } k_{t_i}^{(r)} = m \\ 0 & \text{otherwise} \end{cases} \tag{5.22}$$

and $\xi_{t_i}^{(p)}$ as

$$\xi_{t_i}^{(p)} = \begin{cases} 1 & \text{if } l_{t_i}^{(r)} = p \\ 0 & \text{otherwise.} \end{cases} \tag{5.23}$$

Therefore the measurement model for measurement $\mathbf{z}_{t_i}^{(r)}$ for sensor model p and target model m is written

$$\mathbf{z}_{t_i}^{(r)} = h^{(m,p)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,p)}\right)\Big|_{\substack{m=k_{t_i}^{(r)} \\ p=l_{t_i}^{(r)}}} \qquad i = 1, 2, \ldots, T, \quad r = 1, 2, \ldots, n_{t_i} \tag{5.24}$$

which implies that the target assignments and sensor assignments are known.

For linear Gaussian systems, the general measurement model is

$$\mathbf{z}_{t_i}^{(r)} = \mathbf{H}_{t_i}^{(m,p)}\mathbf{x}_{t_i}^{(m)} + \mathbf{v}_{t_i}^{(m,p)}\Big|_{\substack{m=k_{t_i}^{(r)} \\ p=l_{t_i}^{(r)}}} \qquad i = 1, 2, \ldots, T, \quad r = 1, 2, \ldots, n_{t_i} \tag{5.25}$$

with the covariance of the measurement noise $\mathbf{v}_{t_i}^{(m,p)}$ denoted $\mathbf{R}_{t_i}^{(m,p)}$ (cf. 5.15). A similar model has been considered by Giannopoulos, Streit and Swaszek (1996).

Usually it is only the noise component of the model that will vary between sensor models for a particular physical sensor, because this noise represents the resolution, accuracy, sensor noise, etc. of the sensor, and it is these characteristics that are most likely to change.

CHAPTER 6

Multi-Sensor Probabilistic Multi-Hypothesis Tracking (msPMHT)

The *multi-sensor Probabilistic Multi-Hypothesis Tracking* (msPMHT) algorithm is based on the original Probabilistic Multi-Hypothesis Tracking (PMHT) algorithm developed by Streit and Luginbuhl (1993). This algorithm uses soft or probabilistic measurement to target assignments instead of the hard assignments used in Multiple Hypothesis Tracking (MHT) algorithms. This overcomes the exponential complexity problems associated with the MHT algorithm, because each measurement is apportioned to each target by these probabilistic target assignments. Therefore the complexity of the PMHT is determined by the number of targets, which is fixed at a value at least as large as the maximum number of targets expected. The target state estimates are determined by estimating the probability distributions of the target assignments and states. It is a batch algorithm that may be solved iteratively.

The following algorithm development follows from the problem definition, observer structure and models introduced in chapter 5. While largely the same as the approach used by Streit and Luginbuhl (1995), the complete derivation is given both for completeness and to highlight precisely where this work generalises the PMHT algorithm.

The derivation commences considering the restricted synchronous measurement problem formulated in section 5.3.

6.1 Observer Likelihood

As discussed in section 5.2, the observer consists of a set of scan observers, one for each measurement scan. Each of the scan observers $\mathbf{O}_{t_i} \equiv [\mathbf{X}_{t_i}, \mathbf{k}_{t_i}]$, $i = 1, 2, \ldots, T$, contains a continuous component, the target states, and a discrete component, the target assignments. The objective is to estimate the target states by finding the value of the observer

(target states and target assignments) that maximise the likelihood or probability of the observer. As the measurements are given and provide the inputs to the observer, it is really the conditional probability of the observer, conditioned on the measurements, that should be maximised. Therefore the estimate of the observer $\hat{\mathbf{O}}^T$ is defined as

$$\hat{\mathbf{O}}^T \triangleq \arg\max_{\mathbf{O}^T} \mathrm{p}\left(\mathbf{O}^T \,|\, \mathbf{Z}^T\right) \equiv \arg\max_{\mathbf{O}^T} \mathrm{p}\left(\mathbf{O}^T, \mathbf{Z}^T\right) \equiv \arg\max_{\mathbf{O}^T} \mathrm{p}\left(\mathbf{O}^T\right) \mathrm{p}\left(\mathbf{Z}^T \,|\, \mathbf{O}^T\right) \quad (6.1)$$

where the denominator $\mathrm{p}\left(\mathbf{Z}^T\right)$ is omitted because it contributes nothing to the maximisation. This is actually a *maximum a posteriori* (MAP) estimator (Bar-Shalom and Fortmann 1988). The first step in this derivation is to determine the probability of the observer states and the probability of the measurements conditioned on the observer, i.e., $\mathrm{p}\left(\mathbf{O}^T\right)$ and $\mathrm{p}\left(\mathbf{Z}^T \,|\, \mathbf{O}^T\right)$.

Commencing with the probability of the observer, and following Streit and Luginbuhl (1995), the conditional probability density functions (pdf's) of the target states for the m^{th} target model are denoted

$$\psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)}\right) \qquad i = 1, 2, \ldots, T \quad (6.2)$$

and the pdf of the assumed *a priori* distribution at time t_0 is,

$$\psi^{(m)}\left(\mathbf{x}_{t_0}^{(m)}\right). \quad (6.3)$$

Then, assuming independence between the target dynamic models, the conditional pdf of all target states at t_i, i.e., for all target models, is written

$$\Psi^{(\mathbf{X})}\left(\mathbf{X}_{t_i} \mid \mathbf{X}_{t_{i-1}}\right) = \prod_{m=1}^{M} \psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)}\right) \qquad i = 1, 2, \ldots, T \quad (6.4)$$

and, assuming independent target initialisations, the pdf of the target states at t_0 is

$$\Psi^{(\mathbf{X})}\left(\mathbf{X}_{t_0}\right) = \prod_{m=1}^{M} \psi^{(m)}\left(\mathbf{x}_{t_0}^{(m)}\right). \quad (6.5)$$

The target state at any time depends on the last known value of the state, i.e., it is a first order Markovian process. This model is particularly suitable for dynamic targets because the target moves to a new position using the current position, velocity, etc. Higher order dynamic states evolve in the same way.

As discussed in section 5.3, the measurements within a measurement scan are uniquely indexed by their sensor number. This differs from the arbitrary measurement numbers

used by Streit and Luginbuhl (1995) for measurements from a single sensor.

The target assignment from sensor s in measurement scan t_i, $k_{t_i}^{(s)}$, is an integer indicating the target from which the measurement $\mathbf{z}_{t_i}^{(s)}$ was received. Let $\pi_{t_i}^{(m)}$ represent the probability of the event that a measurement at time t_i originates from the target m, i.e.,

$$\pi_{t_i}^{(m)} = \mathrm{p}\left(k_{t_i}^{(s)} = m\right) \tag{6.6}$$

for any sensor s. This probability, referred to as the *target measurement probability*, may be viewed as the fraction of measurements at time t_i received from target m or, more generally, the expected fraction of measurements from target m. The fraction of measurements from a particular target is assumed to be the same for all sensors. This is obviously not always the case, e.g., if a target is within the detection range of one sensor and outside the range of another. However, $\pi_{t_i}^{(m)}$ is taken as the expectation of the fraction of measurements from target m over all sensors, although this does not use all the available information.

The collection of these probabilities is denoted

$$\mathbf{\Pi}^T \equiv \left[\boldsymbol{\pi}_{t_1}, \boldsymbol{\pi}_{t_2}, \ldots, \boldsymbol{\pi}_{t_T}\right], \qquad \boldsymbol{\pi}_{t_i} \equiv \left[\pi_{t_i}^{(1)}, \pi_{t_i}^{(2)}, \ldots, \pi_{t_i}^{(M)}\right]^{\mathsf{T}}. \tag{6.7}$$

The parameter $\pi_{t_i}^{(m)}$ defines the distribution of $k_{t_i}^{(s)}$ and, because the measurements are assumed to be conditionally independent and identically distributed, it is the same for all measurements within the scan at time t_i. This assumption allows the probability mass function (pmf) of the scan target assignments $\mathbf{k}_{t_i}$ to be defined as

$$\Psi^{(\mathsf{K})}\left(\mathbf{k}_{t_i}\right) \equiv \mathrm{p}\left(\mathbf{k}_{t_i} = \left[k_{t_i}^{(1)}, k_{t_i}^{(2)}, \ldots, k_{t_i}^{(S)}\right]^{\mathsf{T}}\right) = \prod_{s=1}^{S} \pi_{t_i}^{(m)}\Big|_{m=k_{t_i}^{(s)}}. \tag{6.8}$$

No measurements are available at time t_0, therefore the probability function of the scan observer at this time is simply the pdf of the target states, i.e.,

$$\mathrm{p}\left(\mathbf{O}_{t_0}\right) \equiv \Psi^{(\mathsf{X})}\left(\mathbf{X}_{t_0}\right). \tag{6.9}$$

The target assignments within the scan at t_i are assumed to be independent of the target states at that time and of the assignments at time t_{i-1}. Therefore, as in the PMHT (Streit and Luginbuhl 1995), the conditional probability of the scan observer at time t_i is given as

$$\begin{aligned} \mathrm{p}\left(\mathbf{O}_{t_i} \,|\, \mathbf{O}_{t_{i-1}}\right) &\equiv \mathrm{p}\left(\mathbf{X}_{t_i}, \mathbf{k}_{t_i} \,|\, \mathbf{X}_{t_{i-1}}, \mathbf{k}_{t_{i-1}}\right) \equiv \mathrm{p}\left(\mathbf{X}_{t_i}, \mathbf{k}_{t_i} \,|\, \mathbf{X}_{t_{i-1}}\right) \\ &\equiv \mathrm{p}\left(\mathbf{X}_{t_i} \,|\, \mathbf{X}_{t_{i-1}}\right) \mathrm{p}\left(\mathbf{k}_{t_i}\right) \equiv \Psi^{(\mathsf{X})}\left(\mathbf{X}_{t_i} \,|\, \mathbf{X}_{t_{i-1}}\right) \Psi^{(\mathsf{K})}\left(\mathbf{k}_{t_i}\right). \end{aligned} \tag{6.10}$$

The probability of the entire observer is the joint probability of all the scan observers,

and is written

$$\mathrm{p}\left(\mathbf{O}^T\right) \equiv \mathrm{p}\left(\mathbf{O}_{t_0}, \mathbf{O}_{t_1}, \ldots, \mathbf{O}_{t_T}\right) \equiv \Psi^{(\mathbf{X})}\left(\mathbf{X}_{t_0}\right) \prod_{i=1}^{T} \Psi^{(\mathbf{X})}\left(\mathbf{X}_{t_i} \,|\, \mathbf{X}_{t_{i-1}}\right) \Psi^{(\mathbf{K})}\left(\mathbf{k}_{t_i}\right). \tag{6.11}$$

Now, considering $\mathrm{p}\left(\mathbf{Z}^T \,\middle|\, \mathbf{O}^T\right)$, the conditional measurement pdf is defined for the measurement $\mathbf{z}_{t_i}^{(s)}$ as

$$\mathrm{p}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{O}_{t_i}\right) \equiv \mathrm{p}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{X}_{t_i}, \mathbf{k}_{t_i}\right) = \zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\Big|_{m=k_{t_i}^{(s)}} \tag{6.12}$$

where $\zeta^{(m,s)}$ represents the conditional pdf of the measurement $\mathbf{z}_{t_i}^{(s)}$ from sensor s, given that it originates from target m.

Assuming conditional independence of measurements from different sensors, the conditional pdf for all measurements in the scan at t_i, conditioned on the target states, can be written as

$$\mathrm{p}\left(\mathbf{z}_{t_i} \,|\, \mathbf{O}_{t_i}\right) \equiv \mathrm{p}\left(\mathbf{z}_{t_i} \,|\, \mathbf{X}_{t_i}, \mathbf{k}_{t_i}\right) = \prod_{s=1}^{S} \zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\Big|_{m=k_{t_i}^{(s)}} \tag{6.13}$$

and the pdf for all measurements in the batch as

$$\mathrm{p}\left(\mathbf{Z}^T \,|\, \mathbf{O}^T\right) = \prod_{i=1}^{T} \prod_{s=1}^{S} \zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\Big|_{m=k_{t_i}^{(s)}}. \tag{6.14}$$

This formulation contains $m \times s$ measurement pdf's, one for each possible combination of target and sensor, i.e., each target model has a measurement pdf for each sensor. This differs from that of Streit and Luginbuhl (1995), who only used m measurement pdf's. Their target models each contained only a single measurement pdf, a reasonable assumption given that they only considered measurements that originated from a single sensor.

Recalling (6.11), the joint pdf of the measurements and the batch observer may therefore be written

$$\begin{aligned} \mathrm{p}\left(\mathbf{Z}^T, \mathbf{O}^T\right) &\equiv \mathrm{p}\left(\mathbf{O}^T\right) \mathrm{p}\left(\mathbf{Z}^T \,|\, \mathbf{O}^T\right) \\ &\equiv \Psi^{(\mathbf{X})}\left(\mathbf{X}_{t_0}\right) \prod_{i=1}^{T} \Psi^{(\mathbf{X})}\left(\mathbf{X}_{t_i} \,|\, \mathbf{X}_{t_{i-1}}\right) \Psi^{(\mathbf{K})}\left(\mathbf{k}_{t_i}\right) \mathrm{p}\left(\mathbf{z}_{t_i} \,|\, \mathbf{O}_{t_i}\right). \end{aligned} \tag{6.15}$$

Substituting (6.4), (6.5), (6.8) and (6.14) into (6.15), the joint pdf becomes

$$\mathrm{p}\left(\mathbf{Z}^T, \mathbf{O}^T\right) \equiv \mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T, \mathbf{K}^T\right) = \left\{ \prod_{\nu=1}^{M} \psi^{(\nu)}\left(\mathbf{x}_{t_0}^{(\nu)}\right) \right\}$$

$$\times \prod_{i=1}^{T} \left\{ \left[\prod_{m=1}^{M} \psi^{(m)} \left(\mathbf{x}_{t_i}^{(m)} | \mathbf{x}_{t_{i-1}}^{(m)} \right) \right] \prod_{s=1}^{S} \left[\pi_{t_i}^{(m)} \zeta^{(m,s)} \left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)} \right) \Big|_{m=k_{t_i}^{(s)}} \right] \right\} \quad (6.16)$$

which differs from that of Streit and Luginbuhl (1995) only in the measurement pdf's, i.e., (6.14).

6.2 Development of the msPMHT Algorithm

It is the joint pdf (6.16) that must be maximised (with respect to $\mathbf{X}^T$) to obtain the MAP estimate of the target states. However, without knowing the target assignments, an iterative solution that considers every possible combination of target assignments is required. Such an approach would be computationally impractical for any measurement batch of reasonable length.

The alternative suggested here, as in (Streit and Luginbuhl 1995), is to treat the problem as one of missing data and use the *Expectation-Maximisation* (EM) algorithm (Dempster, Laird and Rubin 1977, Wu 1983, Moon 1996) to develop a solution. In this application, the target assignments $\mathbf{K}^T$ are the missing data, and the complete data comprises the measurements, target states and target assignments, i.e., $\mathbf{Z}^T$, $\mathbf{X}^T$ and $\mathbf{K}^T$. In general, it is assumed that $\mathbf{X}^T$ is the set of parameters defining the target process and measurement pdf's, and the target states are estimated from these parameters. In the linear Gaussian case with known covariances, the target process and measurement pdf's are linear functions of the state means, and it is these state means that are estimated and subsequently used as the state estimates.

The EM algorithm is an iterative algorithm that converges to a maximum (or stationary point in some circumstances) on the joint likelihood surface. Each iteration consists of two steps, an expectation (E) step and a maximisation (M) step. In the E-step, an auxiliary function Q is derived as the expectation of the log likelihood function. Any increase in this function will increase the likelihood function, therefore the solution may be found by maximising the auxiliary function. The M-step involves maximising the auxiliary function with respect to the observer states. Each iteration of the algorithm uses the estimates from the previous iteration as its initial values.

6.2.1 E-Step

The E-step commences by defining the expectation of the log likelihood of the observer (6.16) over all possible values of $\mathbf{K}^T$. This becomes the auxiliary function, given as

$$Q \equiv Q\left(\mathbf{\Pi}^T, \mathbf{X}^T \,|\, \mathbf{\Pi}^{T'}, \mathbf{X}^{T'}\right) = \sum_{\mathbf{K}^T} \left[\log \mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T, \mathbf{K}^T : \mathbf{\Pi}^T\right)\right] \mathrm{p}\left(\mathbf{K}^T \,|\, \mathbf{Z}^T, \mathbf{X}^{T'} : \mathbf{\Pi}^{T'}\right) \quad (6.17)$$

where the dependence on $\mathbf{\Pi}^T$ is made explicit, and the primed variables ($'$) denote initial values, i.e., estimates from the previous iteration.

Using Bayes rule, the second term on the right hand side of (6.17) can be written

$$\mathrm{p}\left(\mathbf{K}^T \,|\, \mathbf{Z}^T, \mathbf{X}^T\right) \equiv \frac{\mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T, \mathbf{K}^T\right)}{\mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T\right)} \tag{6.18}$$

where the denominator represents the marginal distribution of (6.16) over the components $\mathbf{K}^T$. It is defined as

$$\begin{aligned}\mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T\right) \equiv \sum_{\mathbf{K}^T} \mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T, \mathbf{K}^T\right) = \left\{\prod_{\nu=1}^{M} \psi^{(\nu)}\left(\mathbf{x}_{t_0}^{(\nu)}\right)\right\} \\ \times \prod_{i=1}^{T}\left\{\left[\prod_{m=1}^{M} \psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)}\right)\right] \prod_{s=1}^{S}\left[\sum_{\rho=1}^{M} \pi_{t_i}^{(\rho)} \zeta^{(\rho,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(\rho)}\right)\right]\right\}\end{aligned} \tag{6.19}$$

where the summation over $\mathbf{K}^T$ is defined as

$$\sum_{\mathbf{K}^T} \equiv \sum_{k_{t_1}^{(1)}=1}^{M} \cdots \sum_{k_{t_1}^{(S)}=1}^{M} \sum_{k_{t_2}^{(1)}=1}^{M} \cdots \sum_{k_{t_{T-1}}^{(S)}=1}^{M} \sum_{k_{t_T}^{(1)}=1}^{M} \cdots \sum_{k_{t_T}^{(S)}=1}^{M}. \tag{6.20}$$

Equation (6.19) illustrates the concept of representing each measurement as a finite mixture of conditional measurement pdf's, where each pdf is conditioned on a different target model, and the measurement probabilities $\mathbf{\Pi}^T$ act as the mixing parameters or proportions, i.e.,

$$\sum_{\rho=1}^{M} \pi_{t_i}^{(\rho)} \zeta^{(\rho,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(\rho)}\right).$$

Substituting (6.16) and (6.19) into (6.18) gives

$$\mathrm{p}\left(\mathbf{K}^T \,|\, \mathbf{Z}^T, \mathbf{X}^T\right) = \prod_{i=1}^{T} \prod_{s=1}^{S} \omega_{t_i}^{(m,s)}\Big|_{m=k_{t_i}^{(s)}} \tag{6.21}$$

where the *target assignment probability*, $\omega_{t_i}^{(m,s)}$, represents the probability that the measurement from sensor s in scan t_i is assigned to target m. It is defined as

$$\omega_{t_i}^{(m,s)} = \frac{\pi_{t_i}^{(m)} \zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)}{\sum_{\rho=1}^{M} \pi_{t_i}^{(\rho)} \zeta^{(\rho,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(\rho)}\right)}. \tag{6.22}$$

The first term of the right hand side of (6.17) is obtained by taking the logarithm of

(6.16), i.e.,

$$\begin{aligned}\log \mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T, \mathbf{K}^T\right) = &\sum_{\nu=1}^{M} \log \psi^{(\nu)}\left(\mathbf{x}_{t_0}^{(\nu)}\right) + \sum_{i=1}^{T}\sum_{m=1}^{M} \log \psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)}\right) \\ &+ \sum_{i=1}^{T}\sum_{s=1}^{S}\left[\log \pi_{t_i}^{(m)} + \log \zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\right]\Big|_{m=k_{t_i}^{(s)}}.\end{aligned} \tag{6.23}$$

Let $\sum_{\mathbf{K}^T|k_{t_i}^{(s)}}$ represent the summation over all target assignments except one, i.e., the target assignment for the measurement from sensor s in scan t_i is assumed to be known, and define it as

$$\sum_{\mathbf{K}^T|k_{t_i}^{(s)}} \equiv \sum_{k_{t_1}^{(1)}=1}^{M} \cdots \sum_{k_{t_1}^{(S)}=1}^{M} \sum_{k_{t_2}^{(1)}=1}^{M} \cdots \sum_{k_{t_i}^{(s-1)}=1}^{M} \sum_{k_{t_i}^{(s+1)}=1}^{M} \cdots \sum_{k_{t_{T-1}}^{(S)}=1}^{M} \sum_{k_{t_T}^{(1)}=1}^{M} \cdots \sum_{k_{t_T}^{(S)}=1}^{M}. \tag{6.24}$$

Then substituting (6.23) and (6.21) into (6.17), rearranging the summation order

$$\sum_{\mathbf{K}^T}\sum_{i=1}^{T}\sum_{r=1}^{n_{t_i}} \equiv \sum_{i=1}^{T}\sum_{r=1}^{n_{t_i}}\sum_{k_{t_i}^{(s)}=1}^{M}\sum_{\mathbf{K}^T|k_{t_i}^{(s)}} \tag{6.25}$$

and using the identity

$$\sum_{\mathbf{K}^T|k_{t_i}^{(r)}} \mathrm{p}\left(\mathbf{K}^T \,\middle|\, \mathbf{Z}^T, \mathbf{X}^T\right) = \omega_{t_i}^{(m,r)}\Big|_{m=k_{t_i}^{(r)}} \tag{6.26}$$

that was introduced by Streit and Luginbuhl (1995), the auxiliary function becomes

$$\begin{aligned}Q = &\sum_{\nu=1}^{M} \log \psi^{(\nu)}\left(\mathbf{x}_{t_0}^{(\nu)}\right) + \sum_{i=1}^{T}\sum_{m=1}^{M} \log \psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)}\right) \\ &+ \sum_{i=1}^{T}\sum_{s=1}^{S}\sum_{m=1}^{M}\left[\log \pi_{t_i}^{(m)} + \log \zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\right]\omega_{t_i}^{(m,s)'}.\end{aligned} \tag{6.27}$$

$\omega_{t_i}^{(m,s)'}$ denotes the target assignment probabilities calculated using $\mathbf{X}^{T'}$ and $\mathbf{\Pi}^{T'}$, i.e., calculated from previous estimates.

The function Q can be broken into $T + M$ independent sub-functions that may be maximised individually by separating the variables, i.e.,

$$Q = \sum_{i=1}^{T} Q_{t_i}^{(\mathbf{\Pi})} + \sum_{m=1}^{M} Q_m^{(\mathbf{X})} \tag{6.28}$$

where

$$Q_{t_i}^{(\Pi)} = \sum_{m=1}^{M} \left(\log \pi_{t_i}^{(m)} \right) \sum_{s=1}^{S} \omega_{t_i}^{(m,s)'} \qquad i = 1, 2, \ldots, T \tag{6.29}$$

$$\begin{aligned} Q_m^{(\mathbf{X})} &= \log \psi^{(m)} \left(\mathbf{x}_{t_0}^{(m)} \right) + \sum_{i=1}^{T} \log \psi^{(m)} \left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)} \right) \\ &\quad + \sum_{i=1}^{T} \sum_{s=1}^{S} \omega_{t_i}^{(m,s)'} \log \zeta^{(m,s)} \left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)} \right) \qquad m = 1, 2, \ldots, M. \end{aligned} \tag{6.30}$$

Again it is only the measurement pdf's that differ from the results obtained by Streit and Luginbuhl (1995).

6.2.2 M-Step

In this step, the parameters of the probability distributions for the target assignments and target states are estimated from the auxiliary function (6.28). This has been decoupled into a maximisation problem for each of the T target measurement probability vectors $\boldsymbol{\pi}_{t_i}$ and the M target state sequences $\mathbf{X}^{(m)}$, and each of these will be dealt with individually, i.e., each equation of (6.29) and (6.30) will be maximised separately.

6.2.2.1 Target Measurement Probabilities

Consider the decoupled maximisation problem for the target assignments in a single measurement scan, $\boldsymbol{\pi}_{t_i}$. These assignment probabilities in the scan at t_i must sum to unity, as each represents a fraction of the measurements and the total fraction must be one. The maximisation of $Q_{t_i}^{(\Pi)}$ (6.29) with respect to $\boldsymbol{\pi}_{t_i}$ is therefore subject to the constraint

$$\sum_{m=1}^{M} \pi_{t_i}^{(m)} = 1. \tag{6.31}$$

This constraint may be incorporated into the maximisation by formulating the dual problem (section 2.2.9.2), in which the constraint is introduced into $Q_{t_i}^{(\Pi)}$ (6.29) through the Lagrangian multiplier γ_{t_i}. The problem then becomes one of maximising the unconstrained function

$$\mathcal{L}_{t_i}^{(\Pi)} = \sum_{m=1}^{M} \left(\log \pi_{t_i}^{(m)} \right) \sum_{s=1}^{S} \omega_{t_i}^{(m,s)'} + \gamma_{t_i} \left(1 - \sum_{m=1}^{M} \pi_{t_i}^{(m)} \right). \tag{6.32}$$

Maximising this with respect to $\pi_{t_i}^{(m)}$ gives

$$\hat{\pi}_{t_i}^{(m)} = \frac{1}{\gamma_{t_i}} \sum_{s=1}^{S} \omega_{t_i}^{(m,s)'} \tag{6.33}$$

where $\hat{\pi}_{t_i}^{(m)}$ is the estimate of the measurement probability for target m at time t_i. Substituting (6.33) into the constraint (6.31) and solving for γ_{t_i} gives $\gamma_{t_i} = S$ and hence

$$\hat{\pi}_{t_i}^{(m)} = \frac{1}{S} \sum_{s=1}^{S} \omega_{t_i}^{(m,s)'} \tag{6.34}$$

which corresponds to that obtained by Streit and Luginbuhl (1995). This procedure is repeated for each of the T measurement scans.

6.2.2.2 Target State Sequences

The target state sequence for each model m is estimated by maximising $Q_m^{(\mathbf{X})}$ (6.30) with respect to $\mathbf{X}^{(m)}$. This produces a set of $T+1$ simultaneous equations for each target model m, and the solution of this set gives the unknown parameters defining the pdf of the target states for that target. The target states are estimated from these parameters.

An alternative is to take the exponential of $Q_m^{(\mathbf{X})}$ (Streit and Luginbuhl 1995). This gives

$$\begin{aligned} \exp\left(Q_m^{(\mathbf{X})}\right) &= \psi^{(m)}\left(\mathbf{x}_{t_0}^{(m)}\right) \prod_{i=1}^{T} \left[\psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)}\right) \prod_{s=1}^{S} \left(\zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\right)^{\omega_{t_i}^{(m,s)'}} \right] \\ &\propto \psi^{(m)}\left(\mathbf{x}_{t_0}^{(m)}\right) \prod_{i=1}^{T} \left[\psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)}\right) \Lambda_{t_i}^{(m)}\left(\mathbf{z}_{t_i} \,|\, \mathbf{x}_{t_i}^{(m)}\right)\right] \end{aligned} \tag{6.35}$$

where $\Lambda_{t_i}^{(m)}$ is the conditional measurement pdf at measurement time t_i, defined as

$$\Lambda_{t_i}^{(m)}\left(\mathbf{z}_{t_i} \,|\, \mathbf{x}_{t_i}^{(m)}\right) = c \prod_{s=1}^{S} \left(\zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\right)^{\omega_{t_i}^{(m,s)'}} \tag{6.36}$$

and c is a normalisation constant. The required estimate $\hat{\mathbf{X}}^{(m)}$ is the value of $\mathbf{X}^{(m)}$ that maximises (6.35), where $\mathbf{X}^{(m)}$ represents the parameters describing the pdf of the target states. In general, an iterative algorithm is employed to solve this problem, and such an algorithm is conceptually equivalent to a single target maximum a posteriori (MAP) tracker. In the special case of a linear Gaussian system, this is equivalent to the fixed interval *Kalman smoother* (Gelb 1992).

6.2.2.3 Linear Gaussian Case

For the special case of a linear Gaussian system, the estimated parameters are the target state means. As the means are the maximum likelihood estimates of the target states for Gaussian distributions, $\hat{\mathbf{X}}^{(m)}$ is in fact the estimates of the target states. Therefore the target state sequences can be obtained directly for linear Gaussian systems.

The linear Gaussian target state and measurement pdf's for the target model m are

defined as

$$\begin{aligned}
\psi^{(m)}\left(\mathbf{x}_{t_0}^{(m)}\right) &\sim \mathcal{N}\left(\bar{\mathbf{x}}_{t_0}^{(m)}, \bar{\mathbf{\Sigma}}_{t_0}^{(m)}\right) \\
\psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \middle| \mathbf{x}_{t_{i-1}}^{(m)}\right) &\sim \mathcal{N}\left(\mathbf{F}_{t_{i-1}}^{(m)}\mathbf{x}_{t_{i-1}}^{(m)}, \mathbf{Q}_{t_i}^{(m)}\right) \\
\zeta^{(m,s)}\left(\mathbf{z}_{t_i}^{(s)} \middle| \mathbf{x}_{t_i}^{(m)}\right) &\sim \mathcal{N}\left(\mathbf{H}_{t_i}^{(m,s)}\mathbf{x}_{t_i}^{(m)}, \mathbf{R}_{t_i}^{(m,s)}\right)
\end{aligned} \tag{6.37}$$

where $\mathcal{N}(\mu, \Sigma)$ denotes the normal or Gaussian distribution with mean μ and covariance Σ.

Using the linear Gaussian models introduced in section 5.3.1.1, the above pdf's are substituted into (6.30) and, maximising with respect to each $\mathbf{x}_{t_i}^{(m)}$, $i = 1, 2, \ldots, T$, the following symmetric tridiagonal system of equations is obtained for each target model.

$$\begin{aligned}
&\left(\bar{\mathbf{\Sigma}}_{t_0}^{(m)-1} + \mathbf{D}_{t_0}^{(m)}\right)\mathbf{x}_{t_0}^{(m)} - \mathbf{B}_{t_0}^{(m)}\mathbf{x}_{t_1}^{(m)} = \bar{\mathbf{\Sigma}}_{t_0}^{(m)-1}\bar{\mathbf{x}}_{t_0}^{(m)} && (i=0) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
-\mathbf{B}_{t_{i-1}}^{(m)\mathsf{T}}\mathbf{x}_{t_{i-1}}^{(m)} &+ \left(\mathbf{A}_{t_i}^{(m)} + \mathbf{D}_{t_i}^{(m)}\right)\mathbf{x}_{t_i}^{(m)} - \mathbf{B}_{t_i}^{(m)}\mathbf{x}_{t_{i+1}}^{(m)} = \mathbf{c}_{t_i}^{(m)} && i = 1, 2, \ldots, T-1 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
-\mathbf{B}_{t_{T-1}}^{(m)\mathsf{T}}\mathbf{x}_{t_{T-1}}^{(m)} &+ \mathbf{A}_{t_T}^{(m)}\mathbf{x}_{t_T}^{(m)} = \mathbf{c}_{t_T}^{(m)} && (i=T)
\end{aligned} \tag{6.38}$$

where

$$\begin{aligned}
\mathbf{A}_{t_i}^{(m)} &= \mathbf{Q}_{t_{i-1}}^{(m)-1} + \sum_{s=1}^{S} \omega_{t_i}^{(m,s)\prime}\mathbf{H}_{t_i}^{(m,s)\mathsf{T}}\mathbf{R}_{t_i}^{(m,s)-1}\mathbf{H}_{t_i}^{(m,s)} && i = 1, 2, \ldots, T \\
\mathbf{B}_{t_i}^{(m)} &= \mathbf{F}_{t_i}^{(m)\mathsf{T}}\mathbf{Q}_{t_i}^{(m)-1} && i = 0, 1, \ldots, T-1 \\
\mathbf{c}_{t_i}^{(m)} &= \sum_{s=1}^{S} \omega_{t_i}^{(m,s)\prime}\mathbf{H}_{t_i}^{(m,s)\mathsf{T}}\mathbf{R}_{t_i}^{(m,s)-1}\mathbf{z}_{t_i}^{(s)} && i = 1, 2, \ldots, T \\
\mathbf{D}_{t_i}^{(m)} &= \mathbf{F}_{t_i}^{(m)\mathsf{T}}\mathbf{Q}_{t_i}^{(m)-1}\mathbf{F}_{t_i}^{(m)} && i = 0, 1, \ldots, T-1
\end{aligned} \tag{6.39}$$

This result differs from that of Streit and Luginbuhl (1995) in that they did not explicitly weight each measurement by the inverse of its covariance because, being from the same sensor, all their measurements had the same covariance. The remainder of this section introduces a composite measurement model for measurements from dissimilar sensors (Krieg and Gray 1996a), a problem not addressed by the PMHT algorithm.

For each target model, a single composite (or synthetic) measurement model for each target[1] is introduced as

$$\tilde{\mathbf{z}}_{t_i}^{(m)} = \tilde{\mathbf{H}}_{t_i}^{(m)}\mathbf{x}_{t_i}^{(m)} + \tilde{\mathbf{v}}_{t_i}^{(m)} \qquad m = 1, 2, \ldots, M, \quad i = 1, 2, \ldots, T \tag{6.40}$$

[1]This represents the single measurement model in the formulation of Streit and Luginbuhl (1995).

where $\tilde{\mathbf{v}}_{t_i}^{(m)}$ denotes the composite zero mean Gaussian measurement noise with covariance $\tilde{\mathbf{R}}_{t_i}^{(m)}$.

The composite measurement matrix $\tilde{\mathbf{H}}_{t_i}^{(m)}$ maps the state space of the target model onto a composite measurement space that contains all possible measurement types from all sensors. It is determined by forming the matrix $\left[\mathbf{H}_{t_i}^{(m,1)\mathsf{T}}, \mathbf{H}_{t_i}^{(m,2)\mathsf{T}}, \cdots, \mathbf{H}_{t_i}^{(m,S)\mathsf{T}}\right]^{\mathsf{T}}$ and, using Gauss-Jordan or similar methods, simplifying it into reduced row echelon form. The zero rows of this simplified matrix are removed, leaving the composite measurement matrix $\tilde{\mathbf{H}}_{t_i}^{(m)}$. The linearly independent rows of this matrix each contain a single one; all other elements are zero. Therefore every (independent) type of measurement will appear once in the composite model.

Introducing the relationship $\mathbf{H}_{t_i}^{(m,s)} = \bar{\mathbf{H}}_{t_i}^{(m,s)}\tilde{\mathbf{H}}_{t_i}^{(m)}$ for all $s = 1, 2, \ldots, S$, the *measurement transformation matrix* for sensor s and target m, $\bar{\mathbf{H}}_{t_i}^{(m,s)}$, is defined as

$$\bar{\mathbf{H}}_{t_i}^{(m,s)} = \mathbf{H}_{t_i}^{(m,s)}\tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}}\left(\tilde{\mathbf{H}}_{t_i}^{(m)}\tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}}\right)^{-1}. \tag{6.41}$$

Given the aforementioned structure of $\tilde{\mathbf{H}}_{t_i}^{(m)}$, the inverse in the above equation not only exists, but it is always the identity matrix. Therefore (6.41) is simply

$$\bar{\mathbf{H}}_{t_i}^{(m,s)} = \mathbf{H}_{t_i}^{(m,s)}\tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}}. \tag{6.42}$$

$\mathbf{A}_{t_i}^{(m)}$ and $\mathbf{c}_{t_i}^{(m)}$ in (6.39) may now be re-written as

$$\begin{aligned}\mathbf{A}_{t_i}^{(m)} &= \mathbf{Q}_{t_{i-1}}^{(m)-1} + \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}}\tilde{\mathbf{R}}_{t_i}^{(m)-1}\tilde{\mathbf{H}}_{t_i}^{(m)} \\ \mathbf{c}_{t_i}^{(m)} &= \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}}\tilde{\mathbf{R}}_{t_i}^{(m)-1}\tilde{\mathbf{z}}_{t_i}^{(m)}\end{aligned} \tag{6.43}$$

where the composite measurement $\tilde{\mathbf{z}}_{t_i}^{(m)}$, and its covariance $\tilde{\mathbf{R}}_{t_i}^{(m)}$, are defined as

$$\tilde{\mathbf{z}}_{t_i}^{(m)} = \tilde{\mathbf{R}}_{t_i}^{(m)}\sum_{s=1}^{S}\omega_{t_i}^{(m,s)'}\bar{\mathbf{H}}_{t_i}^{(m,s)\mathsf{T}}\mathbf{R}_{t_i}^{(m,s)-1}\mathbf{z}_{t_i}^{(s)} \tag{6.44}$$

$$\tilde{\mathbf{R}}_{t_i}^{(m)} = \left(\sum_{s=1}^{S}\omega_{t_i}^{(m,s)'}\bar{\mathbf{H}}_{t_i}^{(m,s)\mathsf{T}}\mathbf{R}_{t_i}^{(m,s)-1}\bar{\mathbf{H}}_{t_i}^{(m,s)}\right)^{-1}. \tag{6.45}$$

By way of example, consider the radar and optical measurements in section 3.3.3 and, in particular, the measurement and covariance matrices in (3.11) and (3.12). The composite measurement matrix for this example is

$$\tilde{\mathbf{H}}_{t_i}^{(m)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \tag{6.46}$$

i.e., the same as $\mathbf{H}_{t_i}^{(m,1)}$, and from (6.42)

$$\bar{\mathbf{H}}_{t_i}^{(m,1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \bar{\mathbf{H}}_{t_i}^{(m,2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{6.47}$$

Then, denoting the composite measurements by $\tilde{}$, the composite measurement vector is

$$\tilde{\mathbf{z}}_{t_i}^{(m)} \equiv \begin{bmatrix} \tilde{\eta}_{t_i}^{(m)} \\ \tilde{\varepsilon}_{t_i}^{(m)} \\ \tilde{R}_{t_i}^{(m)} \\ \tilde{\dot{R}}_{t_i}^{(m)} \end{bmatrix} = \tilde{\mathbf{R}}_{t_i}^{(m)} \begin{bmatrix} \omega_{t_i}^{(m,1)'} \sigma_\eta^{(1)\,-2} \eta_{t_i}^{(1)} + \omega_{t_i}^{(m,2)'} \sigma_\eta^{(2)\,-2} \eta_{t_i}^{(2)} \\ \omega_{t_i}^{(m,1)'} \sigma_\varepsilon^{(1)\,-2} \varepsilon_{t_i}^{(1)} + \omega_{t_i}^{(m,2)'} \sigma_\varepsilon^{(2)\,-2} \varepsilon_{t_i}^{(2)} \\ \omega_{t_i}^{(m,1)'} \sigma_R^{(1)\,-2} R_{t_i}^{(1)} \\ \omega_{t_i}^{(m,1)'} \sigma_{\dot{R}}^{(1)\,-2} \dot{R}_{t_i}^{(1)} \end{bmatrix} \tag{6.48}$$

and its covariance $\tilde{\mathbf{R}}_{t_i}^{(m)}$ is a diagonal matrix, where the elements on the primary diagonal are

$$\begin{bmatrix} \left(\omega_{t_i}^{(m,1)'} \sigma_\eta^{(1)\,-2} + \omega_{t_i}^{(m,2)'} \sigma_\eta^{(2)\,-2}\right)^{-1} \\ \left(\omega_{t_i}^{(m,1)'} \sigma_\varepsilon^{(1)\,-2} + \omega_{t_i}^{(m,2)'} \sigma_\varepsilon^{(2)\,-2}\right)^{-1} \\ \left(\omega_{t_i}^{(m,1)'} \sigma_R^{(1)\,-2}\right)^{-1} \\ \left(\omega_{t_i}^{(m,1)'} \sigma_{\dot{R}}^{(1)\,-2}\right)^{-1} \end{bmatrix}. \tag{6.49}$$

This is similar to the data compression mentioned by Willner et al. (1976), i.e., for a single dimension measurement, $\mathbf{z}_{t_i} = \mathbf{R}_{t_i}\left(\sigma^{(1)\,-2}\mathbf{z}_{t_i}^{(1)} + \sigma^{(2)\,-2}\mathbf{z}_{t_i}^{(2)}\right)$ and $\mathbf{R}_{t_i} = \left(\sigma^{(1)\,-2} + \sigma^{(2)\,-2}\right)^{-1}$.

The tri-diagonal system of equations of each target model (6.38) may now be solved for the unknown target states using a fixed interval Kalman smoother (figure 6.1) with the appropriate composite measurement model (*see* Appendix B).

This completes the M-step.

6.3 Linear Gaussian msPMHT in Iterative Form

The msPMHT algorithm is based on the EM algorithm and is therefore iterative. This section completes the derivation of the msPMHT algorithm by presenting the linear Gaussian version in its iterative form.

Assume that the batch measurement $\mathbf{Z}^T$ is available, and initial values for the measurement to target probabilities ($\mathbf{\Pi}^{T'}$) and the target state sequences ($\mathbf{X}^{T'}$) have been chosen. The $'$ denotes the initial values for the first iteration and the results obtained from the previous iteration otherwise. The algorithm is stopped when the likelihood function has

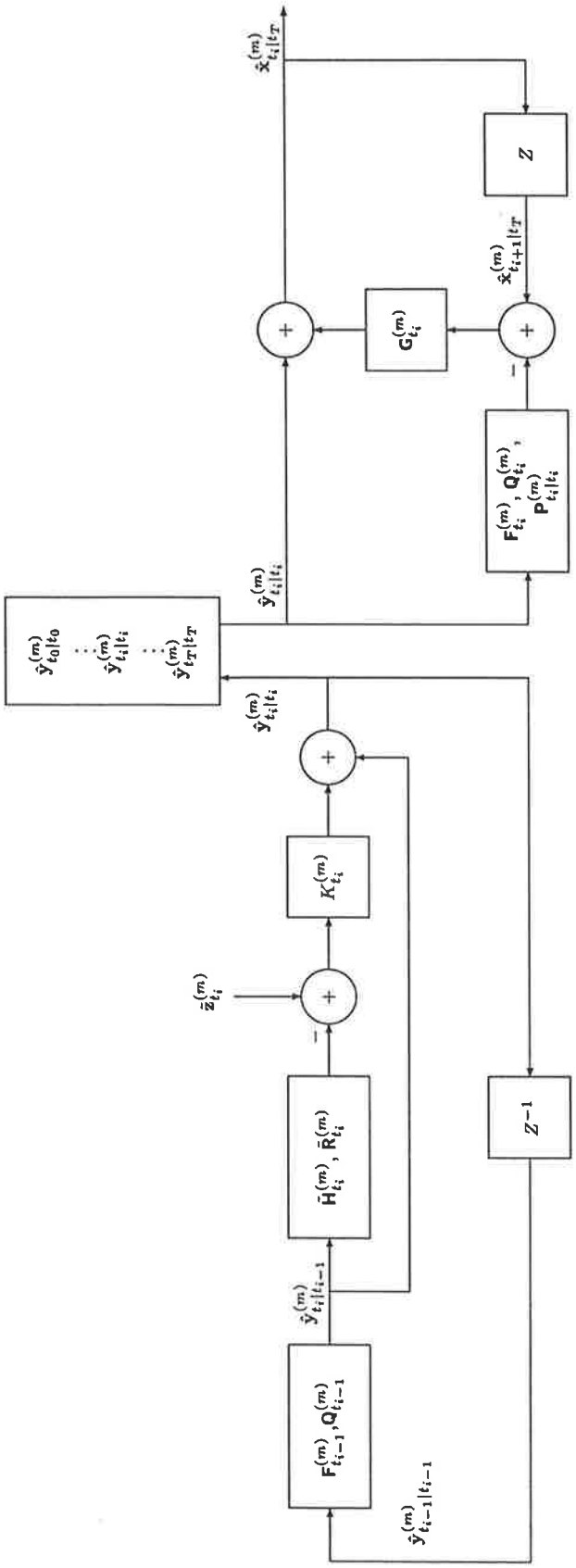


Figure 6.1: Block diagram of the fixed interval Kalman smoother

converged to a constant value.

First the target assignment probabilities

$$\omega_{t_i}^{(m,s)} = \frac{\pi_{t_i}^{(m)'} \mathcal{N}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{H}_{t_i}^{(m,s)} \mathbf{x}_{t_i}^{(m)'}, \mathbf{R}_{t_i}^{(m,s)}\right)}{\sum_{\rho=1}^{M} \pi_{t_i}^{(\rho)'} \mathcal{N}\left(\mathbf{z}_{t_i}^{(s)} \,\middle|\, \mathbf{H}_{t_i}^{(\rho,s)} \mathbf{x}_{t_i}^{(\rho)'}, \mathbf{R}_{t_i}^{(\rho,s)}\right)} \tag{6.50}$$

are computed for $i = 1, 2, \ldots, T$, $m = 1, 2, \ldots, M$ and $s = 1, 2, \ldots, S$. Next the target measurement probabilities

$$\pi_{t_i}^{(m)} = \frac{1}{S} \sum_{s=1}^{S} \omega_{t_i}^{(m,s)} \tag{6.51}$$

are updated for all m and i. The composite measurements and their covariances are computed for each target model using (6.45) and (6.44), and these are used in a fixed interval Kalman smoother to obtain the state estimates.

The fixed interval Kalman smoother is implemented for each target $m = 1, 2, \ldots, M$ as follows. The intermediate state variables $\hat{\mathbf{y}}_{t_i|t_j}^{(m)}$ and their covariances $\mathbf{P}_{t_i|t_j}^{(m)}$ are introduced to represent the filtered estimates obtained from the forward recursion of the smoother. They are initialised as

$$\begin{aligned} \hat{\mathbf{y}}_{t_0|t_0}^{(m)} &= \bar{\mathbf{x}}_{t_0}^{(m)} \\ \mathbf{P}_{t_0|t_0}^{(m)} &= \bar{\Sigma}_{t_0}^{(m)} \end{aligned} \tag{6.52}$$

and the forward recursion, or filtering, proceeds for $i = 1, 2, \ldots, T$ as

$$\begin{aligned} \mathbf{P}_{t_i|t_{i-1}}^{(m)} &= \mathbf{F}_{t_{i-1}}^{(m)} \mathbf{P}_{t_{i-1}|t_{i-1}}^{(m)} \mathbf{F}_{t_{i-1}}^{(m)\mathsf{T}} + \mathbf{Q}_{t_{i-1}}^{(m)} \\ \mathbf{W}_{t_i}^{(m)} &= \mathbf{P}_{t_i|t_{i-1}}^{(m)} \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} \left(\tilde{\mathbf{H}}_{t_i}^{(m)} \mathbf{P}_{t_i|t_{i-1}}^{(m)} \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} + \tilde{\mathbf{R}}_{t_i}^{(m)}\right)^{-1} \\ \mathbf{P}_{t_i|t_i}^{(m)} &= \left(\mathbf{1} - \mathbf{W}_{t_i}^{(m)} \tilde{\mathbf{H}}_{t_i}^{(m)}\right) \mathbf{P}_{t_i|t_{i-1}}^{(m)} \\ \hat{\mathbf{y}}_{t_i|t_i}^{(m)} &= \mathbf{F}_{t_{i-1}}^{(m)} \hat{\mathbf{y}}_{t_{i-1}|t_{i-1}}^{(m)} + \mathbf{W}_{t_i}^{(m)} \left(\tilde{\mathbf{z}}_{t_i}^{(m)} - \tilde{\mathbf{H}}_{t_i}^{(m)} \mathbf{F}_{t_{i-1}}^{(m)} \hat{\mathbf{y}}_{t_{i-1}|t_{i-1}}^{(m)}\right). \end{aligned} \tag{6.53}$$

The backward recursion, or smoothing, commences with $\mathbf{x}_{t_T}^{(m)} = \hat{\mathbf{y}}_{t_T|t_T}^{(m)}$ and $\Sigma_{t_T}^{(m)} = \mathbf{P}_{t_T|t_T}^{(m)}$, and it continues for $i = (T-1), \ldots, 1, 0$ as

$$\begin{aligned} \mathbf{x}_{t_i}^{(m)} &= \hat{\mathbf{y}}_{t_i|t_i}^{(m)} + \mathbf{P}_{t_i|t_i}^{(m)} \mathbf{F}_{t_i}^{(m)\mathsf{T}} \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \left(\mathbf{x}_{t_{i+1}}^{(m)} - \mathbf{F}_{t_i}^{(m)} \hat{\mathbf{y}}_{t_i|t_i}^{(m)}\right) \\ \Sigma_{t_i}^{(m)} &= \mathbf{P}_{t_i|t_i}^{(m)} + \mathbf{P}_{t_i|t_i}^{(m)} \mathbf{F}_{t_i}^{(m)\mathsf{T}} \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \left(\Sigma_{t_{i+1}}^{(m)} - \mathbf{P}_{t_{i+1}|t_i}^{(m)}\right) \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \mathbf{F}_{t_i}^{(m)} \mathbf{P}_{t_i|t_i}^{(m)} \end{aligned} \tag{6.54}$$

where $\Sigma_{t_i}^{(m)}$ denotes the state estimate error covariance for target model m at time t_i. The state estimate error covariance is included for completeness only, it is not required for estimating the target states.

This algorithm is summarised in figure 6.2.

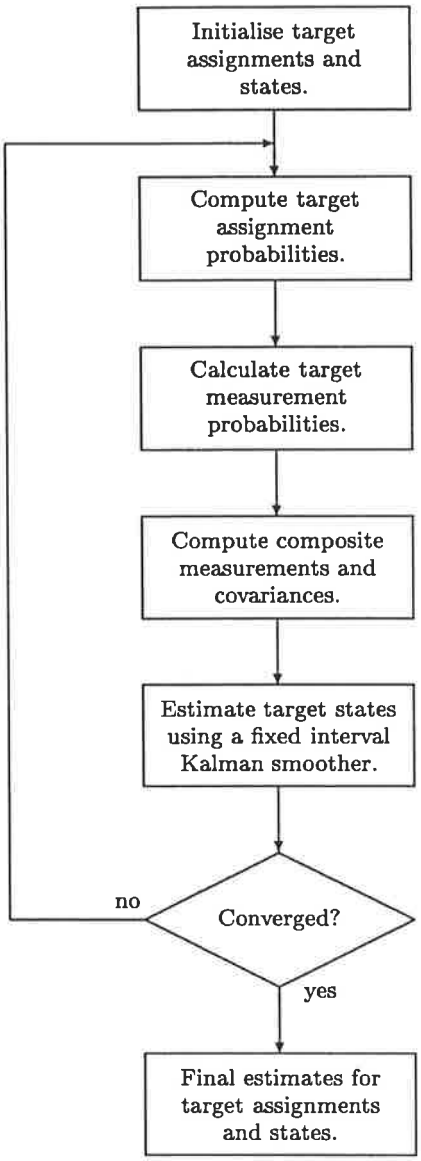


Figure 6.2: Block diagram of the iterative linear Gaussian msPMHT algorithm

6.4 Asynchronous Sensors

The restriction of a single measurement from each sensor in each measurement scan can be lifted, allowing any number of measurements, including none, from each sensor in each measurement scan.

As mentioned in the asynchronous problem formulation in section 5.4, the main change is in the indexing of measurements and target assignments within each scan. Therefore the previous development of the msPMHT algorithm for simultaneous measurements also holds for the asynchronous version of the algorithm, provided the appropriate indexing changes outlined in section 5.4 are applied. Given this similarity, only the results of the

asynchronous algorithm under linear Gaussian assumptions are presented. These results are easily translated for more general non-linear and/or non-Gaussian formulations.

The target assignment probabilities and the measurement probabilities for the asynchronous version of the msPMHT are given as

$$\omega_{t_i}^{(m,r)} = \frac{\pi_{t_i}^{(m)'} \mathcal{N}\left(\mathbf{z}_{t_i}^{(r)} \middle| \mathbf{H}_{t_i}^{(m,s)} \mathbf{x}_{t_i}^{(m)'}, \mathbf{R}_{t_i}^{(m,s)}\right)}{\sum_{\rho=1}^{M} \pi_{t_i}^{(\rho)'} \mathcal{N}\left(\mathbf{z}_{t_i}^{(r)} \middle| \mathbf{H}_{t_i}^{(\rho,s)} \mathbf{x}_{t_i}^{(\rho)'}, \mathbf{R}_{t_i}^{(\rho,s)}\right)} \tag{6.55}$$

$$\pi_{t_i}^{(m)} = \frac{1}{n_{t_i}} \sum_{r=1}^{n_{t_i}} \omega_{t_i}^{(m,r)} \tag{6.56}$$

where the changes have been in the indexing of the measurements and target assignment probabilities.

For a linear Gaussian system, the states of each target are estimated using a fixed interval Kalman smoother with the composite measurements and covariances

$$\tilde{\mathbf{z}}_{t_i}^{(m)} = \tilde{\mathbf{R}}_{t_i}^{(m)} \sum_{r=1}^{n_{t_i}} \omega_{t_i}^{(m,r)'} \bar{\mathbf{H}}_{t_i}^{(m,s)\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \mathbf{z}_{t_i}^{(r)} \tag{6.57}$$

$$\tilde{\mathbf{R}}_{t_i}^{(m)} = \left(\sum_{r=1}^{n_{t_i}} \omega_{t_i}^{(m,r)'} \bar{\mathbf{H}}_{t_i}^{(m,s)\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \bar{\mathbf{H}}_{t_i}^{(m,s)} \right)^{-1} \tag{6.58}$$

where again the difference between the synchronous and asynchronous msPMHT algorithms is in the measurement indexing. Note that the sensor providing each measurement is known and determines the parameters of the measurement model.

Using the equations (6.55), (6.56), (6.57) and (6.58), the iterative linear Gaussian asynchronous msPMHT is as illustrated in figure 6.2.

6.5 General msPMHT

The *general multi-sensor Probabilistic Multi-Hypothesis Tracking* algorithm (Krieg and Gray 1996a) provides multiple sensor models for each physical sensor, allowing the algorithm to adapt to changing sensor operating conditions. The missing data now consists of target assignments, as for the msPMHT, and measurement to sensor assignments. These sensor assignments are estimated along with the target assignments and target states.

The development of this algorithm is a generalisation of the msPMHT algorithm presented earlier in this chapter. It follows from the the general multi-sensor multi-target problem definition in section 5.5.

6.5.1 General Observer Likelihood Structure

As for the msPMHT, the observer states are estimated from the observer likelihood or probability (6.1). However, the introduction of the sensor assignments (5.19) has changed the observer structure (5.18) from that used in the msPMHT (5.6).

The probability mass functions of the sensor assignments are required to complete the observer likelihood function. Let $\xi_{t_i}^{(p)}$ denote the probability that a measurement in the scan at time t_i is produced by the sensor model p, i.e.,

$$\xi_{t_i}^{(p)} = \mathrm{p}\left(l_{t_i}^{(r)} = p\right) \tag{6.59}$$

for all measurements from the same sensor set (physical sensor) in the scan at time t_i. This probability is referred to as the *sensor measurement probability*, and it is the expectation of the fraction of the total measurements from a sensor set at t_i that are produced by a particular sensor model p from that sensor set. Therefore the collection of sensor measurement probabilities over all batch times is denoted

$$\Xi^T \equiv \left[\boldsymbol{\xi}_{t_1}, \boldsymbol{\xi}_{t_2}, \ldots, \boldsymbol{\xi}_{t_T}\right], \qquad \boldsymbol{\xi}_{t_i} \equiv \left[\xi_{t_i}^{(1)}, \xi_{t_i}^{(2)}, \ldots, \xi_{t_i}^{(S)}\right]^{\mathsf{T}}. \tag{6.60}$$

The parameter $\xi_{t_i}^{(p)}$ defines the distribution of $l_{t_i}^{(r)}$ and, because the sensor assignments are assumed to be independent within measurement scans, the pmf's of the sensor assignments in the scan at time t_i are defined as

$$\Psi^{(\mathsf{L})}(\mathbf{l}_{t_i}) \equiv \mathrm{p}\left(\mathbf{l}_{t_i} = \left[l_{t_i}^{(1)}, l_{t_i}^{(2)}, \ldots, l_{t_i}^{(n_{t_i})}\right]^{\mathsf{T}}\right) = \prod_{r=1}^{n_{t_i}} \xi_{t_i}^{(p)}\Big|_{p=l_{t_i}^{(r)}}. \tag{6.61}$$

Assuming the sensor assignments are statistically independent across measurement scans, and the target states ($\mathbf{X}^T$), the target assignments ($\mathbf{K}^T$) and the sensor assignments ($\mathbf{L}^T$) are all statistically independent, the conditional probability of the scan observer at time t_i, $i = 1, 2, \ldots, T$, (6.10) is now

$$\mathrm{p}\left(\mathbf{O}_{t_i} \,|\, \mathbf{O}_{t_{i-1}}\right) \equiv \Psi^{(\mathsf{X})}\left(\mathbf{X}_{t_i} \,|\, \mathbf{X}_{t_{i-1}}\right) \Psi^{(\mathsf{K})}\left(\mathbf{k}_{t_i}\right) \Psi^{(\mathsf{L})}\left(\mathbf{l}_{t_i}\right). \tag{6.62}$$

The conditional measurement pdf for measurement $\mathbf{z}_{t_i}^{(r)}$ is now defined as

$$\mathrm{p}\left(\mathbf{z}_{t_i}^{(r)} \,\middle|\, \mathbf{O}_{t_i}\right) \equiv \mathrm{p}\left(\mathbf{z}_{t_i}^{(r)} \,\middle|\, \mathbf{X}_{t_i}, \mathbf{k}_{t_i}, \mathbf{l}_{t_i}\right) = \zeta^{(m,p)}\left(\mathbf{z}_{t_i}^{(r)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\Big|_{\substack{m=k_{t_i}^{(r)} \\ p=l_{t_i}^{(r)}}} \tag{6.63}$$

where the dependence on the sensor assignment is now explicit. (Compare this to (6.12) where implicitly $l_{t_i}^{(r)} = s$).

Similarly to (6.16) in section 6.1, the joint pdf of the measurements and the batch

observer is given by

$$\mathrm{p}\left(\mathbf{Z}^T, \mathbf{O}^T\right) \equiv \mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T, \mathbf{K}^T, \mathbf{L}^T\right) = \left\{\prod_{\nu=1}^{M} \psi^{(\nu)}\left(\mathbf{x}_{t_0}^{(\nu)}\right)\right\}$$
$$\times \prod_{i=1}^{T}\left\{\left[\prod_{m=1}^{M} \psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} | \mathbf{x}_{t_{i-1}}^{(m)}\right)\right] \prod_{r=1}^{n_{t_i}}\left[\pi_{t_i}^{(m)} \xi_{t_i}^{(p)} \zeta^{(m,p)}\left(\mathbf{z}_{t_i}^{(r)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)\Big|_{\substack{m=k_{t_i}^{(r)} \\ p=l_{t_i}^{(r)}}}\right]\right\}. \quad (6.64)$$

This differs from that for synchronous measurements (6.16) in only the measurement pdf's, because the target state pdf's have not been affected by the generalisation.

6.5.2 General msPMHT Algorithm Development

The target state estimates $\hat{\mathbf{X}}^T$ are obtained by maximising the joint pdf or observer likelihood (6.64) with respect to $\mathbf{X}^T$. Again this is treated as a missing data problem, but with the complete data now comprising the measurements $\mathbf{Z}^T$, target states $\mathbf{X}^T$, target assignments $\mathbf{K}^T$ and the sensor assignments $\mathbf{L}^T$. In this case, the missing data is the target and sensor assignments, $\mathbf{K}^T$ and $\mathbf{L}^T$. The target states, target assignments and sensor assignments are all estimated using the EM algorithm.

The general msPMHT algorithm development is similar to that of the msPMHT in section 6.2. Therefore only the key elements that differ from the previous development, and the results, will be presented.

For the general msPMHT, the auxiliary function is

$$Q \equiv Q\left(\mathbf{\Pi}^T, \mathbf{\Xi}^T, \mathbf{X}^T \,\middle|\, \mathbf{\Pi}^{T'}, \mathbf{\Xi}^{T'}, \mathbf{X}^{T'}\right) =$$
$$\sum_{\mathbf{K}^T}\sum_{\mathbf{L}^T}\left[\log \mathrm{p}\left(\mathbf{Z}^T, \mathbf{X}^T, \mathbf{K}^T, \mathbf{L}^T : \mathbf{\Pi}^T, \mathbf{\Xi}^T\right)\right] \mathrm{p}\left(\mathbf{K}^T, \mathbf{L}^T \,\middle|\, \mathbf{Z}^T, \mathbf{X}^{T'} : \mathbf{\Pi}^{T'}, \mathbf{\Xi}^{T'}\right) \quad (6.65)$$

where the dependence is now on both $\mathbf{\Pi}^T$ and $\mathbf{\Xi}^T$.

The *assignment probability*, $\omega_{t_i}^{(m,p,r)}$, represents the probability that the r^{th} measurement in the scan at t_i was produced by sensor model p and originated from target m. It is defined as

$$\omega_{t_i}^{(m,p,r)} = \frac{\pi_{t_i}^{(m)} \xi_{t_i}^{(p)} \zeta^{(m,p)}\left(\mathbf{z}_{t_i}^{(r)} \,\middle|\, \mathbf{x}_{t_i}^{(m)}\right)}{\sum_{\rho=1}^{M}\sum_{\upsilon \in \mathcal{S}^{(s)}} \pi_{t_i}^{(\rho)} \xi_{t_i}^{(\upsilon)} \zeta^{(\rho,\upsilon)}\left(\mathbf{z}_{t_i}^{(r)} \,\middle|\, \mathbf{x}_{t_i}^{(\rho)}\right)}. \quad (6.66)$$

Note that not all combinations of measurements and sensor models are possible, i.e., each measurement may only be assigned to a sub-set of the sensor models. The assignment probabilities for the illegal combinations are, by definition, zero, i.e.,

$$\omega_{t_i}^{(m,p,r)} = 0 \qquad \text{for } p \notin \mathcal{S}^{(s)} \quad (6.67)$$

where $\mathcal{S}^{(s)}$ is the sensor set of the sensor associated with the r^{th} measurement in the scan at time t_i.

Each element of the summation in the denominator of (6.66) may be viewed as either of two double finite mixtures of measurement pdf's. The first of these interpretations is a finite mixture of measurement probabilities for each target model, where these measurement probabilities are finite mixtures of measurement probabilities for each sensor model, i.e.,

$$\sum_{\rho=1}^{M} \pi_{t_i}^{(\rho)} \left[\sum_{\upsilon \in \mathcal{S}^{(s)}} \xi_{t_i}^{(\upsilon)} \zeta^{(\rho,\upsilon)} \left(\mathbf{z}_{t_i}^{(r)} \middle| \mathbf{x}_{t_i}^{(\rho)} \right) \right]. \tag{6.68}$$

The alternative is a finite mixture of measurement probabilities for each sensor model, where the measurement probabilities are finite mixtures of measurement probabilities for each target model, i.e.,

$$\sum_{\upsilon \in \mathcal{S}^{(s)}} \xi_{t_i}^{(\upsilon)} \left[\sum_{\rho=1}^{M} \pi_{t_i}^{(\rho)} \zeta^{(\rho,\upsilon)} \left(\mathbf{z}_{t_i}^{(r)} \middle| \mathbf{x}_{t_i}^{(\rho)} \right) \right]. \tag{6.69}$$

Rewriting the denominator of (6.66) as (6.68), the target assignment probability $\omega_{t_i}^{(m,r)}$, introduced in section 6.2.1 on page 74, can be obtained, i.e.,

$$\omega_{t_i}^{(m,r)} = \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)}. \tag{6.70}$$

Similarly, the *sensor assignment probability*, denoted $\bar{\omega}_{t_i}^{(p,r)}$, is defined as the probability that the r^{th} measurement in the scan at t_i was produced by sensor model p. Using (6.69), it is given as

$$\bar{\omega}_{t_i}^{(p,r)} = \sum_{m=1}^{M} \omega_{t_i}^{(m,p,r)}. \tag{6.71}$$

The EM auxiliary function for the general msPMHT becomes

$$\begin{aligned} Q = & \sum_{\nu=1}^{M} \log \psi^{(\nu)} \left(\mathbf{x}_{t_0}^{(\nu)} \right) + \sum_{i=1}^{T} \sum_{m=1}^{M} \log \psi^{(m)} \left(\mathbf{x}_{t_i}^{(m)} \middle| \mathbf{x}_{t_{i-1}}^{(m)} \right) \\ & + \sum_{i=1}^{T} \sum_{m=1}^{M} \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \left[\log \pi_{t_i}^{(m)} + \log \xi_{t_i}^{(p)} + \log \zeta^{(m,p)} \left(\mathbf{z}_{t_i}^{(r)} \middle| \mathbf{x}_{t_i}^{(m)} \right) \right] \omega_{t_i}^{(m,p,r)'} \end{aligned} \tag{6.72}$$

where $\omega_{t_i}^{(m,p,r)'}$ denotes the assignment probabilities calculated using $\mathbf{X}^{T'}$, $\mathbf{\Pi}^{T'}$ and $\mathbf{\Xi}^{T'}$.

This auxiliary function can be split into $2T + M$ independent sub-functions, i.e.,

$$Q = \sum_{i=1}^{T} Q_{t_i}^{(\Pi)} + \sum_{i=1}^{T} Q_{t_i}^{(\Xi)} + \sum_{m=1}^{M} Q_m^{(\mathbf{X})} \tag{6.73}$$

where

$$Q_{t_i}^{(\Pi)} = \sum_{m=1}^{M} \left(\log \pi_{t_i}^{(m)} \right) \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'} \qquad i = 1, 2, \ldots, T \tag{6.74}$$

$$Q_{t_i}^{(\Xi)} = \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \left(\log \xi_{t_i}^{(p)} \right) \sum_{m=1}^{M} \omega_{t_i}^{(m,p,r)'} \qquad i = 1, 2, \ldots, T \tag{6.75}$$

$$\begin{aligned} Q_m^{(\mathbf{X})} &= \log \psi^{(m)} \left(\mathbf{x}_{t_0}^{(m)} \right) + \sum_{i=1}^{T} \log \psi^{(m)} \left(\mathbf{x}_{t_i}^{(m)} \,\middle|\, \mathbf{x}_{t_{i-1}}^{(m)} \right) \\ &\quad + \sum_{i=1}^{T} \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'} \log \zeta^{(m,p)} \left(\mathbf{z}_{t_i}^{(r)} \,\middle|\, \mathbf{x}_{t_i}^{(m)} \right) \qquad m = 1, 2, \ldots, M. \end{aligned} \tag{6.76}$$

and the additional T sub-functions are a result of the inclusion of the sensor assignment probabilities.

Using (6.74) and the constraint in (6.31), the target measurement probability estimates become

$$\hat{\pi}_{t_i}^{(m)} = \frac{1}{n_{t_i}} \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'}. \tag{6.77}$$

Comparing to that obtained in the msPMHT (6.34), the scale factor has changed to reflect the number of measurements from the asynchronous sensors, and the summation includes all the sensor models allowed for each measurement.

The estimation of the sensor measurement probabilities commences with the constraint that each measurement must be assigned to exactly one sensor model in the sensor set associated with that measurement, i.e.,

$$\sum_{p \in \mathcal{S}^{(s)}} \xi_{t_i}^{(p)} = 1 \tag{6.78}$$

for each sensor set (physical sensor) providing one or more measurements in the scan at time t_i. Any sensor sets that do not provide any measurements in the scan are ignored because the associated sensor measurement probabilities are not required for the estimation process. Using this, and the auxiliary sub-function (6.75), the dual function

$$\mathcal{L}_{t_i}^{(\Xi)} = \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \left(\log \xi_{t_i}^{(p)} \right) \sum_{m=1}^{M} \omega_{t_i}^{(m,p,r)'} + \sum_{s} \gamma_{t_i}^{(s)} \left(1 - \sum_{p \in \mathcal{S}^{(s)}} \xi_{t_i}^{(p)} \right), \tag{6.79}$$

where $\sum_s$ denotes the summation over all sensor sets that provide one or more measurements at time t_i, is maximised with respect to $\xi_{t_i}^{(p)}$. This gives the estimate

$$\hat{\xi}_{t_i}^{(p)} = \frac{1}{n_{\mathcal{Z}_{t_i}^{(s)}}} \sum_{r \in \mathcal{Z}_{t_i}^{(s)}} \sum_{m=1}^{M} \omega_{t_i}^{(m,p,r)'}. \tag{6.80}$$

where $n_{\mathcal{Z}_{t_i}^{(s)}}$ is the number of measurements from the sensor set $\mathcal{S}^{(s)}$ at time t_i, and $\mathcal{Z}_{t_i}^{(s)}$ is the set of measurements in the scan at t_i from the sensor models in $\mathcal{S}^{(s)}$. If $n_{\mathcal{Z}_{t_i}^{(s)}}$ is zero, i.e., no measurements from a sensor set in a particular scan, the corresponding sensor measurement probabilities are by definition zero, and (6.80) does not hold.

Similar to the msPMHT (section 6.2.2.2), the target state estimates $\mathbf{X}^T$ may be obtained by maximising

$$\exp\left(Q_m^{(\mathbf{X})}\right) = \psi^{(m)}\left(\mathbf{x}_{t_0}^{(m)}\right) \prod_{i=1}^{T} \left[\psi^{(m)}\left(\mathbf{x}_{t_i}^{(m)} \middle| \mathbf{x}_{t_{i-1}}^{(m)}\right) \prod_{r=1}^{n_{t_i}} \prod_{p \in \mathcal{S}^{(s)}} \left(\zeta^{(m,p)}\left(\mathbf{z}_{t_i}^{(s)} \middle| \mathbf{x}_{t_i}^{(m)}\right)\right)^{\omega_{t_i}^{(m,p,r)'}} \right]. \tag{6.81}$$

For linear Gaussian systems, this is again equivalent to estimating the target states by using a fixed interval Kalman smoother. The only difference to that in section 6.2.2.2 is in the composite measurements and their covariances, which are now given as

$$\tilde{\mathbf{z}}_{t_i}^{(m)} = \tilde{\mathbf{R}}_{t_i}^{(m)} \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'} \bar{\mathbf{H}}_{t_i}^{(m,p)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,p)-1} \mathbf{z}_{t_i}^{(r)} \tag{6.82}$$

$$\tilde{\mathbf{R}}_{t_i}^{(m)} = \left(\sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'} \bar{\mathbf{H}}_{t_i}^{(m,p)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,p)-1} \bar{\mathbf{H}}_{t_i}^{(m,p)} \right)^{-1}. \tag{6.83}$$

The main difference between these and those of the msPMHT in (6.44) and (6.45) is the dual summation over all measurements and possible sensor models, instead of just over the measurements.

Consider a single sensor with multiple sensor models that differ only in their measurement noise covariances. For a nominal sensor measurement noise covariance $\mathbf{R}_{t_i}^{(m)}$, the measurement noise covariance for each model may be expressed as

$$\mathbf{R}_{t_i}^{(m,p)} = \mathcal{K}_{t_i}^{(m,p)} \mathbf{R}_{t_i}^{(m)}.$$

Substituting this into (6.82) and (6.83), the composite measurement and its covariance

may be written

$$\tilde{\mathbf{z}}_{t_i}^{(m)} = \frac{\sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'} \mathcal{K}_{t_i}^{(m,p)-1} \mathbf{z}_{t_i}^{(r)}}{\sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'} \mathcal{K}_{t_i}^{(m,p)-1}}$$

$$\tilde{\mathbf{R}}_{t_i}^{(m)} = \frac{\mathbf{R}_{t_i}^{(m)}}{\sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)'} \mathcal{K}_{t_i}^{(m,p)-1}}$$

which is equivalent to that obtained by Rago et al. (1995a) for their *homothetic PMHT gating*. Therefore, homothetic PMHT gating is a restricted case of the general msPMHT algorithm.

The msPMHT algorithm for asynchronous sensors can be obtained from the general msPMHT by allocating a single sensor model to each physical sensor. Under these conditions, the sensor model summation is only over one model and the sensor measurement probabilities revert to unity, effectively removing both from the formulation.

6.5.3 Linear Gaussian General msPMHT in Iterative Form

In similar fashion to the msPMHT (section 6.3), the general msPMHT algorithm is presented in its iterative form, as illustrated in figure 6.3.

Assuming that the measurements ($\mathbf{Z}^T$) and initial values for the target measurement probabilities ($\mathbf{\Pi}^{T'}$), the sensor measurement probabilities ($\Xi^{T'}$) and the target states ($\mathbf{X}^{T'}$) are given, the target assignment probabilities

$$\omega_{t_i}^{(m,p,r)} = \frac{\pi_{t_i}^{(m)'} \xi_{t_i}^{(p)'} \mathcal{N}\left(\mathbf{z}_{t_i}^{(r)} \middle| \mathbf{H}_{t_i}^{(m,p)} \mathbf{x}_{t_i}^{(m)'}, \mathbf{R}_{t_i}^{(m,p)}\right)}{\sum_{\rho=1}^{M} \sum_{\upsilon \in \mathcal{S}^{(s)}} \pi_{t_i}^{(\rho)'} \xi_{t_i}^{(\upsilon)'} \mathcal{N}\left(\mathbf{z}_{t_i}^{(r)} \middle| \mathbf{H}_{t_i}^{(\rho,\upsilon)} \mathbf{x}_{t_i}^{(\rho)'}, \mathbf{R}_{t_i}^{(\rho,\upsilon)}\right)} \tag{6.84}$$

are calculated for $i = 1, 2, \ldots, T$, $m = 1, 2, \ldots, M$ and $r = 1, 2, \ldots, n_{t_i}$. This is followed by the target and sensor measurement probabilities

$$\pi_{t_i}^{(m)} = \frac{1}{n_{t_i}} \sum_{r=1}^{n_{t_i}} \sum_{p \in \mathcal{S}^{(s)}} \omega_{t_i}^{(m,p,r)} \tag{6.85}$$

and

$$\xi_{t_i}^{(p)} = \frac{1}{n_{\mathcal{Z}_{t_i}^{(s)}}} \sum_{r \in \mathcal{Z}_{t_i}^{(s)}} \sum_{m=1}^{M} \omega_{t_i}^{(m,p,r)}. \tag{6.86}$$

Using (6.82) and (6.83), the composite measurements and their covariances are computed for each target model, and the target states are estimated by using a fixed interval Kalman smoother for each target model.

This sequence is repeated until the likelihood function converges to a constant value.

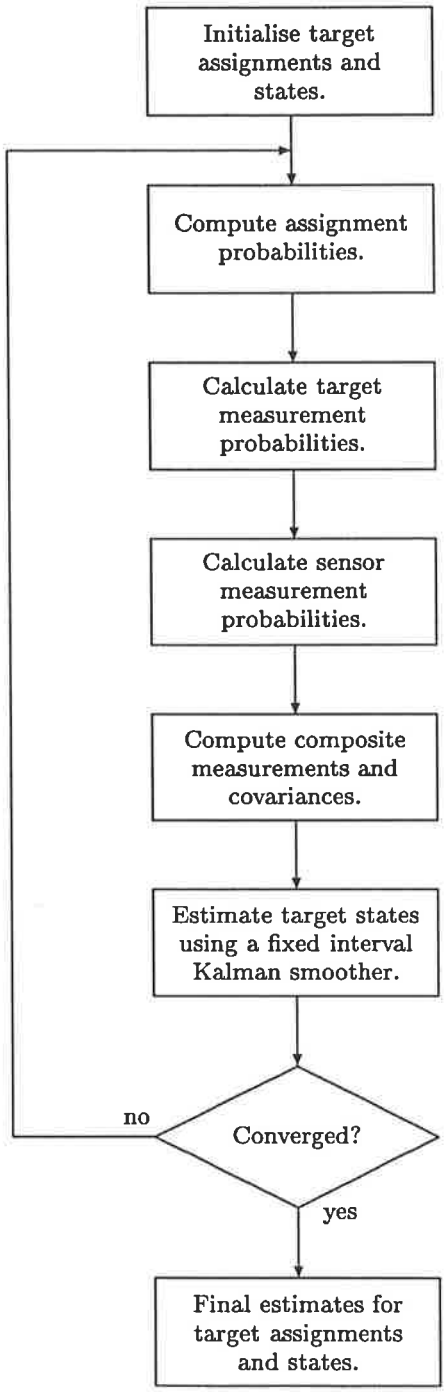


Figure 6.3: Block diagram of the iterative linear Gaussian general msPMHT algorithm

CHAPTER 7

Multi–Sensor Probabilistic Least Squares Tracking (msPLST)

The multi-sensor Probabilistic Multi-Hypothesis Tracking (msPMHT) algorithm (chapter 6) solves the multi-sensor multi-target problem by estimating both the target assignments and the target states. It does this indirectly by estimating the parameters of the underlying probability density functions of the target states and the probability mass functions of the target assignments. To ascertain which parameters to estimate, some knowledge of the underlying distributions is required. Therefore, either the distribution must be known, or it must be approximated by some other appropriate function. The assumed distribution must be a reasonable match with the data for satisfactory tracking performance.

If the probability distributions and their parameters are known, the msPMHT provides accurate estimates. However if the assumed distributions or their parameters are incorrect, the performance of the msPMHT will be degraded. The alternative is to consider an estimator that is not dependent on the probability distributions or their parameters. Such an algorithm will probably not perform as well as the msPMHT if the distributions and parameters are known, because the msPMHT is using more information. However, if the distributions or their parameters are unknown, this type of technique has the potential to outperform the msPMHT.

With this in mind, a new algorithm, the *multi-sensor Probabilistic Least Squares Tracking* (msPLST) algorithm has been developed (Krieg and Gray 1997a). This algorithm uses the same problem formulation and models as the msPMHT (chapter 5), but instead of using maximum likelihood techniques to estimate the target assignments and states, it uses *least squares*. Least squares estimation, and therefore the msPLST algorithm, assumes no knowledge of the probability distributions of the variables being estimated, using only the errors between the measurements and the predicted target states.

The concept of least squares tracking will be introduced through a simple tracking problem. This will be followed by the msPLST derivation for both synchronous and

asynchronous multi-sensor measurements.

7.1 Least Squares Estimation for Mixed Models

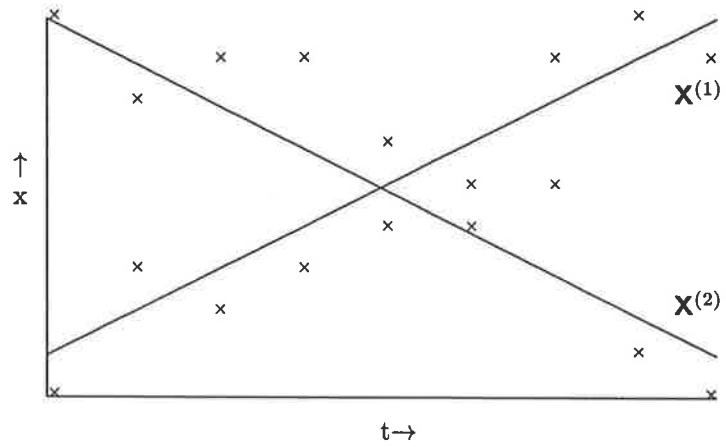


Figure 7.1: Fitting points to straight lines

Consider the tracking problem illustrated in figure 7.1. Here the two sets of noisy measurements over the times $t_1, t_2, \ldots, t_T$,

$$\begin{aligned} \mathbf{Z}^{(1)} &\equiv \left[\mathbf{z}_{t_1}^{(1)}, \mathbf{z}_{t_2}^{(1)}, \ldots, \mathbf{z}_{t_T}^{(1)}\right] \\ \mathbf{Z}^{(2)} &\equiv \left[\mathbf{z}_{t_1}^{(2)}, \mathbf{z}_{t_2}^{(2)}, \ldots, \mathbf{z}_{t_T}^{(2)}\right] \end{aligned} \tag{7.1}$$

are to be fitted to the two unknown straight line target trajectories

$$\begin{aligned} \mathbf{X}^{(1)} &\equiv \left[\mathbf{x}_{t_1}^{(1)}, \mathbf{x}_{t_2}^{(1)}, \ldots, \mathbf{x}_{t_T}^{(1)}\right] \qquad & \mathbf{x}_{t_i}^{(1)} &= \mathbf{m}^{(1)} t_i + \mathbf{c}^{(1)} \\ \mathbf{X}^{(2)} &\equiv \left[\mathbf{x}_{t_1}^{(2)}, \mathbf{x}_{t_2}^{(2)}, \ldots, \mathbf{x}_{t_T}^{(2)}\right] \qquad & \mathbf{x}_{t_i}^{(2)} &= \mathbf{m}^{(2)} t_i + \mathbf{c}^{(2)} \end{aligned} \tag{7.2}$$

where

$$\mathbf{z}_{t_i}^{(\rho)} = \mathbf{x}_{t_i}^{(\rho)} + \text{noise} \tag{7.3}$$

and $\phi = \{\mathbf{m}^{(1)}, \mathbf{c}^{(1)}, \mathbf{m}^{(2)}, \mathbf{c}^{(2)}\}$ is the set of unknown parameters specifying the target trajectories.

If the target assignments are known, i.e., $\mathbf{Z}^{(1)}$ contains only the measurements associated with target $\mathbf{X}^{(1)}$, and $\mathbf{Z}^{(2)}$ contains only those associated with $\mathbf{X}^{(2)}$, the unknown parameters of the two straight line trajectories are determined by minimising the cost function

$$\sum_{i=1}^{T} \left(\epsilon_{t_i}^{(1)}\right)^2 + \sum_{i=1}^{T} \left(\epsilon_{t_i}^{(2)}\right)^2 \tag{7.4}$$

where $\epsilon_{t_i}^{(m)}$ is the error term between a measurement and the target trajectory at that time, i.e.,

$$\epsilon_{t_i}^{(m)} = \mathbf{z}_{t_i}^{(m)} - \left(\mathbf{m}^{(m)} t_i + \mathbf{c}^{(m)}\right). \tag{7.5}$$

If the noise on the measurements varies, for example if the measurements originate from different sensors, each squared error term in the sum may be weighted by the inverse of its noise covariance. This is known as *weighted least squares.*

In practice, the target assignments are generally not known, i.e., we are only given $\mathbf{Z} \equiv \mathbf{Z}^{(1)} \cup \mathbf{Z}^{(2)}$ which contains measurements from both targets. Therefore both the target assignments and states have to be estimated to solve the tracking problem. Often this problem is not trivial because, as shown in figure 7.1, it is not always obvious to which target each measurement belongs, particularly if the targets are close or crossing. To solve this problem, a set of unknown assignment weights, $\alpha_{t_i}^{(m,r)}$, are introduced, one for each possible measurement to target assignment. These weights represent a (normalised) confidence that the r^{th} measurement at time t_i is associated with the target m. A weight of one implies that the measurement definitely originated from the target, while zero implies that it did not. It is these weights or *soft* target assignments, and not the actual *hard* assignments, that are estimated.

Allowing simultaneous measurements in a measurement scan, the error term of (7.5) is redefined as

$$\epsilon_{t_i}^{(m,r)} = \mathbf{z}_{t_i}^{(r)} - \left(\mathbf{m}^{(m)} t_i + \mathbf{c}^{(m)}\right) \tag{7.6}$$

where r denotes a particular measurement within the measurement scan. Using this redefined error term and denoting the number of measurements in the scan at time t_i as n_{t_i}, the least squares criterion for this simplified tracking problem can now be expressed as

$$\arg\min_{\phi} \sum_{m=1}^{2} \sum_{r=1}^{n_{t_i}} \sum_{i=1}^{T} \alpha_{t_i}^{(m,r)^2} \epsilon_{t_i}^{(m,r)^2} \tag{7.7}$$

with the constraint that each measurement must be assigned to exactly one target trajectory, i.e.,

$$\sum_{m=1}^{2} \alpha_{t_i}^{(m,r)} = 1 \qquad i = 1, 2, \ldots, T, \;\; r = 1, 2, \ldots, n_{t_i}.$$

This problem cannot be solved explicitly because there is insufficient information, i.e., the target assignments are missing. Therefore, as in the msPMHT, an iterative approach is implemented where, at each iteration, the target assignment weight estimates and the target trajectory parameter estimates are updated using the values obtained from the

previous iteration.

In the following sections, this simplified example is generalised to the multi-sensor multi-target tracking problem for simultaneous measurements from each sensor.

7.2 Formulation of the Cost Function

Central to the derivation of the msPLST algorithm is the development of a suitable cost function that, when minimised, produces an estimate of the observer state.

From section 5.2, the observer is defined as

$$\mathbf{O}^T \equiv [\mathbf{O}_{t_0}, \mathbf{O}_{t_1}, \ldots, \mathbf{O}_{t_T}] \qquad \mathbf{O}_{t_i} \equiv [\mathbf{X}_{t_i}, \mathbf{k}_{t_i}] \tag{7.8}$$

where $\mathbf{X}_{t_i}$ denotes the set of target state vectors at time t_i, and $\mathbf{k}_{t_i}$ the target assignments at time t_i. As each measurement scan contains a single measurement from each sensor, there are S measurements and therefore S target assignments in each scan. These target assignments,

$$\mathbf{k}_{t_i} \equiv \left[k_{t_i}^{(1)}, k_{t_i}^{(2)}, \ldots, k_{t_i}^{(S)}\right] \tag{7.9}$$

contain the target model numbers to which each measurement in the scan at time t_i is assigned. These hard assignments are replaced by the soft assignments $\alpha_{t_i}^{(m,s)}$, i.e., the target assignment weights, introduced in the previous section (section 7.1).

The estimate of the observer is denoted

$$\hat{\mathbf{O}}^T \triangleq \arg\min_{\mathbf{O}^T} J\left(\mathbf{O}^T, \mathbf{Z}^T\right) \tag{7.10}$$

where J represents the cost function. The parameters explicitly indicate that J is a function of the observer and is dependent on the batch of measurements. In the previous example (section 7.1), the cost function (7.7) is a function of the parameter set ϕ, i.e., the observer, and is also dependent on the measurements in $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$.

Consider the multi-sensor multi-target tracking problem using synchronous measurements from each sensor, as formulated in section 5.3. The cost function will obviously contain the errors between the measurements and the target state estimates, as in (7.7). However, the target state at each time cannot, in general, be determined uniquely from the other states, i.e., there is some uncertainty in the target's dynamic model. Therefore the errors between the target model and the state estimates must also be considered.

The problem formulation in section 5.3 allows for dissimilar sensors, and subsequently the noise contributed by the measurements from each will differ. Therefore some measurements will be more reliable (contain less noise) than others and will have the potential to provide more information. To ensure the more reliable measurements have a greater influence on the estimation process, a weighted least squares algorithm is used, where the

squared error terms in the cost function, e.g., $\epsilon_{t_i}^{(m,r)\,2}$ in (7.7), are scaled by some weighting function. For linear systems, these weighting functions will simply be the inverse of the appropriate noise covariances, as they provide a quantitative measure of the noise or uncertainty in the corresponding measurements or variables.

The general measurement model in (5.8) can be rewritten as the finite mixture

$$\mathbf{z}_{t_i}^{(s)} = \sum_{m=1}^{M} \pi_{t_i}^{(m,s)} h^{(m,s)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,s)}\right) \tag{7.11}$$

where the mixing parameters $\pi_{t_i}^{(m,s)}$ are either 0 or 1, and they must sum to unity for each measurement. Therefore they act as hard assignments, assigning a single target model to each measurement.

The hard assignments $\pi_{t_i}^{(m,s)}$ may be replaced by the soft assignments or target assignment weights $\alpha_{t_i}^{(m,s)}$ introduced in section 7.1, i.e.,

$$\mathbf{z}_{t_i}^{(s)} = \sum_{m=1}^{M} \alpha_{t_i}^{(m,s)} h^{(m,s)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,s)}\right). \tag{7.12}$$

In this model, each measurement is apportioned to each target, where the value $\alpha_{t_i}^{(m,s)}$ is the fraction of the measurement from sensor s at time t_i that is apportioned to target m. Therefore, as each measurement is assumed to originate from a single target, the target assignment weights for each measurement must sum to unity, i.e.,

$$\sum_{m=1}^{M} \alpha_{t_i}^{(m,s)} = 1. \tag{7.13}$$

The cost function consists of a sum of weighted squared errors that are to be minimised, e.g., the cost function in (7.7). Using the general measurement model of (5.11), the difference between the measurement $\mathbf{z}_{t_i}^{(s)}$ and the state of target m at time t_i is

$$\boldsymbol{\epsilon}_{t_i}^{(m,s)} = \mathbf{z}_{t_i}^{(s)} - h^{(m,s)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,s)}\right). \tag{7.14}$$

For a linear system, the measurement model in (5.15), with the Gaussian noise removed, may be used. The error term then becomes

$$\boldsymbol{\epsilon}_{t_i}^{(m,s)} = \mathbf{z}_{t_i}^{(s)} - \mathbf{H}_{t_i}^{(m,s)} \mathbf{x}_{t_i}^{(m)}. \tag{7.15}$$

Using the finite mixture model (7.12), the sum of all weighted squared measurement errors may be written

$$\sum_{i=1}^{T} \sum_{s=1}^{S} \sum_{m=1}^{M} \alpha_{t_i}^{(m,s)\,2} \boldsymbol{\epsilon}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \boldsymbol{\epsilon}_{t_i}^{(m,s)}. \tag{7.16}$$

The weight $\mathbf{R}_{t_i}^{(m,s)}$ is defined as

$$\mathbf{R}_{t_i}^{(m,s)} \equiv \mathrm{E}\left[\epsilon_{t_i}^{(m,s)}\epsilon_{t_i}^{(m,s)\mathsf{T}}\right] - \mathrm{E}\left[\epsilon_{t_i}^{(m,s)}\right]\mathrm{E}\left[\epsilon_{t_i}^{(m,s)\mathsf{T}}\right] \tag{7.17}$$

i.e., the measurement noise covariance of the measurements from target m and sensor s.

The error between the actual target states and those predicted by the dynamic model are obtained from (5.7), i.e.,

$$\varepsilon_{t_i}^{(m)} = \mathbf{x}_{t_i}^{(m)} - f^{(m)}\left(t_i, \mathbf{x}_{t_{i-1}}^{(m)}, \mathbf{w}_{t_{i-1}}^{(m)}\right) \tag{7.18}$$

and the linear variation, derived from (5.13) with the Gaussian noise removed, may be written as

$$\varepsilon_{t_i}^{(m)} = \mathbf{x}_{t_i}^{(m)} - \mathbf{F}_{t_{i-1}}^{(m)}\mathbf{x}_{t_{i-1}}^{(m)} \tag{7.19}$$

for all $i = 1, 2, \ldots, T$ and $m = 1, 2, \ldots, M$. The *a priori* state estimates at time t_0 are assumed known (section 5.2), and the error between this value and the target state estimate at time t_0 for target m is given as

$$\varepsilon_{t_0}^{(m)} = \mathbf{x}_{t_0}^{(m)} - \bar{\mathbf{x}}_{t_0}^{(m)}. \tag{7.20}$$

The sum of all weighted squared state errors becomes

$$\sum_{m=1}^{M} \varepsilon_{t_0}^{(m)\mathsf{T}} \bar{\Sigma}_{t_0}^{(m)-1} \varepsilon_{t_0}^{(m)} + \sum_{i=1}^{T}\sum_{m=1}^{M} \varepsilon_{t_i}^{(m)\mathsf{T}} \mathbf{Q}_{t_{i-1}}^{(m)-1} \varepsilon_{t_i}^{(m)} \tag{7.21}$$

where $\bar{\Sigma}_{t_0}^{(m)}$ is the covariance, or uncertainty, of the *a priori* state estimate at time t_0. The weight $\mathbf{Q}_{t_i}^{(m)}$ is defined as

$$\mathbf{Q}_{t_i}^{(m)} \equiv \mathrm{E}\left[\varepsilon_{t_i}^{(m)}\varepsilon_{t_i}^{(m)\mathsf{T}}\right] - \mathrm{E}\left[\varepsilon_{t_i}^{(m)}\right]\mathrm{E}\left[\varepsilon_{t_i}^{(m)\mathsf{T}}\right] \tag{7.22}$$

which represents the process noise covariance of target m at time t_i.

Therefore the cost function is obtained by combining (7.16) and (7.21), i.e.,

$$\begin{aligned} J = &\sum_{i=1}^{T}\sum_{s=1}^{S}\sum_{m=1}^{M} \alpha_{t_i}^{(m,s)2} \epsilon_{t_i}^{(m,s)\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \epsilon_{t_i}^{(m,s)} \\ &+ \sum_{m=1}^{M} \varepsilon_{t_0}^{(m)\mathsf{T}} \bar{\Sigma}_{t_0}^{(m)-1} \varepsilon_{t_0}^{(m)} + \sum_{i=1}^{T}\sum_{m=1}^{M} \varepsilon_{t_i}^{(m)\mathsf{T}} \mathbf{Q}_{t_{i-1}}^{(m)-1} \varepsilon_{t_i}^{(m)}. \end{aligned} \tag{7.23}$$

7.3 Development of the msPLST Algorithm

The cost function (7.23) must be minimised with respect to $\mathbf{X}^T$ to obtain the least squares estimate of the target states. As for the msPMHT development (section 6.2), the target assignments are unknown and can be treated as nuisance parameters.

An iterative least squares algorithm is used to estimate both the target assignments and the target states. In this approach, initial values for the target states are chosen. This may involve selecting initial target assignments and then estimating the target states using these assumed target assignments and the measurements. These state estimates are then used to estimate new target assignment weights, that are in turn used to produce new estimates for the target states. These target state estimates replace the initial state estimates, and the process is repeated until the value of the cost function converges to a constant value.

At each iteration of this algorithm, new estimates for the target assignments and the target states are chosen such that the cost function is minimised. Therefore the cost function must decrease (or remain the same) at each iteration, and therefore it converges to a minimum value (or possibly a stationary point).

7.3.1 Target Assignment Weights

To estimate the target assignment weights, the cost function (7.23) must be minimised with respect to the target assignment weights, given the most recent estimates for the target states.

This minimisation is subject to the constraints in (7.13), i.e., each measurement must be assigned to exactly one target. As in the msPMHT derivation, the constraints may be included in an equivalent dual problem by using Lagrangian multipliers. This dual function is given as

$$\mathcal{L}^{(\mathbf{A})} = \sum_{i=1}^{T}\sum_{s=1}^{S}\sum_{m=1}^{M} \alpha_{t_i}^{(m,s)\,2} \boldsymbol{\epsilon}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \boldsymbol{\epsilon}_{t_i}^{(m,s)} + \sum_{i=1}^{T}\sum_{s=1}^{S} \gamma_{t_i}^{(s)} \left(1 - \sum_{m=1}^{M} \alpha_{t_i}^{(m,s)}\right) \tag{7.24}$$

where $\gamma_{t_i}^{(s)}$ represents s^{th} Lagrangian multiplier at time t_i, and the terms in (7.23) that do not contribute to the minimisation have been omitted.

Minimising (7.24) with respect to $\alpha_{t_i}^{(m,s)}$ yields the solution

$$\hat{\alpha}_{t_i}^{(m,s)} = \frac{\left(\boldsymbol{\epsilon}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \boldsymbol{\epsilon}_{t_i}^{(m,s)}\right)^{-1}}{\sum_{\rho=1}^{M} \left(\boldsymbol{\epsilon}_{t_i}^{(\rho,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(\rho,s)\,-1} \boldsymbol{\epsilon}_{t_i}^{(\rho,s)}\right)^{-1}} \tag{7.25}$$

where $\hat{\alpha}_{t_i}^{(m,s)}$ denotes the estimated value of the target assignment weight for target m and measurement s.

7.3.2 Target State Sequences

To estimate the target states, the cost function J can be separated into M sub-functions, each representing a single target model, i.e.,

$$J = \sum_{m=1}^{M} J_m^{(\mathbf{X})} \qquad \text{where} \tag{7.26}$$

$$J_m^{(\mathbf{X})} = \sum_{i=1}^{T} \sum_{s=1}^{S} \alpha_{t_i}^{(m,s)\,2} \boldsymbol{\epsilon}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \boldsymbol{\epsilon}_{t_i}^{(m,s)} + \boldsymbol{\varepsilon}_{t_0}^{(m)\,\mathsf{T}} \bar{\boldsymbol{\Sigma}}_{t_0}^{(m)\,-1} \boldsymbol{\varepsilon}_{t_0}^{(m)} + \sum_{i=1}^{T} \boldsymbol{\varepsilon}_{t_i}^{(m)\,\mathsf{T}} \mathbf{Q}_{t_{i-1}}^{(m)\,-1} \boldsymbol{\varepsilon}_{t_i}^{(m)}. \tag{7.27}$$

The result of minimising each cost sub-function with respect to $\mathbf{X}^{(m)}$ is a set of $T+1$ simultaneous equations for each target model $m = 1, 2, \ldots, M$, i.e.,

$$\begin{array}{rcl}
\left(\bar{\boldsymbol{\Sigma}}_{t_0}^{(m)\,-1} + \mathbf{D}_{t_0}^{(m)}\right) \mathbf{x}_{t_0}^{(m)} - \mathbf{B}_{t_0}^{(m)} \mathbf{x}_{t_1}^{(m)} &=& \bar{\boldsymbol{\Sigma}}_{t_0}^{(m)\,-1} \bar{\mathbf{x}}_{t_0}^{(m)} \qquad (i=0) \\
\vdots & & \\
-\mathbf{B}_{t_{i-1}}^{(m)\,\mathsf{T}} \mathbf{x}_{t_{i-1}}^{(m)} + \left(\mathbf{A}_{t_i}^{(m)} + \mathbf{D}_{t_i}^{(m)}\right) \mathbf{x}_{t_i}^{(m)} - \mathbf{B}_{t_i}^{(m)} \mathbf{x}_{t_{i+1}}^{(m)} &=& \mathbf{c}_{t_i}^{(m)} \qquad i = 1, 2, \ldots, T-1 \\
\vdots & & \\
-\mathbf{B}_{t_{T-1}}^{(m)\,\mathsf{T}} \mathbf{x}_{t_{T-1}}^{(m)} + \mathbf{A}_{t_T}^{(m)} \mathbf{x}_{t_T}^{(m)} &=& \mathbf{c}_{t_T}^{(m)} \qquad (i=T)
\end{array} \tag{7.28}$$

where

$$\begin{array}{ll}
\mathbf{A}_{t_i}^{(m)} = \mathbf{Q}_{t_{i-1}}^{(m)\,-1} + \sum_{s=1}^{S} \alpha_{t_i}^{(m,s)\,2} \mathbf{H}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \mathbf{H}_{t_i}^{(m,s)} & i = 1, 2, \ldots, T \\
\mathbf{B}_{t_i}^{(m)} = \mathbf{F}_{t_i}^{(m)\,\mathsf{T}} \mathbf{Q}_{t_i}^{(m)\,-1} & i = 0, 1, \ldots, T-1 \\
\mathbf{c}_{t_i}^{(m)} = \sum_{s=1}^{S} \alpha_{t_i}^{(m,s)\,2} \mathbf{H}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \mathbf{z}_{t_i}^{(s)} & i = 1, 2, \ldots, T \\
\mathbf{D}_{t_i}^{(m)} = \mathbf{F}_{t_i}^{(m)\,\mathsf{T}} \mathbf{Q}_{t_i}^{(m)\,-1} \mathbf{F}_{t_i}^{(m)} & i = 0, 1, \ldots, T-1
\end{array} \tag{7.29}$$

whose solution gives the target state estimates.

This is the same set of tri-diagonal equations that were obtained for the linear Gaussian msPMHT (*see* (6.38) and (6.39)), but with $\omega_{t_i}^{(m,s)\prime}$ replaced by $\alpha_{t_i}^{(m,s)\,2}$. Therefore the composite measurement model introduced in section 6.2.2.3, i.e., (6.40), may be used here, with the composite measurements and their covariances defined as

$$\tilde{\mathbf{z}}_{t_i}^{(m)} = \tilde{\mathbf{R}}_{t_i}^{(m)} \sum_{s=1}^{S} \alpha_{t_i}^{(m,s)\,2} \bar{\mathbf{H}}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \mathbf{z}_{t_i}^{(s)} \tag{7.30}$$

$$\tilde{\mathbf{R}}_{t_i}^{(m)} = \left(\sum_{s=1}^{S} \alpha_{t_i}^{(m,s)\,2} \bar{\mathbf{H}}_{t_i}^{(m,s)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \bar{\mathbf{H}}_{t_i}^{(m,s)} \right)^{-1} \tag{7.31}$$

where $\bar{\mathbf{H}}_{t_i}^{(m,s)}$ is the measurement transformation matrix (6.42). Using the definition for the composite measurement matrix $\tilde{\mathbf{H}}_{t_i}^{(m)}$ introduced in section 6.2.2.3, $\mathbf{A}_{t_i}^{(m)}$ and $\mathbf{c}_{t_i}^{(m)}$ (7.29) may now be written as

$$\begin{aligned} \mathbf{A}_{t_i}^{(m)} &= \mathbf{Q}_{t_{i-1}}^{(m)-1} + \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_i}^{(m)-1} \tilde{\mathbf{H}}_{t_i}^{(m)} \\ \mathbf{c}_{t_i}^{(m)} &= \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_i}^{(m)-1} \tilde{\mathbf{z}}_{t_i}^{(m)} \end{aligned} \tag{7.32}$$

and, as for the linear Gaussian msPMHT, the target states may be estimated using a fixed interval Kalman smoother with this composite measurement model.

7.4 msPLST in Iterative Form

The msPLST algorithm may be conveniently expressed in iterative form.

Assume that the batch measurement ($\mathbf{Z}^T$) and initial values for the target assignment weights ($\alpha_{t_i}^{(m,s)'}$) and target states ($\mathbf{X}^{T'}$) are available. The $'$ denotes the initial values for the first iteration and the results obtained from the previous iteration for all other iterations. The algorithm is stopped when the cost function has converged to a constant value.

The target assignment weights are calculated for $i = 1, 2, \ldots, T$, $m = 1, 2, \ldots, M$ and $s = 1, 2, \ldots, S$, i.e.,

$$\alpha_{t_i}^{(m,s)} = \frac{\left(\boldsymbol{\epsilon}_{t_i}^{(m,s)'\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \boldsymbol{\epsilon}_{t_i}^{(m,s)'}\right)^{-1}}{\sum_{\rho=1}^{M} \left(\boldsymbol{\epsilon}_{t_i}^{(\rho,s)'\mathsf{T}} \mathbf{R}_{t_i}^{(\rho,s)-1} \boldsymbol{\epsilon}_{t_i}^{(\rho,s)'}\right)^{-1}} . \tag{7.33}$$

The composite measurements and their covariances are then computed for each target model using (7.31) and (7.30), and these are used in a fixed interval Kalman smoother to obtain the state estimates (*see* section 6.3).

This algorithm is summarised in figure 7.2.

7.5 Asynchronous Sensors

To facilitate asynchronous sensors, the restriction of a single measurement from every sensor in each measurement scan is lifted. The resulting algorithm development is based on the asynchronous problem formulation in section 5.4, and it closely follows the derivation of the synchronous msPLST algorithm presented in the preceding sections. Therefore only the modified cost function and results are presented here.

As in the msPMHT, the key difference between the derivation of the synchronous and asynchronous msPLST algorithms is in the indexing of the measurements and target assignments within each measurement scan. This is reflected in the measurement models and therefore in the measurement error terms in the cost function. The modified cost

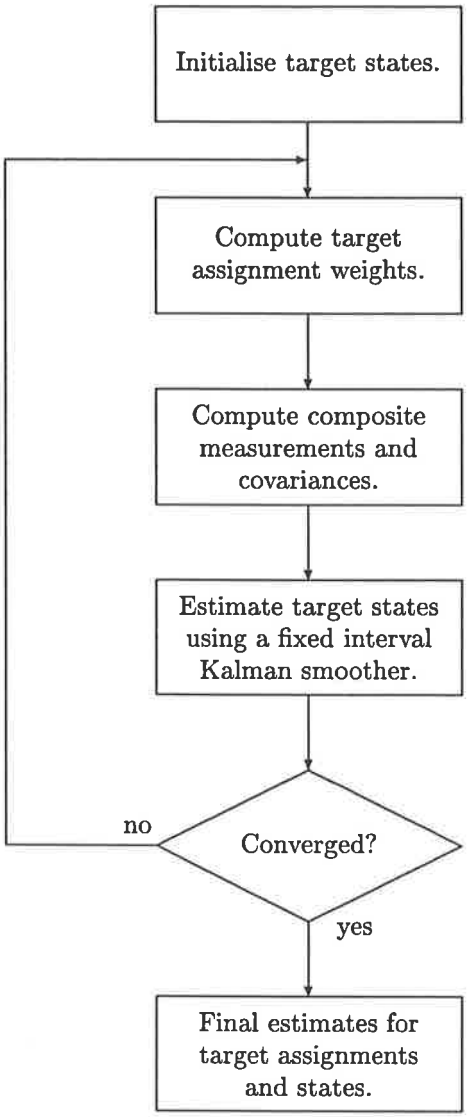


Figure 7.2: Block diagram of the iterative msPLST algorithm

function is given as

$$J = \sum_{i=1}^{T} \sum_{r=1}^{n_{t_i}} \sum_{m=1}^{M} \alpha_{t_i}^{(m,r)\,2} \epsilon_{t_i}^{(m,r)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \epsilon_{t_i}^{(m,r)} + \sum_{m=1}^{M} \epsilon_{t_0}^{(m)\,\mathsf{T}} \bar{\mathbf{\Sigma}}_{t_0}^{(m)\,-1} \epsilon_{t_0}^{(m)} + \sum_{i=1}^{T} \sum_{m=1}^{M} \epsilon_{t_i}^{(m)\,\mathsf{T}} \mathbf{Q}_{t_{i-1}}^{(m)\,-1} \epsilon_{t_i}^{(m)}. \tag{7.34}$$

where $\epsilon_{t_i}^{(m,r)}$ is defined as

$$\epsilon_{t_i}^{(m,r)} = \mathbf{z}_{t_i}^{(r)} - h^{(m,s)}\left(t_i, \mathbf{x}_{t_i}^{(m)}, \mathbf{v}_{t_i}^{(m,s)}\right) \tag{7.35}$$

and the sensor s that produced the measurement is assumed to be known (cf. (7.23) and (7.14)). The linear error term (7.15) may be similarly modified.

The target assignment weights are given as

$$\alpha_{t_i}^{(m,r)} = \frac{\left(\boldsymbol{\epsilon}_{t_i}^{(m,r)\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \boldsymbol{\epsilon}_{t_i}^{(m,r)}\right)^{-1}}{\sum_{\rho=1}^{M} \left(\boldsymbol{\epsilon}_{t_i}^{(\rho,r)\mathsf{T}} \mathbf{R}_{t_i}^{(\rho,s)-1} \boldsymbol{\epsilon}_{t_i}^{(\rho,r)}\right)^{-1}} \tag{7.36}$$

and the composite measurements and covariances, to be used by the fixed interval Kalman smoother to estimate the target states, become

$$\tilde{\mathbf{z}}_{t_i}^{(m)} = \tilde{\mathbf{R}}_{t_i}^{(m)} \sum_{r=1}^{n_{t_i}} \alpha_{t_i}^{(m,r)2} \bar{\mathbf{H}}_{t_i}^{(m,s)\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \mathbf{z}_{t_i}^{(r)} \tag{7.37}$$

$$\tilde{\mathbf{R}}_{t_i}^{(m)} = \left(\sum_{r=1}^{n_{t_i}} \alpha_{t_i}^{(m,r)2} \bar{\mathbf{H}}_{t_i}^{(m,s)\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \bar{\mathbf{H}}_{t_i}^{(m,s)} \right)^{-1} . \tag{7.38}$$

7.6 Multiple Sensor Models

The general msPMHT (*see* section 6.5) provides multiple sensor models for each physical sensor, and determines which is the most appropriate for each measurement. This allows the algorithm to adapt to changing sensor operating conditions.

The msPLST formulation does not make any assumptions regarding the pdf's of the measurements or target states. Therefore any variation in sensor models will only result in a change to the weighting of the squared measurement errors in the cost function. To achieve the desired minimum cost, any choice in these weights will result in the squared errors with the smallest weight always being selected; the other squared errors being redundant.

Therefore a generalised msPLST of this form would be of little practical use.

CHAPTER 8

Evaluation and Comparison of msPMHT and msPLST Algorithms

The structures of the linear Gaussian msPMHT and the msPLST algorithms are very similar, the major difference being the methods used to estimate the measurement to target assignments. These structures and their differences are discussed, and the performance of the algorithms are evaluated and compared using both simulated and real data.

In this analysis, a linear Gaussian system is assumed and therefore only the linear Gaussian msPMHT is evaluated. Therefore any reference to the msPMHT assumes the linear Gaussian version, unless explicitly stated otherwise.

8.1 Comparison of Algorithm Structures

Both the msPMHT and the msPLST are based on the same multi-sensor multi-target problem formulation (section 5). They are both iterative algorithms that estimate soft or probabilistic measurement to target assignments simultaneously with the target states. Within each iteration, the target states are estimated using a fixed interval Kalman smoother. The key difference is the calculation of the target assignment probabilities in the msPMHT and the target assignment weights in the msPLST. These assignments determine the the composite measurements and covariances used by the Kalman smoother, and therefore influence the target state estimation.

The target assignment probabilities and weights represent soft target assignments. The estimation of hard assignments from these soft assignments is unnecessary, because they contribute nothing new to the estimation of the target states, and it is the estimation of the target states that is the primary objective of target tracking.

For a linear Gaussian system with known covariances, the estimated parameters of the target state pdf's are the target state means. These are the maximum likelihood estimates of the actual target states, and are therefore taken as the target state estimates in the msPMHT. Using these same covariances as the weights for the squared errors in

the msPLST would provide the same target state estimates as in the msPMHT if the composite measurements and covariances were identical. (This is expected, as maximum likelihood and least squares estimates are identical for Gaussian random variables.)

However, the composite measurements and covariances are not the same in both algorithms because they depend on the soft target assignments. It is the calculation of these target assignments that distinguishes the two algorithms. The msPMHT is based on maximising the observer likelihood or probability and, as such, requires knowledge of the appropriate probability density functions. It estimates the parameters (e.g., means) of these pdf's, from which estimates of the target states and the soft or probabilistic target assignments are obtained. The msPLST uses a least squares technique, where the total weighted squared error between each measurement and the predicted target state is minimised. This does not require any knowledge of the underlying probability distributions.

8.1.1 Log Likelihood and Cost Functions

The msPMHT algorithm's estimates are obtained by maximising the EM auxiliary function (6.27). This function is derived from the expectation, over all possible measurement to target assignments, of the observer log likelihood function (*see* section 6.2.1) and, for a linear Gaussian system, is written[1]

$$\begin{aligned} Q = -J^{(X)} - \sum_{i=1}^{T} \sum_{r=1}^{n_{t_i}} \sum_{m=1}^{M} \omega_{t_i}^{(m,r)'} \boldsymbol{\epsilon}_{t_i}^{(m,r)\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)-1} \boldsymbol{\epsilon}_{t_i}^{(m,r)} \\ + \sum_{i=1}^{T} \sum_{r=1}^{n_{t_i}} \sum_{m=1}^{M} \omega_{t_i}^{(m,r)'} \log \pi_{t_i}^{(m)} \end{aligned} \tag{8.1}$$

where

$$J^{(X)} = \sum_{m=1}^{M} \boldsymbol{\varepsilon}_{t_0}^{(m)\mathsf{T}} \bar{\boldsymbol{\Sigma}}_{t_0}^{(m)-1} \boldsymbol{\varepsilon}_{t_0}^{(m)} + \sum_{i=1}^{T} \sum_{m=1}^{M} \boldsymbol{\varepsilon}_{t_i}^{(m)\mathsf{T}} \mathbf{Q}_{t_{i-1}}^{(m)-1} \boldsymbol{\varepsilon}_{t_i}^{(m)} \tag{8.2}$$

and $\boldsymbol{\epsilon}_{t_i}^{(m,r)}$, $\boldsymbol{\varepsilon}_{t_i}^{(m)}$ and $\boldsymbol{\varepsilon}_{t_0}^{(m)}$ are the measurement errors, target state errors and initial target state errors respectively, as defined in (7.15), (7.19) and (7.20).

$J^{(X)}$ contains the terms from the probability density functions that represent the dynamic evolution of the target states and contribute to the maximisation of the msPMHT log likelihood. The last summation in (8.1) contains the relevant terms from the probability mass function of the measurement to target assignments. Collectively these contain all the terms from the probability function of the observer that contribute to the maximisation of the msPMHT log likelihood function.

The summation over m in the remaining term of (8.1) can be viewed as a finite mix-

[1]Those terms that do not contribute to the maximisation have been omitted.

ture, where each term in the mixture represents the conditional probability of the r^{th} measurement at time t_i, given the state of the m^{th} target. The mixing parameters $\omega_{t_i}^{(m,r)'}$ represent the probability that the measurement originated from target m, and therefore this mixture represents the probability of the measurement given the complete observer, i.e., the states of all targets and the measurement to target assignments. Therefore maximising (8.1), which is equivalent to maximising the joint probability of the measurements and the observer, produces maximum *a posteriori* estimates, i.e., the msPMHT is a MAP estimator.

The msPLST cost function (7.23), i.e.,

$$J = J^{(X)} + \sum_{i=1}^{T} \sum_{r=1}^{n_{t_i}} \sum_{m=1}^{M} \alpha_{t_i}^{(m,r)\,2} \boldsymbol{\epsilon}_{t_i}^{(m,r)\,\mathsf{T}} \mathbf{R}_{t_i}^{(m,s)\,-1} \boldsymbol{\epsilon}_{t_i}^{(m,r)} \tag{8.3}$$

is almost the negative of (8.1) above, and minimising it is therefore similar to maximising (8.1). The key difference is the absence of the terms from the probability mass function of the target assignments, i.e., the last summation in (8.1). This is equivalent to treating the target measurement probabilities $\pi_{t_i}^{(m)}$ as constants, i.e., they take the same value for all measurement ensembles. This is clearly not the case in practice and, since this represents reduced knowledge of the system, one would expect a lower confidence in the assignment probabilities.

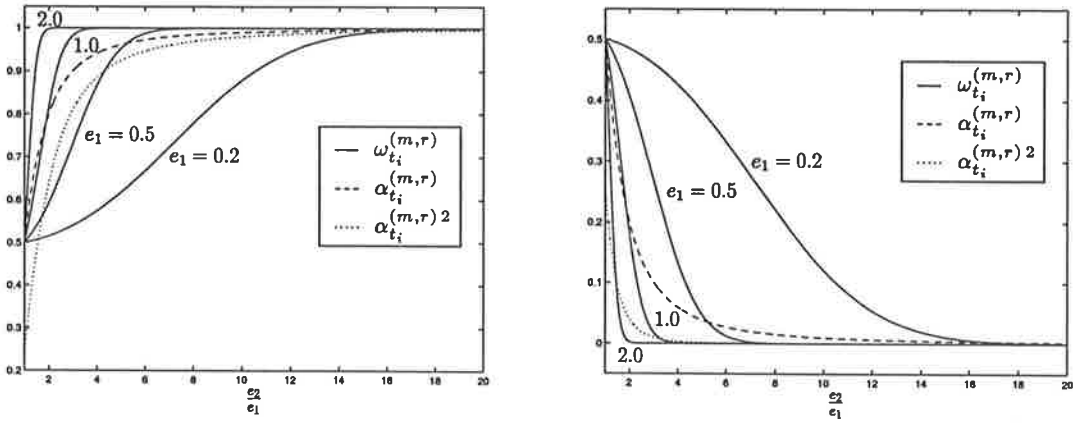
This reduced confidence in the assignment probabilities is identified in the triple summation of the cost function (8.3), where the assignment probabilities are squared. This effectively reduces the fraction of each measurement that is assigned to each target, resulting in an increase in the measurement noise covariance of the composite measurements (7.31). This indicates a decrease of confidence in the composite measurement.

The summation of the weighted squared measurement errors in (8.3) over the M target models can be viewed as the expectation of the squared measurement errors over all target models, with the assignment weights $\alpha_{t_i}^{(m,r)}$ defining the probability for each target model. Therefore the msPLST produces minimum mean squared error (MMSE) estimates. Although this is usually equivalent to maximum *a posteriori* estimation for Gaussian data, it is not so here because the measurements are modelled as finite Gaussian mixtures, and the mode, i.e., the most probable value, and the mean of a Gaussian mixture are not necessarily identical. Also in this case, the msPMHT and msPLST finite mixtures are different because the soft assignments that act as the mixing parameters are different. Therefore the msPMHT and msPLST algorithms do not produce identical estimates, even if the mixtures are Gaussian.

The MMSE estimates are the state estimates that minimise the expected sum of the squared errors between the state estimates and their true values, given the measurements. Therefore, although not equivalent to the maximum *a posteriori* estimates of the msPMHT, the msPLST estimates are in fact *a posteriori* estimates.

8.1.2 Target Assignments

Comparison of (6.50) and (7.33) indicates that the reciprocal of the measurement error is exponentiated in the msPMHT algorithm, whereas in the msPLST algorithm it is linear. From this it can readily be shown that as the error increases, the relevant msPMHT assignment probability will become smaller relative to that of the msPLST.



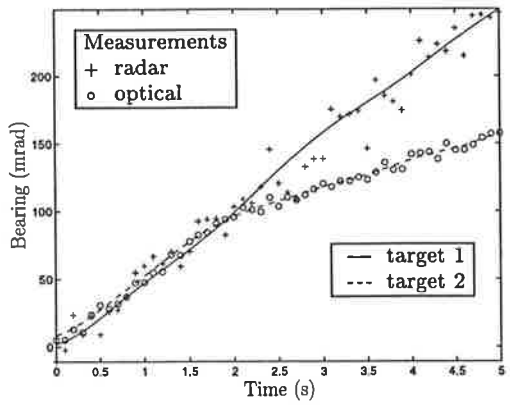
(a) Target 1

(b) Target 2

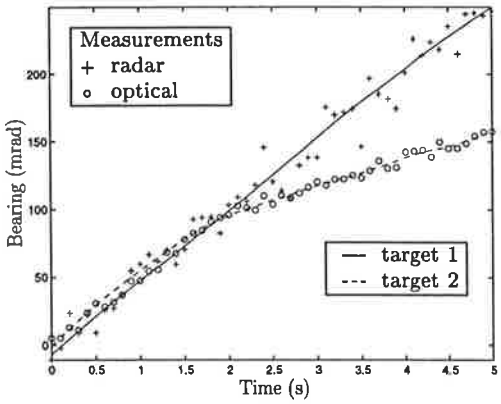
Figure 8.1: Target assignment probabilities and weights

Consider a two target example where the target measurement probabilities of each target are identical. Assuming that all measurements originate from one of the two targets, each measurement will be in the vicinity of at least one of the estimated target trajectories. Let the distance between the measurement and the closest target be denoted e_1, and the distance between the measurement and the most distant target e_2. Furthermore, assume that these error distances are expressed in units of standard deviation (σ) of measurement noise. Figure 8.1 shows the target assignment probabilities $\omega_{t_i}^{(m,r)}$ and the target assignment weights $\alpha_{t_i}^{(m,r)}$ for various ratios of e_2 to e_1. Note that because the reciprocals of the measurement errors are linear in the msPLST, the $\alpha_{t_i}^{(m,r)}$'s depend only on the ratio $\frac{e_2}{e_1}$ and therefore are the same for all values of e_1. In the msPMHT, as the reciprocal of the measurement error is exponentiated, the $\omega_{t_i}^{(m,r)}$'s are determined by the difference between e_1 and e_2. Therefore $\omega_{t_i}^{(m,r)}$ is dependent on both the ratio $\frac{e_2}{e_1}$ and the value of e_1, and the multiple solid lines in figure 8.1 represent the plots of $\omega_{t_i}^{(m,r)}$ for e_1 values of 0.2σ, 0.5σ, 1.0σ and 2.0σ. The $\alpha_{t_i}^{(m,r)\,2}$ curve has also been included for comparison.

It can be readily shown that when the distance between a measurement and the target most distant from it, i.e., e_2, is greater than approximately three standard deviations of its measurement noise, the measurement's assignment to the closest target is more likely to be harder in the msPMHT algorithm than in the msPLST. As this distance increases relative to the separation between the measurement and the closest target, the likelihood of the msPMHT assignments being harder than those in the msPLST increases.

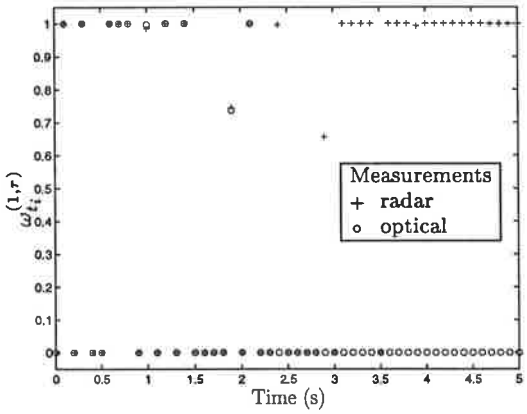


(a) msPMHT

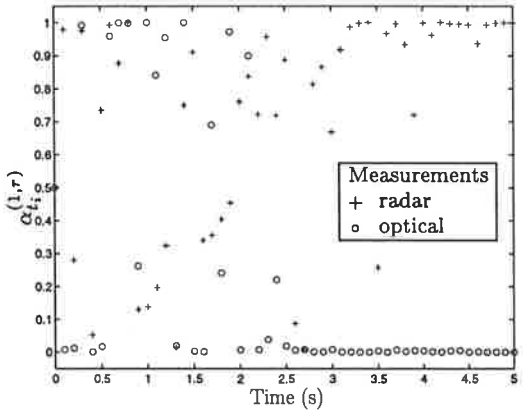


(b) msPLST

Figure 8.2: Example target tracks obtained from simulated measurements



(a) Target assignment probabilities for the msPMHT



(b) Target assignment weights for the msPLST

Figure 8.3: Example soft target assignments from simulated measurements

This explains the experimental observation that the msPLST is reluctant to make hard assignments, whereas the msPMHT more readily does so. For example, consider the simulated measurements from a radar and optical sensor and the resulting tracks shown in figure 8.2, where the track of target 1 is shown as a solid line and that of target 2 as a dashed line. Radar measurements are indicated by +'s, and optical by o's. The measurement noise covariance of the radar is ten times that of the optical.

Of particular interest is the msPMHT target assignment probabilities (figure 8.3(a)) and the msPLST target assignment weights (figure 8.3(b)), where the assignments for the radar measurements to target 1 are indicated by +'s and optical measurement assignments to target 1 by o's.

During the period before the targets diverge, both target tracks are very close together (figure 8.2). Therefore the measurements are often similar distances from both targets, giving a ratio $\frac{e_1}{e_2}$ near unity. As shown in figure 8.3(b), many of the msPLST's assignments are *soft* when this ratio is low. For the msPMHT, the assignments will be harder than

those of the msPLST if this ratio is near unity and the distance e_1 is large enough. As shown in figure 8.2, this often occurs for measurements to one side of both target tracks. For those measurements very close to one target track, i.e., e_1 is very small, the ratio $\frac{e_1}{e_2}$ becomes large, and the msPMHT assignments are again harder than those of the msPLST.

Therefore, in general, the msPMHT makes *harder* assignments, i.e., it is confident of its choices of assignment, while the msPLST only becomes sure of its assignments once the targets are well separated.

8.1.3 Track Error Covariance

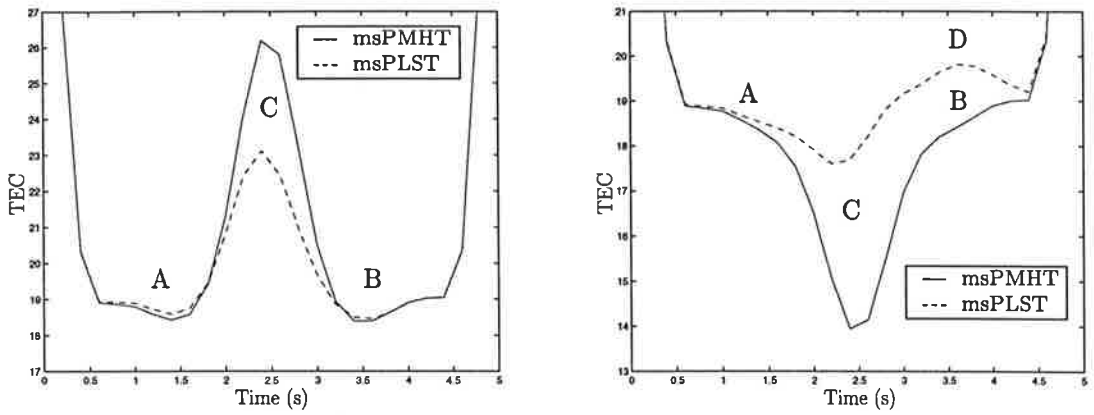
The soft target assignments form the weights for the inverse covariance of the composite measurement for each target (*see* (6.45) and (7.31)). This variation in composite measurement covariance affects the tracking performance of the algorithms, i.e., the track error covariance. For a constant measurement covariance, a target's TEC will be greatest at either end of the batch of measurements, and it will approach a constant lower value in the centre, provided the batch is long enough. If the composite measurement covariance varies, the centre region of the batch will no longer have a constant TEC.

To illustrate, consider two crossing straight line target trajectories produced from the measurements of two identical sensors. Figure 8.4 shows the different track error covariance obtained for each of the two targets from such an example.

The minimum track error covariance possible from the Kalman smoother occurs when those measurements associated with the target have an assignment of one, and all others have assignments of zero. In this case, there is no uncertainty in which measurements are assigned to each target. If the assignments take values between one and zero, the track error covariance increases because, as seen from (6.45) and (7.31), the composite measurement covariance increases. This additional covariance reflects the uncertainty in the target assignments. The worst case performance occurs when all assignments are 0.5, i.e., the algorithm has no idea which target of the two to assign each measurement to.

While the targets are well separated, the algorithms assign each of the two measurements at each time to a single target each. The msPMHT makes slightly harder assignments and this, coupled with the additional increase in the msPLST composite covariance caused by squaring the weights, results in a marginally lower composite measurement covariance in the msPMHT, and subsequently a lower TEC, as shown at 'A' and 'B' in figure 8.4. As the separation between targets increases further, the difference between the msPMHT and msPLST target assignments falls, and the TEC from both become similar.

In this example, the msPMHT assigns all measurements in the region where the targets cross to target 2. This absence of measurements assigned to target 1 causes a large increase in the covariance of the composite measurement, and this in turn increases the TEC, as shown at 'C' in figure 8.4(a). Conversely, the increase in measurements assigned to target 2



(a) Target 1

(b) Target 2

Figure 8.4: Track error covariance

provides more data and subsequently a lower composite measurement covariance, causing the decrease in msPMHT TEC shown at 'C' in figure 8.4(b). The msPLST behaves similarly, but the changes in TEC are not so pronounced. There are two reasons for this. The first is that the softer target assignments of the msPLST will result in higher assignment weights for target 1 and, even allowing for the squaring of these weights, the composite measurement covariance does not increase as much as that of the msPMHT, resulting in a lower TEC. Conversely, the msPLST assignment weights for target 2 are lower than the assignment probabilities of the msPMHT, and the decrease in composite measurement covariance and TEC for target 2 is not as great as in the msPMHT. The second reason is that one of the measurements in this region is assigned by the msPLST to target 1. This has the effect of reducing the composite measurement covariance in target 1 and increasing it in target 2.

The increase in msPLST TEC at 'D' in figure 8.4(b) is caused by a fall of 20% in the assignment weight of a single measurement to target 2 at 3.6 s, i.e., a reduction from 1.0 to 0.8. The squaring of this weight causes this fall to significantly increase the composite measurement covariance, hence the rise in TEC above the value obtained from the msPMHT. The effect of the corresponding rise, i.e., 0.0 to 0.2, in the assignment weight for target 1 is largely negated by squaring the weight when calculating the composite measurement covariance. Therefore figure 8.4(a) shows no obvious evidence of the presence of this assignment.

Therefore, for widely separated targets, the TEC of the msPMHT is marginally superior to that of the msPLST. For close targets, the TEC depends on the assignment of measurements to each target. Generally, for a particular target, the msPMHT TEC is lower than that of the msPLST if most measurements are assigned to that target, and higher if few measurements are assigned to the target.

Note that had equal numbers of measurements been assigned to each target in the

vicinity of the point where the targets cross, the msPMHT would have produced a TEC at 'C' similar to that at 'A' and 'B', by virtue of its harder assignments. The TEC of the msPLST would have been significantly higher at 'C' than at 'A' because of its softer assignments at this time. Therefore, under these conditions, the TEC of the msPMHT is significantly lower than that of the msPLST.

8.2 Algorithm Initialisation

The msPLST and msPMHT both converge to the nearest stationary point, making initialisation critical. Therefore it is essential that the algorithm be initialised in the vicinity of the global maximum (msPMHT) or minimum (msPLST).

Giannopoulos et al. (1996) introduced *measurement covariance deflation* for initialising the PMHT algorithm. It attempts to smooth the multi-modal log likelihood function of the PMHT by increasing the measurement noise covariances. The PMHT algorithm is then run using these increased covariances and, because many local maxima have been removed, the resulting log likelihood should be somewhere near its global maximum. The PMHT is then re-initialised using the estimates obtained from this run, the covariances are decreased and the algorithm is run again. This step is repeated until the covariances reach their correct values.

This technique is used for initialising the msPMHT and msPLST algorithms. In each measurement covariance deflation step, the algorithm is run until a (relaxed) convergence criteria is achieved. To further smooth the log likelihood or cost function, the process noise covariances are initially decreased, and then increased at each measurement covariance deflation step. This reduces the manoeuvrability of the target, preventing it from following individual measurements.

To commence this initialisation procedure, the measurements are randomly assigned to the available targets.

The criteria for convergence, i.e., the magnitude of the change in the log likelihood or cost function between iterations, is a trade off between number of iterations and the cost error on algorithm termination. Little improvement in the track estimates is obtained by an excessive number of iterations. In most examples presented here, a change in log likelihood or cost between iterations of 0.01% for the initial measurement covariance deflation steps, and 0.0001% for the final step, was found to provide reasonable results.

Table 8.1 shows the statistics for 100 *monte carlo* simulations using the same data but different initialisations. These results, for well separated crossing targets (see figure 8.6), show little change in the log likelihood and cost functions. Repeating the *monte carlo* simulations with different data ensembles gave the results in table 8.2. Again the log likelihood and cost functions do not widely differ, most the difference being due to the different noise on each ensemble.

	msPMHT	msPLST
mean	-166.0	37.6
std dev	2.60×10^{-5}	6.87×10^{-14}

Table 8.1: Log likelihood and cost function statistics for same data with different initialisations

	msPMHT	msPLST
mean	-166.3	37.4
std dev	4.2	7.9

Table 8.2: Log likelihood and cost function statistics for different data ensembles

Tables 8.3 and 8.4 show the average number of iterations at each measurement covariance deflation step for a crossing target example and diverging target example respectively. The *covariance scale* in these tables represents the scale factor applied to the measurement covariances at each measurement covariance deflation step. The process noise covariances were scaled by the reciprocal of these values.

Covariance scale		100.0	50.0	25.0	13.0	6.0	3.0	1.5	1.0	Total
msPMHT	mean	7.4	5.5	5.8	5.4	6.2	6.1	5.3	28.4	70.2
(log likelihood)	std dev	1.5	1.6	2.7	1.6	2.8	2.9	3.5	23.1	22.9
msPLST	mean	5.9	2.8	2.7	2.9	3.1	3.1	3.3	5.3	29.1
(cost)	std dev	0.6	0.4	0.4	0.4	0.5	0.3	0.6	2.1	3.1

Table 8.3: Average number of iterations at each level of covariance deflation for a crossing target example

Covariance scale		100.0	50.0	25.0	13.0	6.0	3.0	1.5	1.0	Total
msPMHT	mean	4.0	3.0	2.9	13.7	35.3	14.8	13.9	97.9	185.5
(log likelihood)	std dev	6.9	4.2	3.3	19.2	15.2	7.5	8.8	74.4	72.2
msPLST	mean	10.3	6.8	7.7	8.3	6.9	7.0	6.4	9.5	63.0
(cost)	std dev	2.0	1.6	3.3	5.2	3.4	5.2	4.5	8.3	11.8

Table 8.4: Average number of iterations at each level of covariance deflation for a diverging target example

These results indicate that the msPLST performs more iterations in the earlier steps compared to the msPMHT. Note that at least two iterations are mandatory to determine if convergence has been reached. In both cases, the msPMHT performs about a third of its iterations in the last step, compared to the msPLST which performs about a sixth of its total iterations in the final step. This is because the msPMHT is more prone to covariance errors when calculating its target assignments (see section 8.6), and therefore more correction is required at this step, where the true covariance values are used.

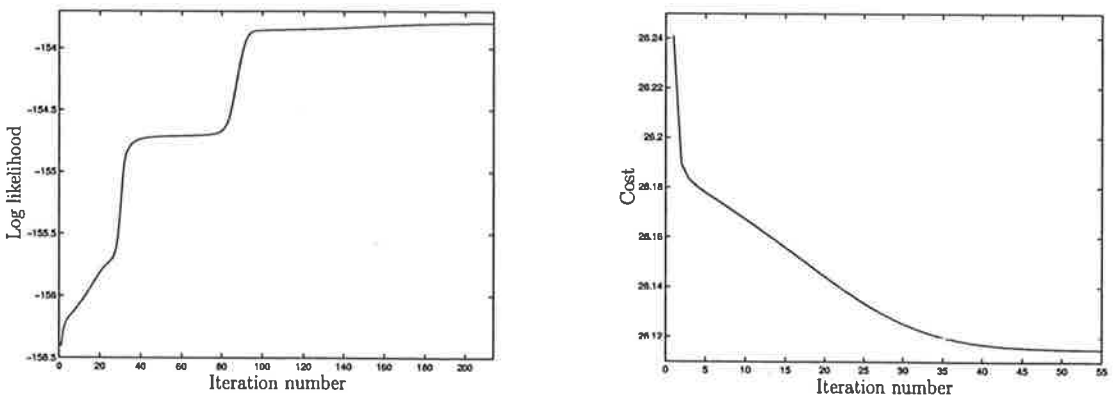
8.3 Computational Complexity

Both algorithms are computationally similar within each iteration, but in most cases the msPMHT took significantly more iterations than the msPLST algorithm to reach convergence. Examples from *monte carlo* simulations for five different two target two sensor scenarios are shown in figure 8.5 for identical batch lengths. From these examples, it can be seen that the msPMHT took between 2 and 3 times the number of iterations to converge than the msPLST. The standard deviations are generally greater for the msPMHT, indicating that the number of iterations in the msPMHT fluctuate more than in the msPLST.

Example		1	2	3	4	5
msPMHT	mean	70.2	144.7	80.0	137.8	185.5
	standard deviation	22.9	44.3	19.5	14.1	72.2
msPLST	mean	29.1	50.7	29.7	48.7	63.0
	standard deviation	3.1	12.9	5.0	14.1	11.8
Ratio of means		2.4	2.9	2.7	2.8	2.9

Table 8.5: Comparison of total iterations for msPMHT and msPLST

The reason for the larger number of iterations in the msPMHT is obvious from the example likelihood function in figure 8.5(a). This shows a series of plateaus that not only slowed the rate of convergence, but also made it difficult to determine when convergence had occurred. Note that not all msPMHT likelihood functions contain such obvious plateauing. The msPLST cost function (figure 8.5(b)) gradually approaches convergence without plateauing. The rate of convergence is therefore faster than for the msPMHT, and it is also much easier to determine when the algorithm has converged.



(a) msPMHT

(b) msPLST

Figure 8.5: Example log likelihood and cost functions

8.4 Evaluation Using Simulated Data

A number of two sensor two target scenarios have been simulated. The performance of the msPMHT and msPLST algorithms have been evaluated using 100 *monte carlo* simulations for each of these scenarios, and the average or mean results are presented with error bars indicating the standard deviation of the results both above and below the mean value of the estimated target tracks. In these examples, the measurements from sensor 1 are indicated by $\times$'s and the measurements from sensor 2 by $\circ$'s. Simultaneous measurements from each sensor are assumed. Note that the illustrated measurements are examples from a single ensemble.

Only the single dimension *position* measurements and tracks are presented for clarity. As for the AFKF, it is assumed that the dynamic behaviour of each target in one dimension is independent of that of other dimensions, e.g., bearing is independent of range.

Only the target measurement probabilities for target 1, $\pi_{t_i}^{(1)}$, are presented as the sum of the measurement assignments for both targets is unity in each measurement scan. Similarly, the measurement assignment probabilities $\omega_{t_i}^{(m,r)}$ and weights $\alpha_{t_i}^{(m,r)}$ also sum to unity for each measurement, and therefore only $\omega_{t_i}^{(1,r)}$ and $\alpha_{t_i}^{(1,r)}$ for $r = 1, 2$ are presented. Similar to that of the msPMHT (6.51), the msPLST target measurement probability for target m is defined as the mean of the target assignment probabilities at that time, i.e.,

$$\pi_{t_i}^{(m)} \equiv \frac{1}{S} \sum_{r=1}^{S} \alpha_{t_i}^{(1,r)} \tag{8.4}$$

where, in this case, $S = 2$ denotes the number simultaneous measurements in each scan.

8.4.1 Crossing Targets with Similar Sensors

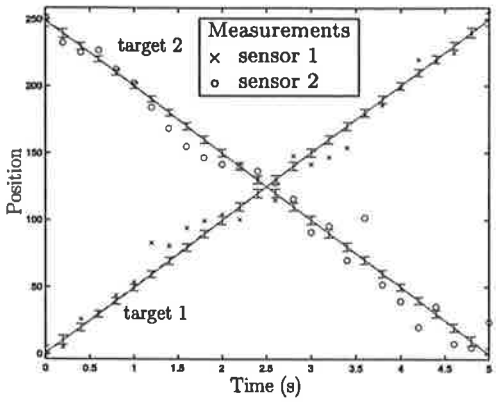
The msPMHT and msPLST algorithms are applied to the problem of tracking crossing targets. In these simulations, the target moving in the direction of increasing position is labelled as target 1, and the other as target 2.

The measurements in these examples arise from similar sensors, i.e., sensors with the same measurement models. In practice, this may represent multiple measurements from a single sensor, or measurements from multiple sensors of the same type.

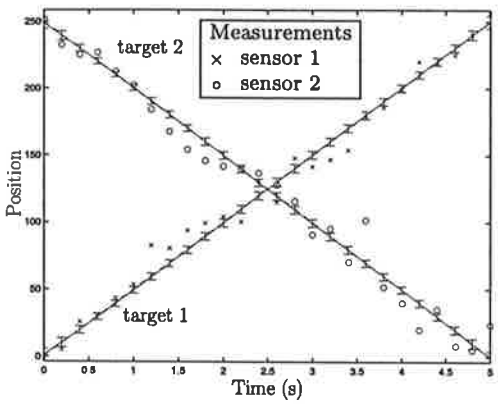
Linear Gaussian dynamic and measurement models have been assumed in all cases.

8.4.1.1 Well Separated Targets

The first example, whose msPMHT and msPLST tracks are displayed in figure 8.6, considers two targets with significantly different velocities crossing the trajectories of each other. In this scenario, the targets are well separated for much of the time. If this represents a two sensor system, the measurements from sensor 1 originate from target 1, and sensor 2 follows target 2.

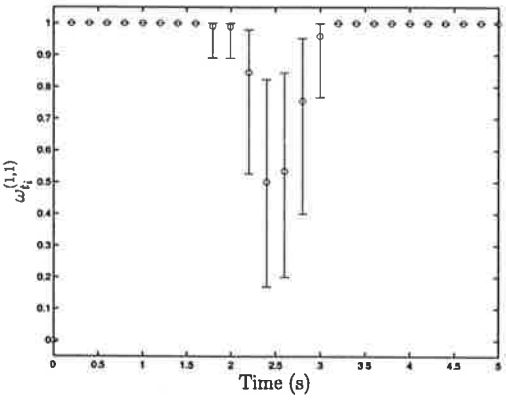


(a) msPMHT

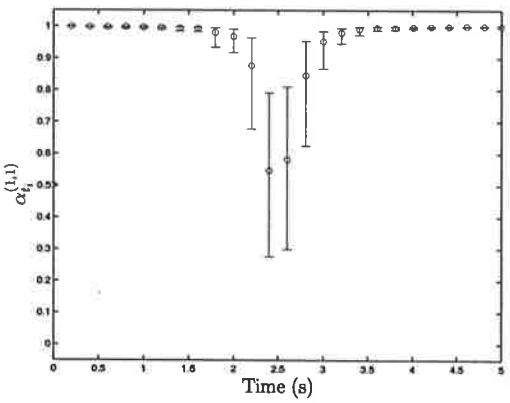


(b) msPLST

Figure 8.6: Target track statistics for crossing targets using similar sensors

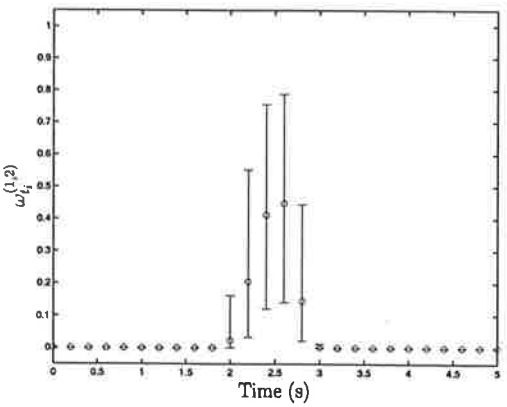


(a) msPMHT ($\omega_{t_i}^{(1,1)}$)

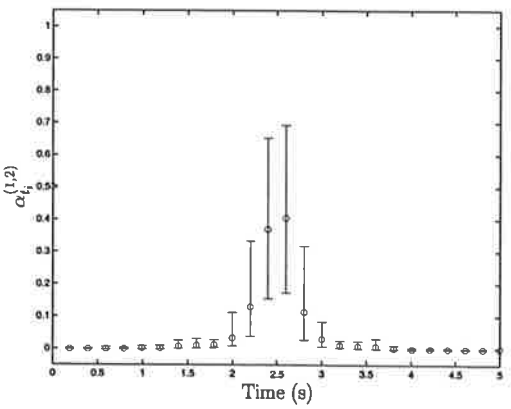


(b) msPLST ($\alpha_{t_i}^{(1,1)}$)

Figure 8.7: Measurement to target assignments of target 1 and sensor 1 for crossing targets using similar sensors

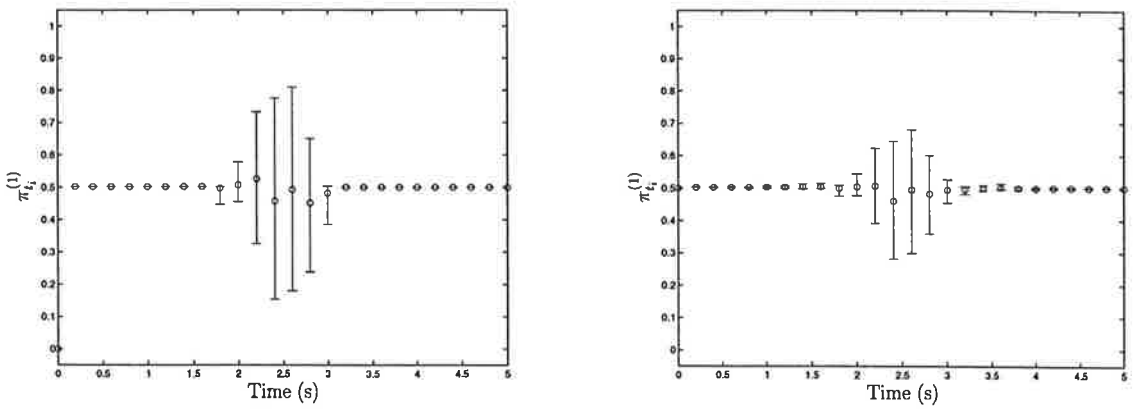


(a) msPMHT ($\omega_{t_i}^{(1,2)}$)



(b) msPLST ($\alpha_{t_i}^{(1,2)}$)

Figure 8.8: Measurement to target assignments of target 1 and sensor 2 for crossing targets using similar sensors



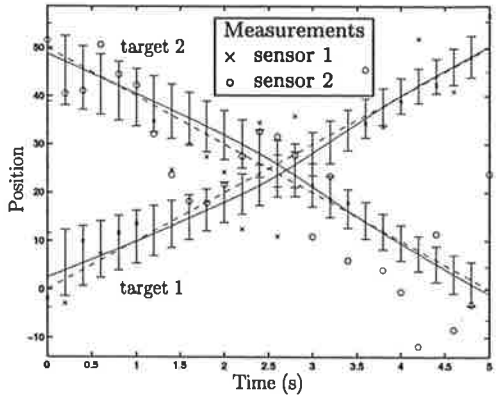
(a) msPMHT (b) msPLST

Figure 8.9: Target 1 measurement probabilities for crossing targets using similar sensors

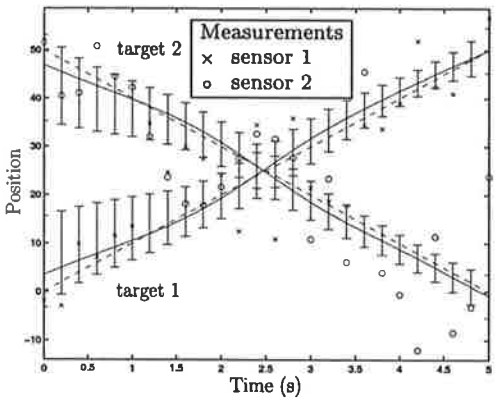
As shown in figure 8.6, both algorithms produce statistically similar tracks, with almost identical means and covariances. In the msPLST, the target tracks are restricted to a few different trajectories, while the msPMHT produces a much more varied collection of possible trajectories. The amount of this variation is related to the measurement noise and process noise covariances. Obviously, if the measurement noise increases, more measurements will occur further from the true target position, giving more possibilities for potential target tracks. The higher the process noise covariance, the more the target is able to manoeuvre, and therefore the greater the range of possible target trajectories.

Each of the target trajectories only differs about the crossing point, i.e., when the targets are close and the assignment decisions are more difficult. When the targets are well separated, both algorithms produce the same target assignments in all cases, as shown by the coincident error bars in figures 8.7 and 8.8. At the point of crossing, the uncertainty in the average msPMHT target assignments is greater than for the msPLST, however as the targets begin to separate, the msPMHT is able to consistently make harder more confident assignments than the msPLST. This is reflected in the target measurement probabilities that, as shown in figure 8.9, remain at about 0.5, i.e., half the measurements are assigned to each target on average, but in individual ensembles this varies when the targets are close. The greater uncertainty in the average assignments of the msPMHT for close targets arises from the harder assignments produced by that algorithm, i.e., although the average is approximately 0.5, the individual values are close to one or zero.

Therefore, the tracking performance of the algorithms are similar when the targets are well separated. In this case, both algorithms consistently produce the same relatively hard target assignments, and therefore similar composite measurements for the Kalman smoother. As the targets move closer together, the assignments of the msPLST become softer while the msPMHT is more likely to maintain harder assignments.

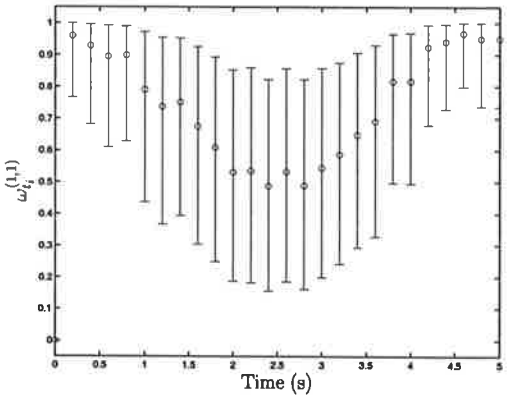


(a) msPMHT

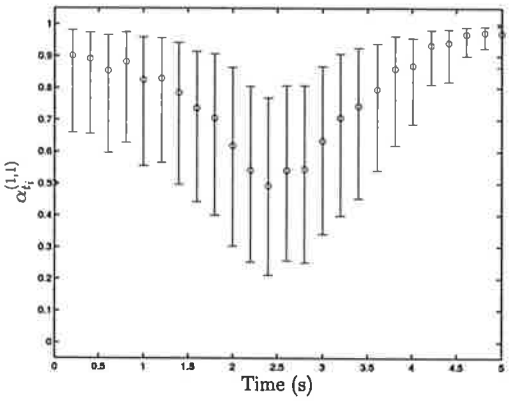


(b) msPLST

Figure 8.10: Target track statistics for close crossing targets using similar sensors

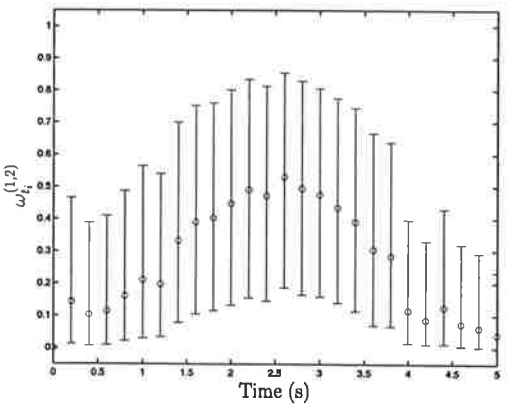


(a) msPMHT ($\omega_{t_i}^{(1,1)}$)

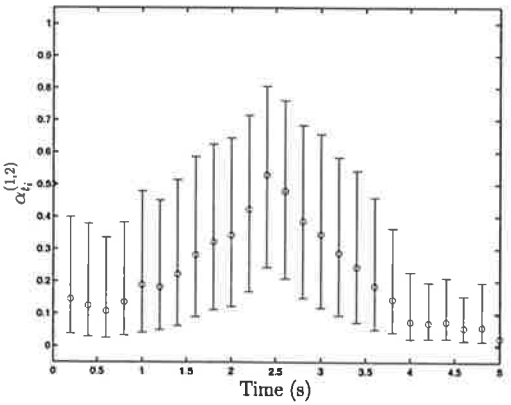


(b) msPLST ($\alpha_{t_i}^{(1,1)}$)

Figure 8.11: Measurement to target assignments of target 1 and sensor 1 for close crossing targets using similar sensors

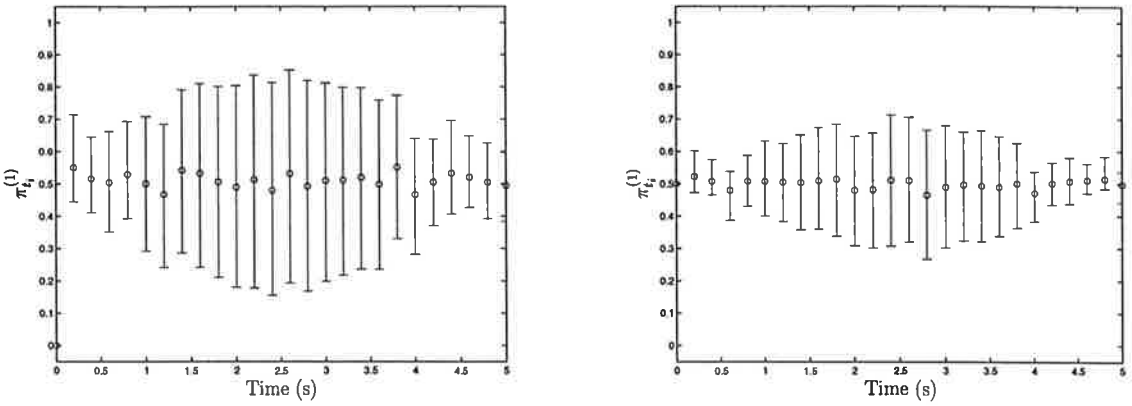


(a) msPMHT ($\omega_{t_i}^{(1,2)}$)



(b) msPLST ($\alpha_{t_i}^{(1,2)}$)

Figure 8.12: Measurement to target assignments of target 1 and sensor 2 for close crossing targets using similar sensors



(a) msPMHT (b) msPLST

Figure 8.13: Target 1 measurement probabilities for close crossing targets using similar sensors

8.4.1.2 Poorly Separated Targets

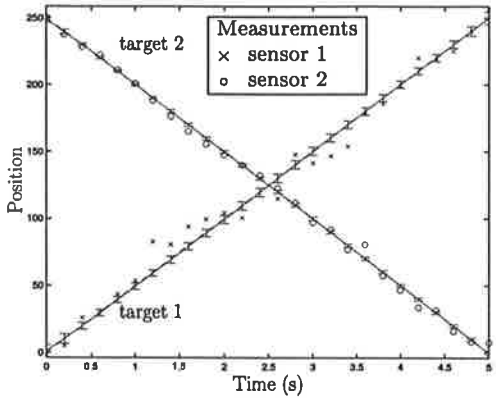
The above example was repeated, but with targets that are not as well separated. In this case, as shown in figure 8.10, the variation in the target tracks is similar. (Note the different scales from the example in the previous section.) However, the number of variations in possible tracks increased significantly from that in the previous example, including scenarios where the targets approached each other and then turned away without crossing.

The target assignment probabilities and weights in figures 8.11 and 8.12 show that when the targets are close, they are on average equally assigned to each target. Again the larger error bars in the msPMHT assignments indicate that they are generally harder than those of the msPLST. This is reinforced by the target measurement probabilities, as shown in figure 8.13 where, although about half the measurements are assigned to each target on average, the individual measurement probabilities of the msPMHT tend to be closer to one or zero than their msPLST counterparts.

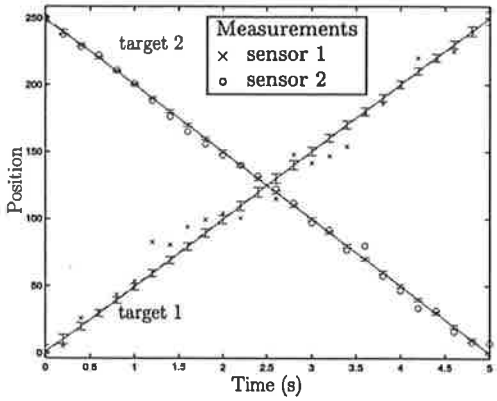
When tracking closely spaced targets, the msPMHT is less consistent with its assignments than the msPLST between data ensembles. This probabilistic behaviour is consistent with that of a human operator.

8.4.2 Crossing Targets With Dissimilar Sensors

The previous two examples are repeated in this section, but using measurements from two dissimilar sensors. In this case, the measurement noise covariance of sensor 2 is one tenth that of sensor 1.

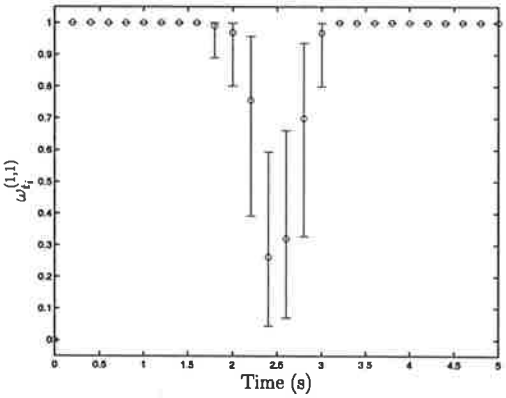


(a) msPMHT

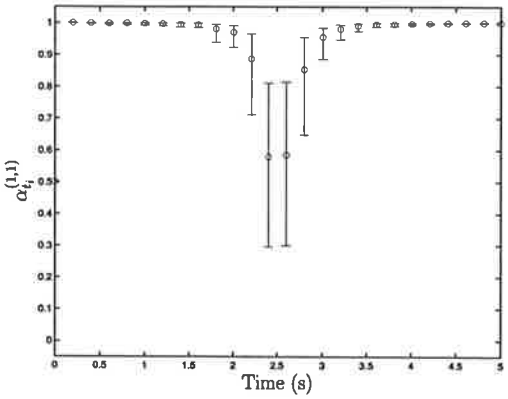


(b) msPLST

Figure 8.14: Target track statistics for crossing targets using dissimilar sensors

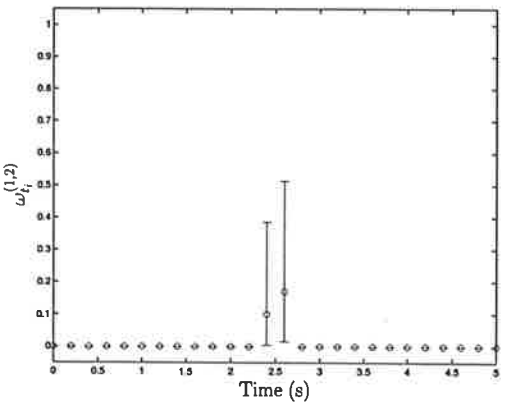


(a) msPMHT ($\omega_{t_i}^{(1,1)}$)

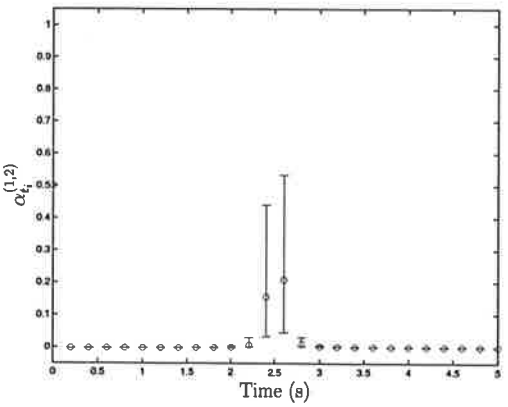


(b) msPLST ($\alpha_{t_i}^{(1,1)}$)

Figure 8.15: Measurement to target assignments of target 1 and sensor 1 for crossing targets using dissimilar sensors

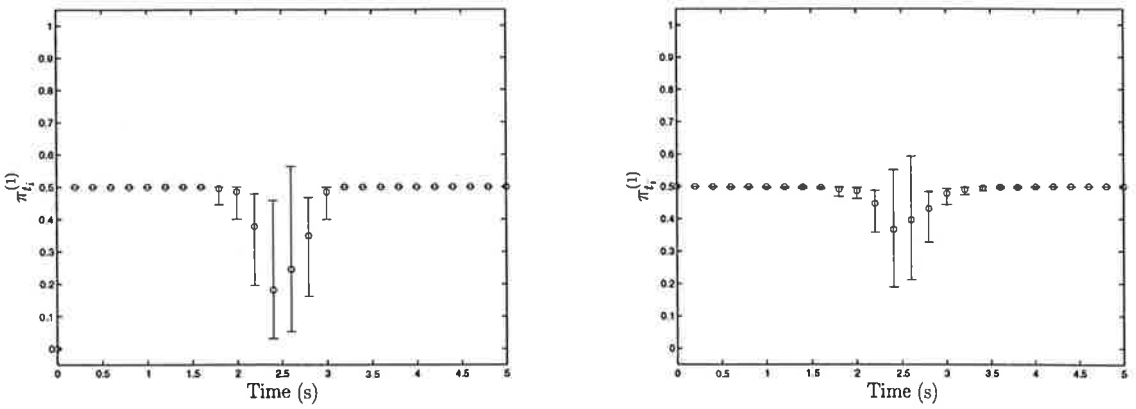


(a) msPMHT ($\omega_{t_i}^{(1,2)}$)



(b) msPLST ($\alpha_{t_i}^{(1,2)}$)

Figure 8.16: Measurement to target assignments of target 1 and sensor 2 for crossing targets using dissimilar sensors



(a) msPMHT (b) msPLST

Figure 8.17: Target 1 measurement probabilities for crossing targets using dissimilar sensors

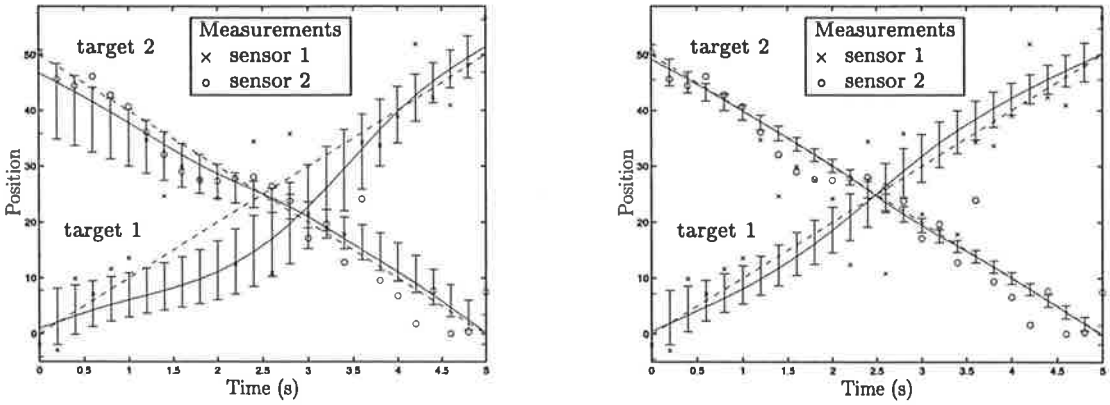
8.4.2.1 Well Separated Targets

The tracks for well separated targets in figure 8.14 are similar to those obtained for similar sensors, with the exception that the covariance in the track estimates for target 2 are much smaller. This occurs because, as illustrated in figures 8.15 and 8.16, only the lower covariance measurements from sensor 2 are assigned to target 2, except near where the targets cross. As the targets approach the point where they cross, most the measurements from both sensors tend to be assigned to target 2, because many of the measurements from sensor one are closer to target 2 than target 1. The lower noise on the measurements from sensor 2 ensure that they are more likely to remain closer to target 2 than target 1. This is more evident in the msPMHT because of the nature of the Gaussian probability distributions. The measurement probability for target 1 (figure 8.17) shows how the fraction of measurements assigned to target 1 falls as the targets cross.

8.4.2.2 Poorly Separated Targets

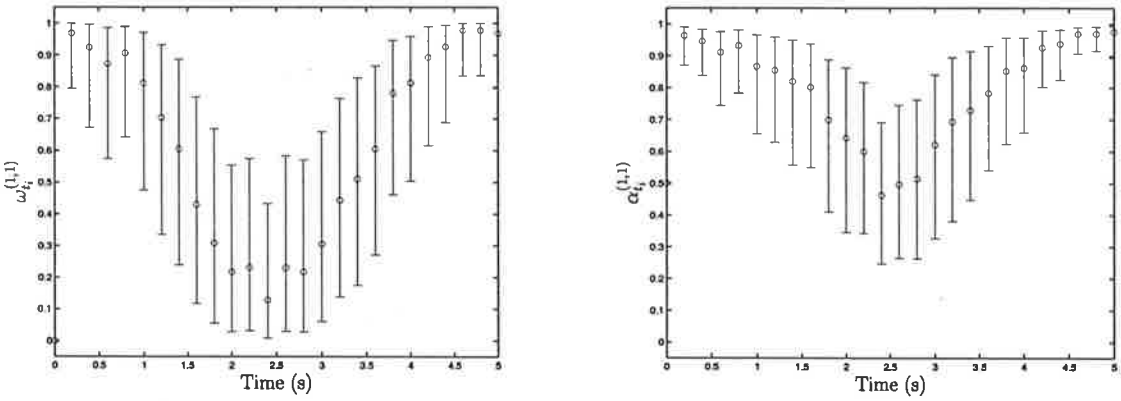
To further evaluate the algorithms performance with closely space targets, the example in section 8.4.1.2 is repeated using measurements from dissimilar sensors. Again the measurement noise covariance of sensor 2 is one tenth that of sensor 1.

In this case, as shown in figure 8.18, the performance of the algorithms is markedly different. The msPMHT track for target 1 deviates significantly from the true target position, indicated by the dashed line, when the two targets are very close. This occurs because, as in the previous section, most the measurements are assigned to target 2 when the targets are in close proximity. This is illustrated by the target assignment probabilities and weights in figures 8.19 and 8.20, where the assignments favour target 2 during this time. The msPLST assignments are softer and it tends to assign measurements at this time to both targets, thereby assigning a sufficient fraction of the measurements



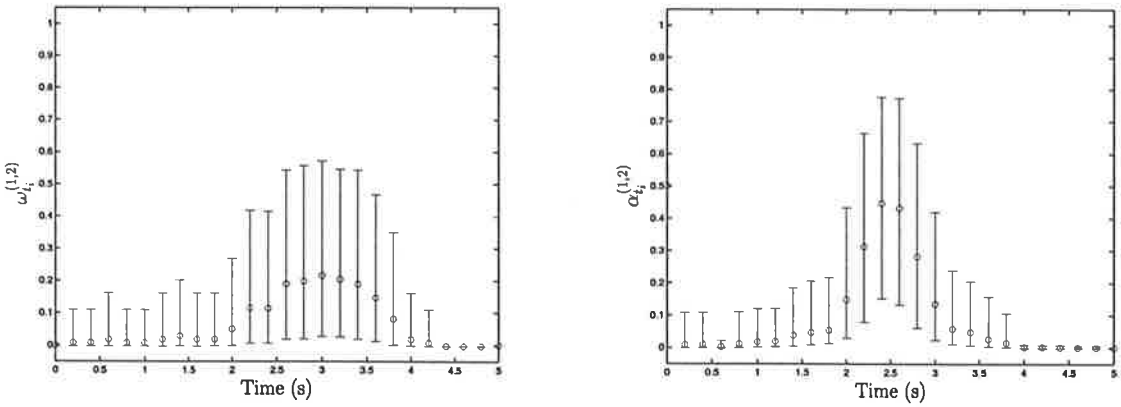
(a) msPMHT (b) msPLST

Figure 8.18: Target track statistics for close crossing targets using dissimilar sensors



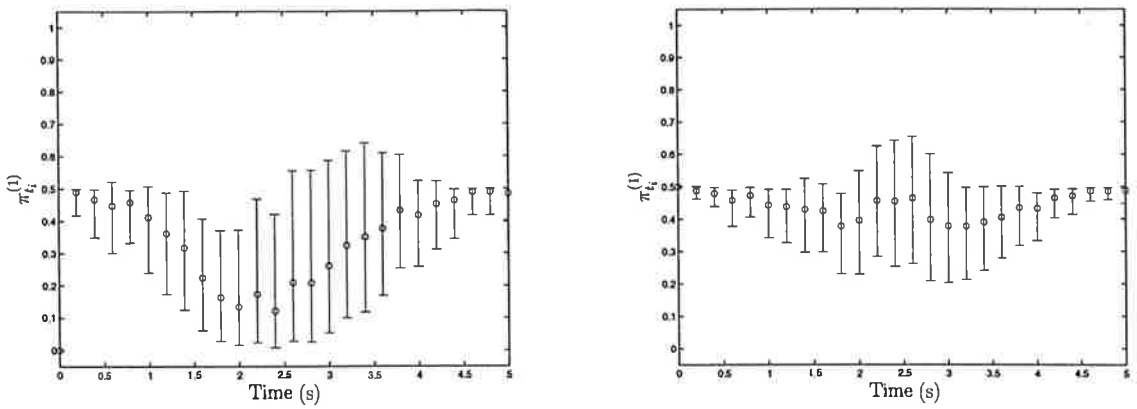
(a) msPMHT ($\omega_{t_i}^{(1,1)}$) (b) msPLST ($\alpha_{t_i}^{(1,1)}$)

Figure 8.19: Measurement to target assignments of target 1 and sensor 1 for close crossing targets using dissimilar sensors



(a) msPMHT ($\omega_{t_i}^{(1,2)}$) (b) msPLST ($\alpha_{t_i}^{(1,2)}$)

Figure 8.20: Measurement to target assignments of target 1 and sensor 2 for close crossing targets using dissimilar sensors



(a) msPMHT (b) msPLST

Figure 8.21: Target 1 measurement probabilities for close crossing targets using dissimilar sensors

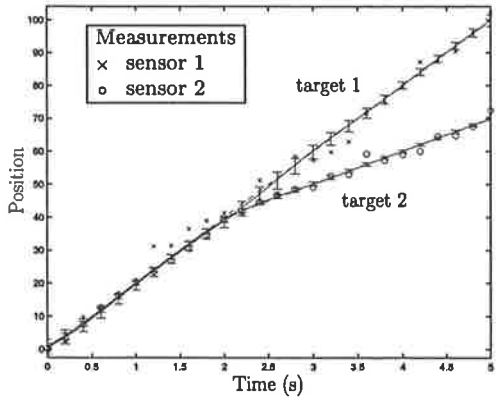
to target 1 to prevent the track deviating from its true trajectory. Figure 8.21 shows how most measurements are assigned to target 2 in the msPMHT, but the msPLST assigns approximately equal numbers of measurements to each target.

The initial large average deviation of the msPMHT track for target 2 (figure 8.18) arises from the assignment of measurements from sensor 1 to target 2 in some ensembles during this period, as indicated by the large error bars in figure 8.19(a). This possibility exists because the target separation is small in terms of the error covariance of sensor 1, and such assignments have the effect of pulling the estimated track of target 2 towards target 1.

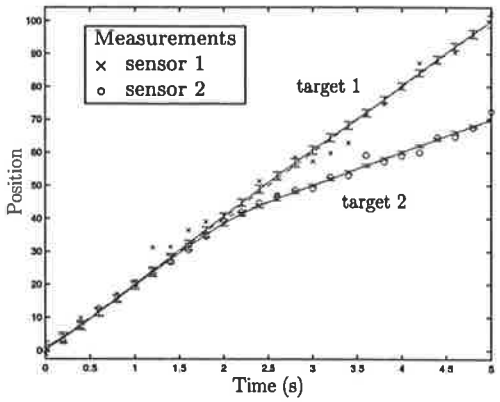
8.4.3 Diverging Targets with Dissimilar Sensors

The final simulated example involves tracking diverging targets with dissimilar sensors, with the measurement noise covariance of sensor 2 again one tenth that of sensor 1. In this example, both sensors are following a formation of two targets, which then split with each sensor following a different target. The tracks obtained from the msPMHT and msPLST algorithms are shown in figure 8.22.

The target assignment probabilities and weights (figures 8.23 and 8.24) illustrate how initially the measurements are shared between the targets, but after divergence the measurements from sensor 1 are assigned to target 1 and those from sensor 2 are assigned to target 2. The larger error bars of the msPMHT assignments prior to divergence again indicate this algorithm's tendency to make harder assignments, albeit to different targets across the data ensembles. As the targets separate, these assignments become more consistent over the ensembles. The target measurement probabilities in figure 8.25 illustrate how the fraction of measurements assigned to each target differs between data ensembles before divergence, but consistently approaches 0.5 as the targets separate.

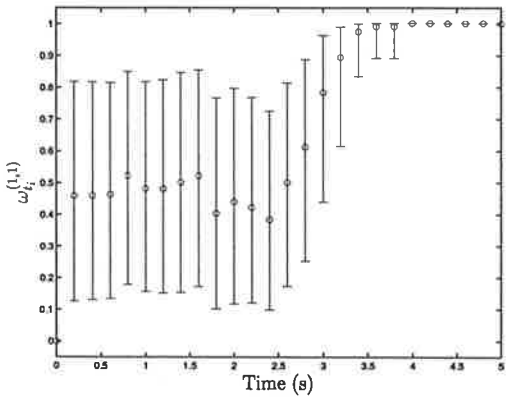


(a) msPMHT

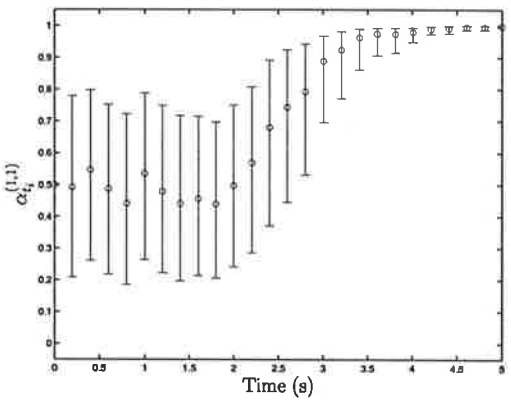


(b) msPLST

Figure 8.22: Target track statistics for diverging targets using dissimilar sensors

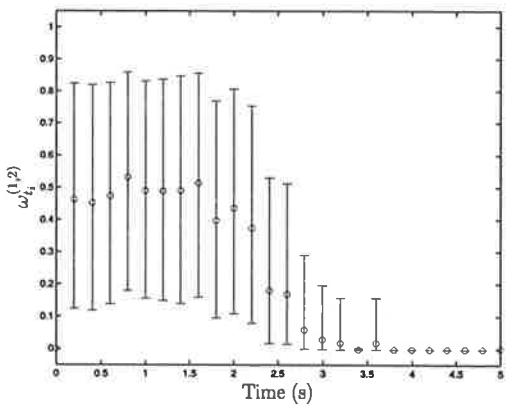


(a) msPMHT ($\omega_{t_i}^{(1,1)}$)

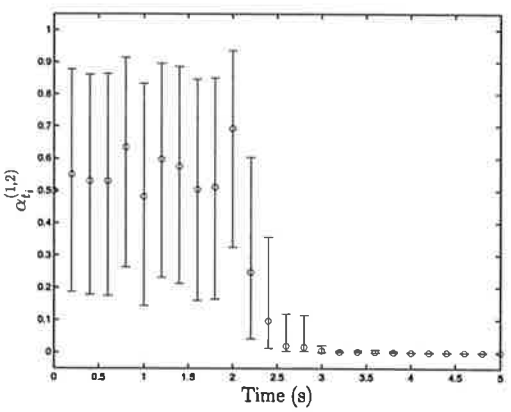


(b) msPLST ($\alpha_{t_i}^{(1,1)}$)

Figure 8.23: Measurement to target assignments of target 1 and sensor 1 for diverging targets using dissimilar sensors

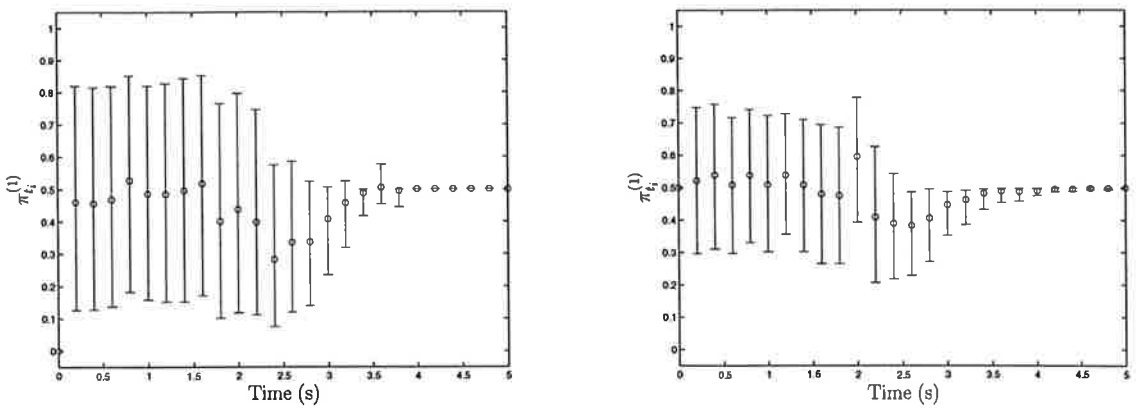


(a) msPMHT ($\omega_{t_i}^{(1,2)}$)



(b) msPLST ($\alpha_{t_i}^{(1,2)}$)

Figure 8.24: Measurement to target assignments of target 1 and sensor 2 for diverging targets using dissimilar sensors



(a) msPMHT (b) msPLST

Figure 8.25: Target 1 measurement probabilities for diverging targets using dissimilar sensors

8.5 Evaluation Using Real Data

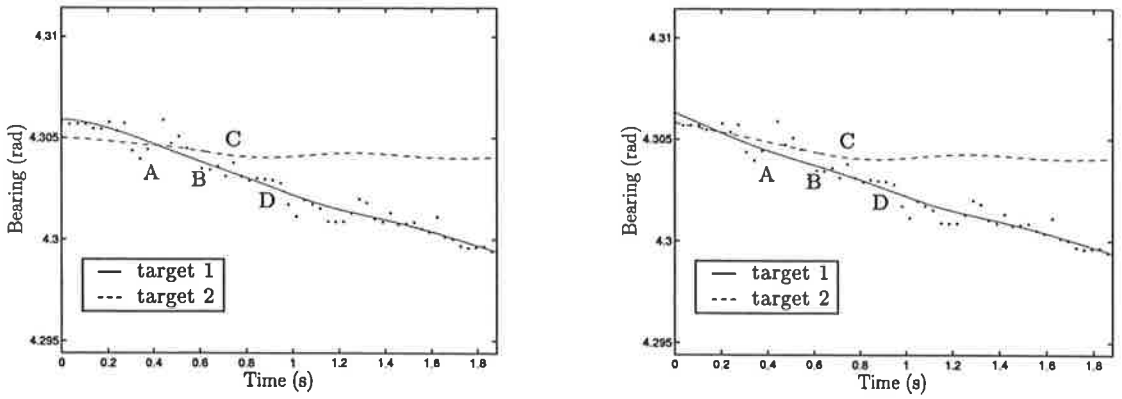
The following examples use real radar and optical measurements collected using the test-bed described in appendix A. The measurement noise covariance of the radar measurements is generally approximately ten times that of the optical measurements. The measurement update rates of the two sensors are such that simultaneous measurements are rare. Since each measurement scan contains only the one measurement, the target measurement probabilities, i.e., the $\pi_{t_i}^{(m)}$'s, are simply the appropriate target assignment probabilities, and therefore they are not presented for these examples. As there are only two targets in these examples, the target assignment probabilities for target 2 are simply one less the target assignment probability for target 1, and are also not explicitly presented.

For clarity, the target tracks obtained from each tracking algorithm are presented twice, once with only the radar measurements and once with only the optical measurements. This is necessary because the high density of measurements tends to obscure the detail if they are combined.

Example 1

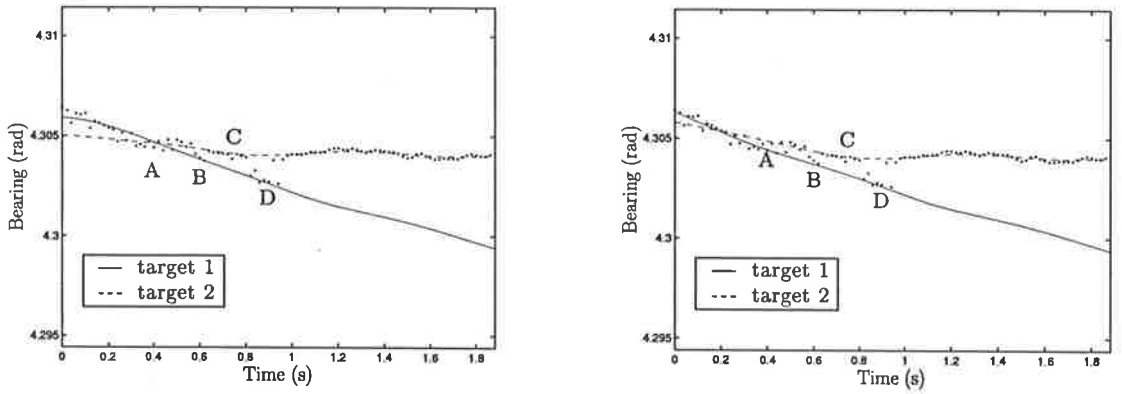
Figure 8.26 shows the tracks with the radar measurements that were obtained from the msPMHT and the msPLST algorithms for crossing targets. The corresponding tracks with the optical measurements are shown in figure 8.27.

Initially the radar and optical sensors both follow target 1, whose track is indicated as a solid line in figures 8.26 and 8.27. At approximately 0.3 s, target 2 (shown as a dashed line) begins to seduce the optical sensor, as shown by the optical measurements at 'A'. The seduction is not perfect, with the optical measurements again originating from target



(a) msPMHT (b) msPLST

Figure 8.26: Tracks from real crossing targets showing radar measurements



(a) msPMHT (b) msPLST

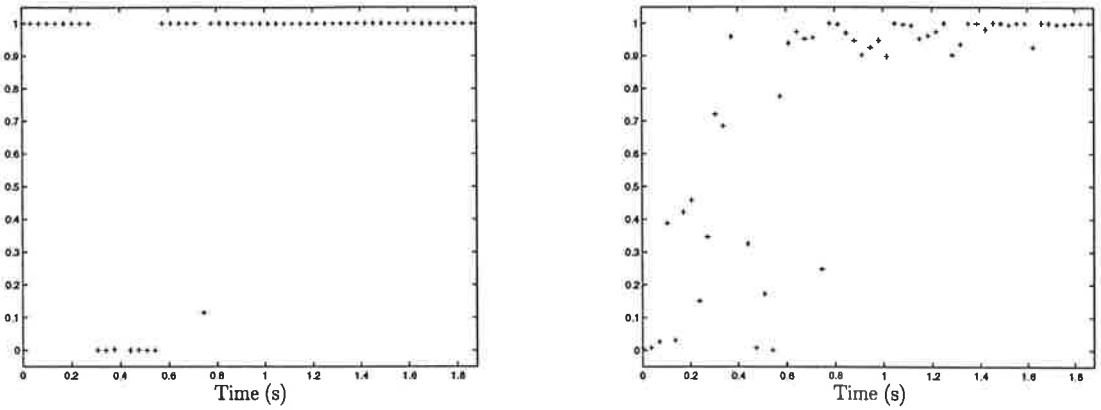
Figure 8.27: Tracks from real crossing targets showing optical measurements

1 at 'B', then target 2 at 'C' and target 1 at 'D', before the seduction is finally complete.[2]

The msPMHT algorithm initially assigns all the measurements to target 1, as shown in figures 8.28(a) and 8.29(a). It also assigns the optical measurements at 'B' and 'D' to target 1, and the optical measurements at 'A' and 'C' to target 2, although it shows some uncertainty with the soft assignments where the targets actually cross ('A'). It is these hard assignments that enable the msPMHT algorithm to determine that the two targets cross, and it uses the dynamic model to estimate the trajectory of target 2 during the initial period when no measurements are assigned to it. Note that the msPMHT incorrectly assigns the radar measurements at 'A' to target 2 (the radar did not produce any measurements from target 2).

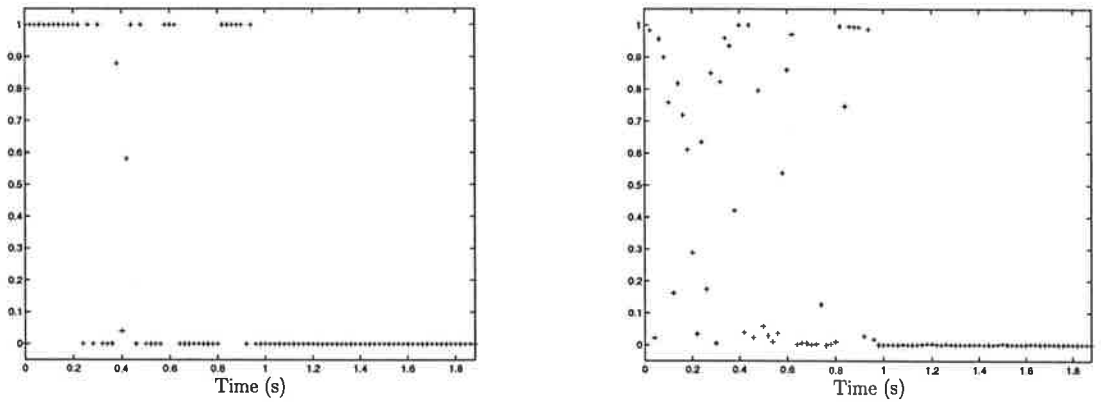
The msPLST algorithm is not certain to which target each measurement in the initial stages is to be assigned. Therefore it produces soft assignments, see figures 8.28(b) and 8.29(b), and during the period before the targets cross, and for some time after, the

[2]This scenario was considered to be the most likely from the video footage taken during data collection.



(a) msPMHT (b) msPLST

Figure 8.28: Measurement to target assignments of target 1 and the radar measurements for real crossing targets



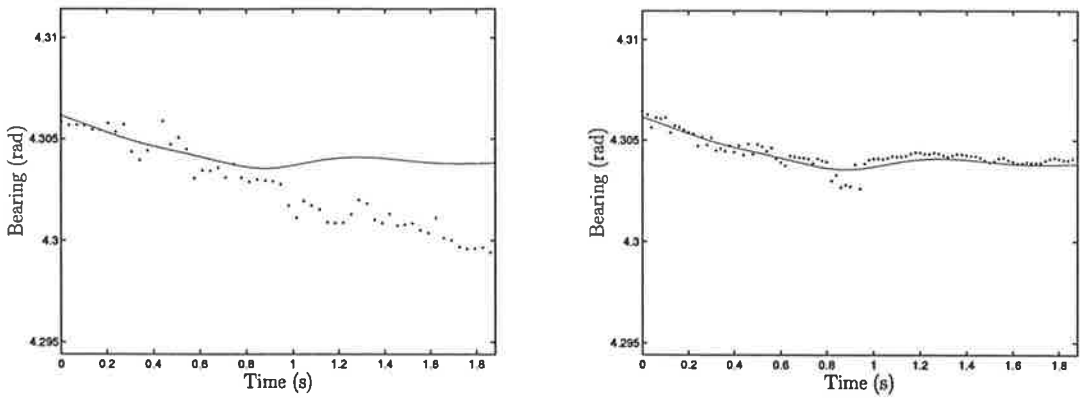
(a) msPMHT (b) msPLST

Figure 8.29: Measurement to target assignments of target 1 and the optical measurements for real crossing targets

measurements are distributed between the two targets. The resulting tracks both follow the measurements, giving the impression of diverging rather than crossing targets. As the targets separate, the algorithm becomes more confident in its assignments, and the tracks become similar to those obtained from the msPMHT.

In this example, the underlying probability distributions allowed the msPMHT algorithm to correctly assign the measurements to the appropriate targets (except where they actually cross), and then produce accurate target tracks, even with the initial absence of measurements from the second target. The msPLST does not assume any underlying probability distributions, relying on minimising prediction errors. As a result, with no model of expected behaviour, it was unable to make confident assignments and, in this case, the resulting tracks did not reflect the true situation.

The performance of the msPMHT and msPLST algorithms was compared to that of



(a) Radar measurements

(b) Optical measurements

Figure 8.30: Fixed interval Kalman smoother track from radar and optical measurements from real crossing targets

a fixed interval Kalman smoother[3] using all the measurements from both sensors.

Figures 8.30(a) and 8.30(b) show the track obtained from the Kalman smoother, with the radar and optical measurements respectively. The Kalman smoother assumes a single target, and that all measurements are suitable for tracking this target, i.e., it does not perform any data association. As can be seen from figure 8.30, the target track from the Kalman smoother follows the optical measurements, because the optical measurements have a significantly lower noise covariance than the radar measurements. Therefore the Kalman smoother applies a higher gain to the optical measurements, and they subsequently contribute more to the estimation process than the radar measurements.[4]

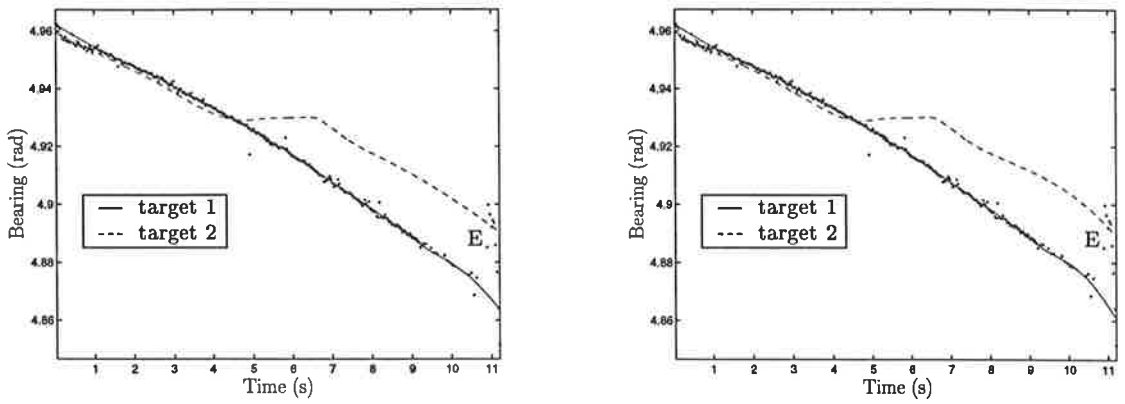
The track obtained from the Kalman smoother is similar to that of target 2 from the msPLST (figure 8.27(b)). Therefore it would appear that similar results could be achieved using a separate Kalman smoother for each of the radar and optical measurements, eliminating the need for data association. This would indeed be true for this example if the seduction of the optical sensor was perfect, i.e., all optical measurements up the time that the targets cross were from target 1, and all the remainder from target 2. However, as shown in figure 8.27, this is not the case, hence the need for data association. Also, in dense multi-target environments, individual sensors may produce multiple simultaneous measurements from different targets.

Example 2

In a second example illustrated in figures 8.31 and 8.32, the radar follows a single target, target 1, until it is lost in clutter at 'E'. The optical sensor produces a single measure-

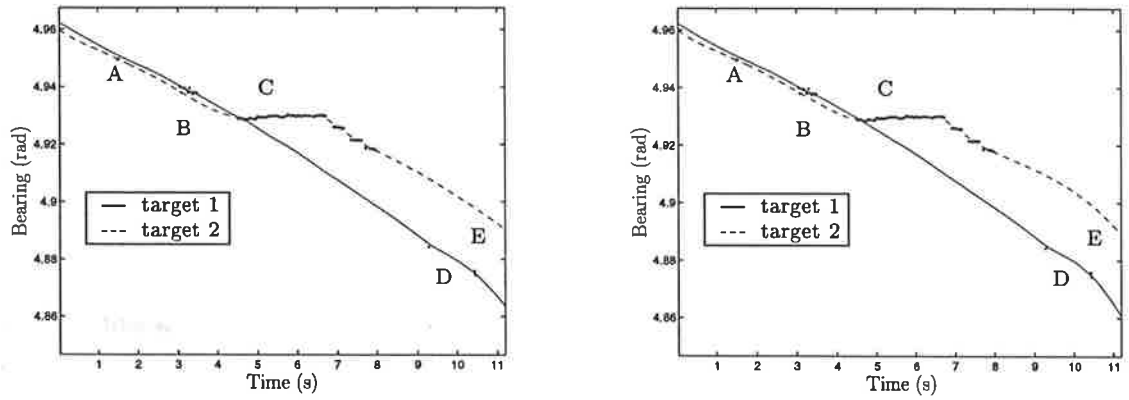
[3]The fixed interval Kalman smoother used here is equivalent to the AFKF (section 3.2) with its state estimates *smoothed* by backward recursion.

[4]The effect of the Kalman smoother gains is similar to that of the AFKF, as discussed in section 3.3.3.2.



(a) msPMHT (b) msPLST

Figure 8.31: Target tracks from a real target in clutter showing radar measurements

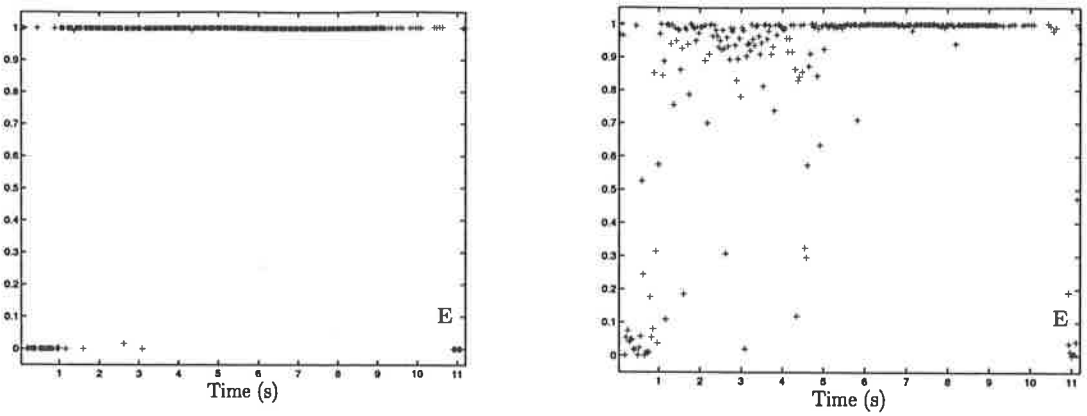


(a) msPMHT (b) msPLST

Figure 8.32: Target tracks from a real target in clutter showing optical measurements

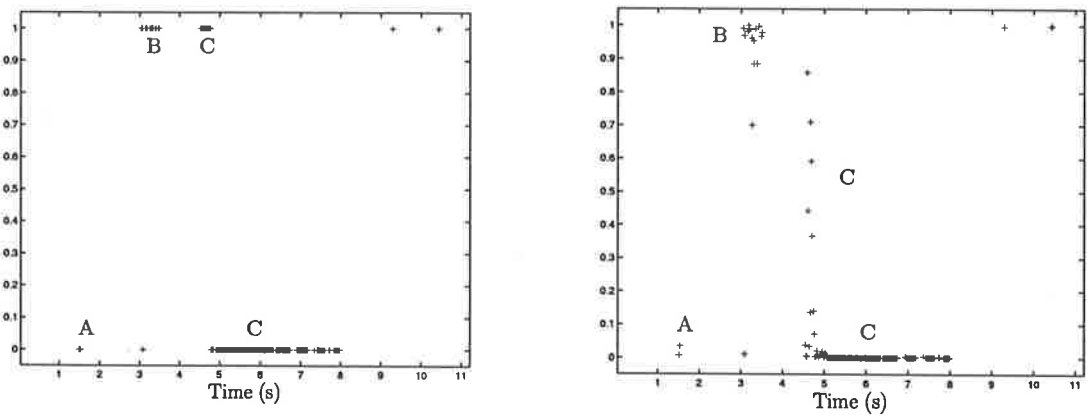
ment at 'A', and a small burst of measurements at 'B', before locking onto a stationary background object at 'C'. Once this leaves the field of view, several other clutter objects are detected, and then the optical measurements cease except for isolated measurements around 'D', assumed to be from the target.

Initially, both algorithms assign most of the radar measurements to target 2, as shown in figure 8.33. Then both algorithms assign most of the radar measurements to target 1, particularly the msPMHT which makes harder assignments than the msPLST. Both algorithms assign the optical measurements at 'A' to target 2, and most those at 'B', to target 1. The initial optical measurements at 'C' are assigned by the msPMHT to target 1 (figure 8.34(a)) and, as they diverge from target 1, the rest are assigned to target 2, whereas the msPLST tends to assign most of the measurements at 'C' to target 2 (figure 8.34(b)). Careful examination of these initial measurements indicates that they could be assigned to either target. The algorithms continue to assign the radar measurements to target 1 and the optical measurements to target 2. After approximately eight seconds,



(a) msPMHT (b) msPLST

Figure 8.33: Measurement to target assignments of target 1 and radar measurements for a real target in clutter



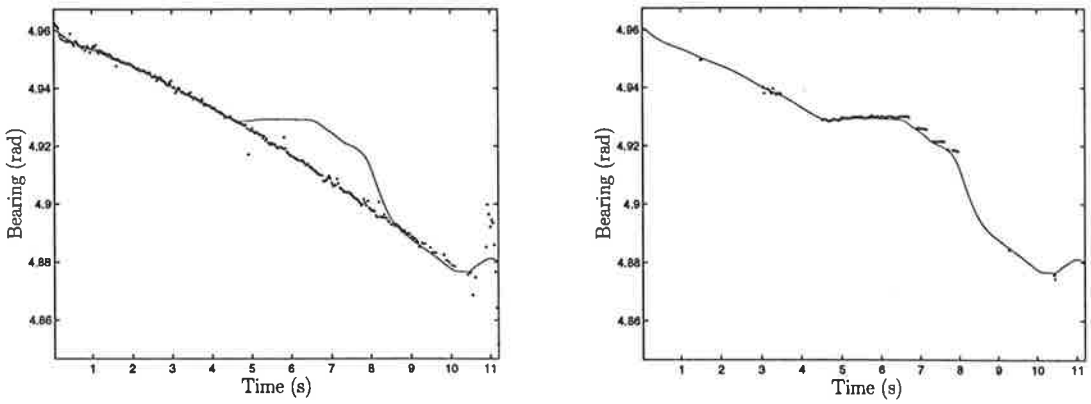
(a) msPMHT (b) msPLST

Figure 8.34: Measurement to target assignments of target 1 and optical measurements for a real target in clutter

no further measurements are assigned to target 2, except for some radar clutter measurements at eleven seconds, indicating that it is probably not a genuine target. The optical measurements at 'D' are assigned to target 1 by both algorithms, as this is their most likely source. During the period of radar clutter at 'E', both algorithms assign most measurements to target 2, with the tracks for target 1 continuing using the assumed target dynamics.

In this example, both algorithms produced similar assignments and tracks, although again the msPMHT was more confident in its assignments. The ability of the algorithms to maintain track in a region of optical clutter, and through a short period of radar clutter, is demonstrated by this example. Both algorithms show promise in this application, but further more detailed evaluation is advised.

A fixed interval Kalman smoother was applied to the data (figure 8.35), and again it preferred to follow the optical measurements when they were present. However, in



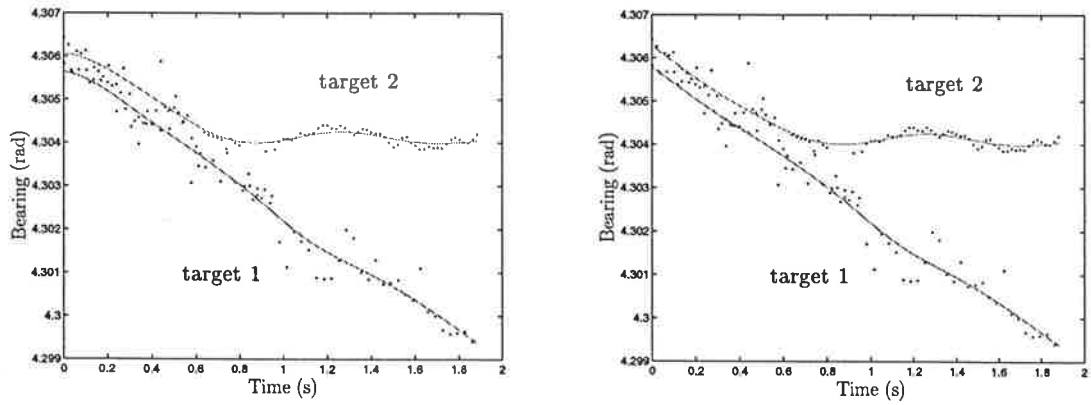
(a) Radar measurements (b) Optical measurements

Figure 8.35: Fixed interval Kalman smoother track from radar and optical measurements from a real target in clutter

this example, the optical measurements arose from clutter, and not from a legitimate target, and consequently the Kalman smoother did not produce a valid target track. This illustrates the unacceptable performance of the fixed interval Kalman smoother, caused by its inability to select only those measurements suitable for tracking, i.e., those originating from the target of interest. Therefore, as for the AFKF (section 3.3.3), it is easily seduced by clutter and other targets.

Example 3

The msPMHT and msPLST tracks in figure 8.36 illustrate a problem arising from the misalignment of sensors, in this case a radar and optical sensor. In this example, target 1 moves in the direction of decreasing bearing, and initially both sensors are following



(a) msPMHT (b) msPLST

Figure 8.36: Target tracks with misaligned sensors

this target. Optical measurements from target 2 only appear after 0.6 s, and this target maintains an almost constant bearing.

The misalignment of the sensors causes the radar and optical measurements from target 1 to be displaced from each other. Therefore, during the first 0.6 s, they appear as two closely spaced targets instead of one, as shown by the target tracks from both the msPMHT and msPLST algorithms in figure 8.36. Therefore sensor misalignment has the potential to produce false target tracks in dense multi-target environments.

8.6 Modelling Errors

The msPMHT algorithm estimates the parameters of the underlying probability distributions, therefore it would be expected that it would be susceptible to model mismatch. On the other hand, the msPLST does not depend on the underlying distributions, except that it uses the covariances as scaling factors. Therefore it should only be susceptible to errors in the covariance values. This section evaluates the performance of both algorithms under the assumption of Gaussian distributions with errors in the assumed measurement noise covariances.

Consider the simple two target problem for one dimensional measurements, and assume that the measurement noise covariances are independent of target model, i.e., $\mathbf{R}_{t_i}^{(1,s)} = \mathbf{R}_{t_i}^{(2,s)} \triangleq \mathbf{R}_{t_i}^{(s)}$, and that all measurements are equally likely to originate from either target, i.e., $\pi_{t_i}^{(1)} = \pi_{t_i}^{(2)} = 0.5$. Then the msPMHT target assignment probability (6.50) for the measurement $\mathbf{z}_{t_i}^{(s)}$ and target 1 is given simply as

$$\omega_{t_i}^{(1,s)} = \left(1 + \exp\left(\frac{1}{2}\mathbf{R}_{t_i}^{(s)\,-1}\left(\boldsymbol{\epsilon}_{t_i}^{(1,s)\,2} - \boldsymbol{\epsilon}_{t_i}^{(2,s)\,2}\right)\right)\right)^{-1}. \tag{8.5}$$

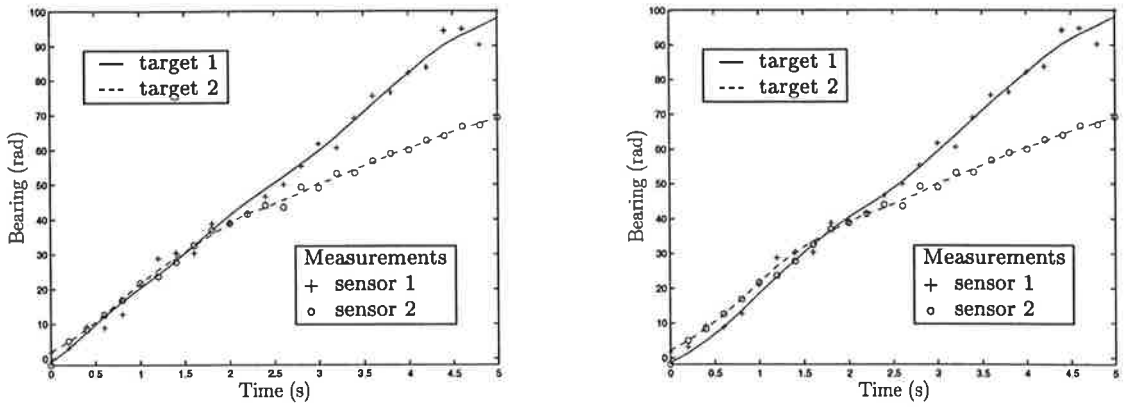
It is obvious that this function is dependent on the covariance of the measurement noise. The corresponding msPLST target assignment weight (7.33) is simplified to

$$\alpha_{t_i}^{(1,s)} = \left(1 + \frac{\boldsymbol{\epsilon}_{t_i}^{(1,s)\,2}}{\boldsymbol{\epsilon}_{t_i}^{(2,s)\,2}}\right)^{-1} \tag{8.6}$$

which is independent of the measurement noise covariance.

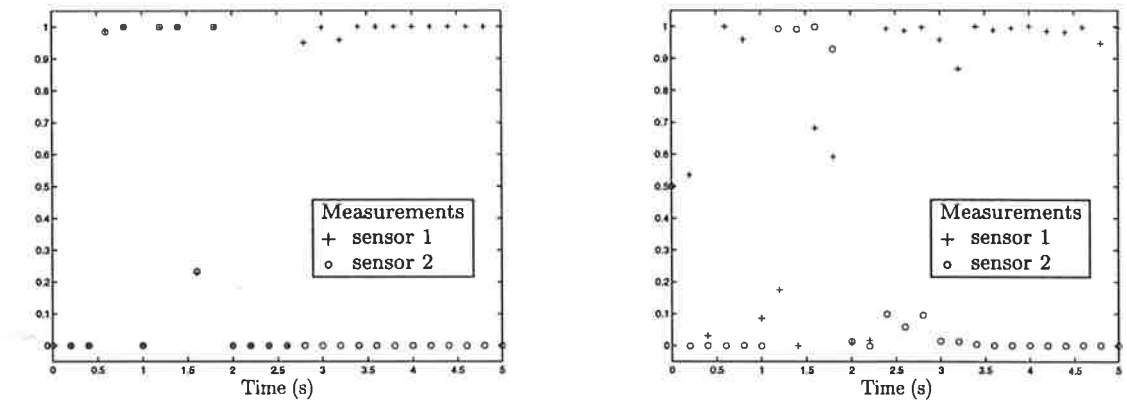
The track error covariance of the target state estimates for each algorithm is dependent on the selected values of measurement noise covariance and process noise covariance, because this estimation is performed by a Kalman smoother. The difference in the TEC of the msPMHT and msPLST compared to that of the fixed interval Kalman smoother is in the covariance of the composite measurements, which is dependent on the target assignments.

Both algorithms estimate the target states with a fixed interval Kalman smoother, and this uses the measurement noise covariances in the estimation process. Any errors in



(a) msPMHT (b) msPLST

Figure 8.37: Target tracks under matched conditions



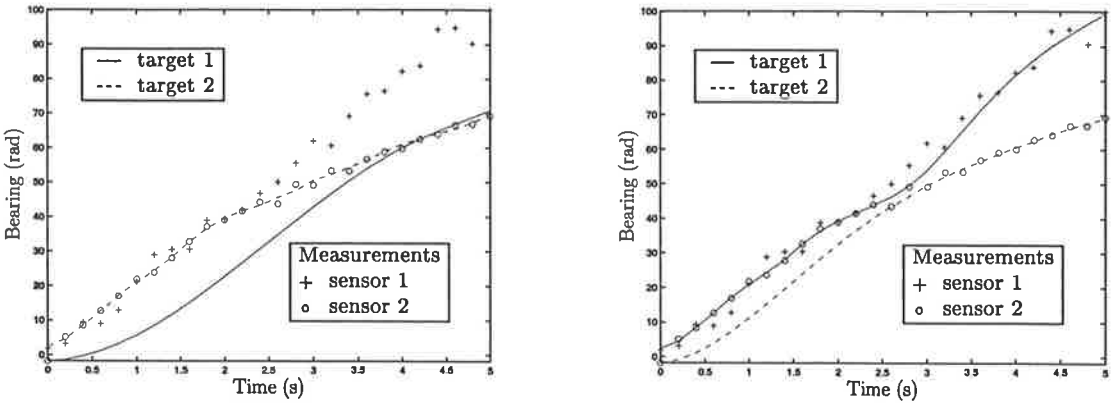
(a) msPMHT (b) msPLST

Figure 8.38: Measurement to target 1 assignments under matched conditions

these covariances will cause an increase in the TEC, hence both algorithms will suffer a degradation in performance from any error in the measurement noise covariances.

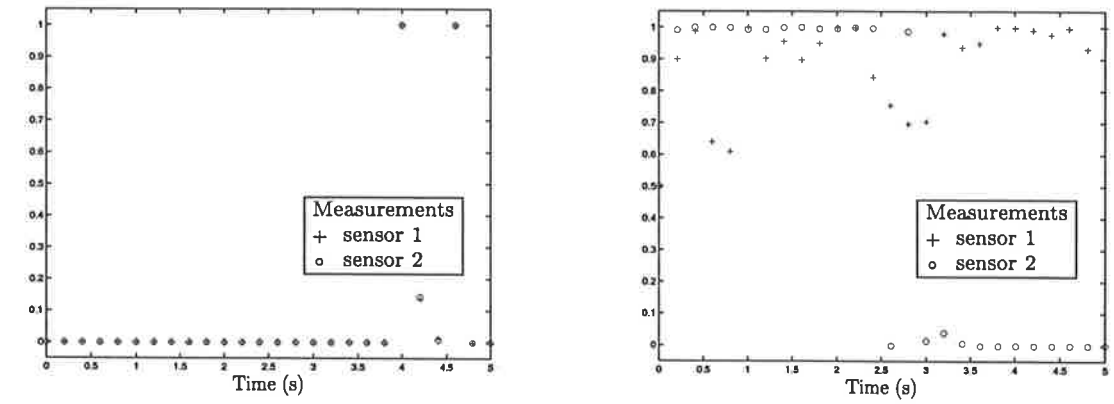
The diverging target example was used to compare the operation of both algorithms under these error conditions. Figure 8.37 shows the tracks obtained from each when the measurement noise covariances are matched to the data. The target assignments for target 1 are shown in figure 8.38. In this example, target 1 is moving in a direction of increasing bearing (azimuth). At two seconds, target 2 crosses target 1, and sensor 2, whose measurements are denoted by o's, is seduced by target 2 and commences to produce measurements from it. Note that sensor 1, whose measurements are indicated by +'s, has a measurement noise covariance ten times that of sensor 2.

It can be seen from figure 8.37 that both algorithms produce similar, but not identical, target tracks, the difference being highlighted by the target assignments as illustrated in figure 8.38. Both algorithms initially assign measurements from both sensors to each target, and after the tracks diverge, assign the measurements from sensor 1 to target 1



(a) msPMHT (b) msPLST

Figure 8.39: Target tracks with $\mathbf{R}_{t_i}^{(1)}$ overestimated



(a) msPMHT (b) msPLST

Figure 8.40: Measurement to target 1 assignments with $\mathbf{R}_{t_i}^{(1)}$ overestimated

and the measurements from sensor 2 to target 2.

This section deals primarily with the differences in the assignments of both algorithms. The effect of the assignments on the composite measurement covariance and TEC was discussed in section 8.1.3.

Overestimation of Covariance in Sensor 1

To evaluate the effect of measurement model errors, the assumed noise covariance on sensor 1 in both the msPMHT and msPLST was increased by a factor of ten, i.e., the algorithms overestimate this noise covariance by a factor of ten. Under these conditions, the msPLST tracks still follow the appropriate targets after divergence (figure 8.39(b)), with some minor changes in track error due to the effect of the mismatch on the Kalman smoother. However, before divergence, instead of assigning measurements to both targets (figure 8.40(b)), the msPLST now assigns all measurements to target 1, and therefore

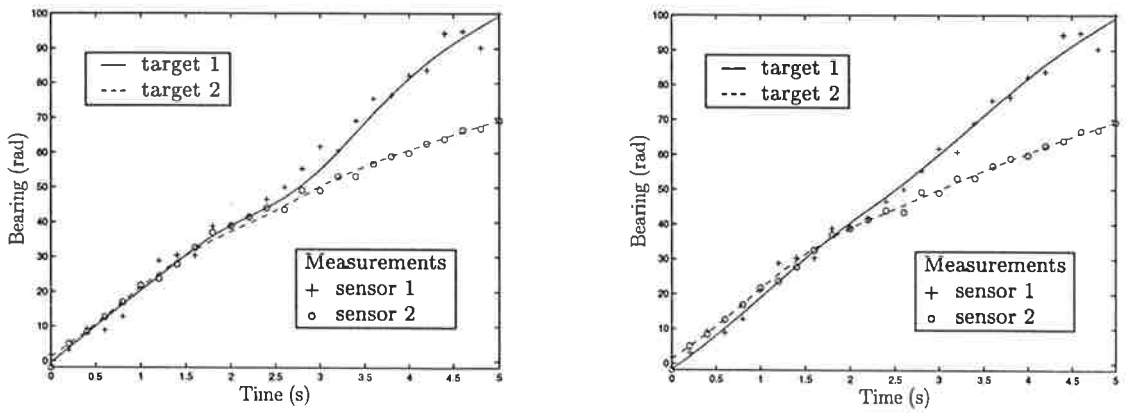
target 2 deviates from the measurements. The target state estimates are affected by the errors in the measurement noise covariance, and this in turn affects the estimated target assignments, i.e., the assignments change as the tracks shift. This effect produces the observed change in the target assignments.

The msPMHT tracks, as shown in figure 8.39(a), differ significantly from those obtained under matched conditions, with one target track now following target 2, and the other following nothing in particular. The algorithm assigns few measurements from either sensor to target 1 (figure 8.40(a)), providing little correction to the track and allowing it to be determined primarily by the target's dynamic model. This occurs because the msPMHT believes that the measurements from sensor 1 are now 100 times (not ten times as previously) noisier than those from sensor 2, and it now assumes that these measurements belong to target 2 and no other valid target is present.

Overestimation of Covariance in Both Sensors

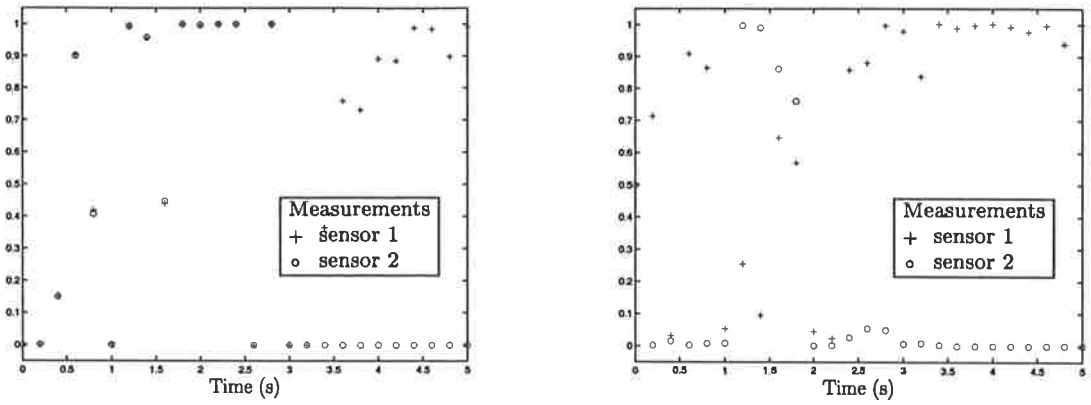
The tracks in figure 8.41 were obtained after increasing the assumed measurement noise covariance of both sensors by a factor of ten over that of the matched conditions. The msPLST results, shown in figures 8.41(b) and 8.42(b), are similar to those under matched conditions, because the ratio of Kalman smoother gains for the radar and optical measurements is similar to those under matched conditions.

The msPMHT tracks (figure 8.41(a)) are again different to those obtained under matched conditions. Here the track following the measurements from sensor 1 follows the measurements from sensor 2 immediately after the targets diverge. During this time, the algorithm cannot discriminate between the targets because of the extra assumed noise. However, the weighting on the measurements from sensor 2 in the Kalman smoother is not as high as when only the covariance of sensor 1 was increased, and therefore the track of target 1 is able to return to the measurements from sensor 1 as the targets separate. Fig-



(a) msPMHT (b) msPLST

Figure 8.41: Target tracks with both $\mathbf{R}_{t_i}^{(1)}$ and $\mathbf{R}_{t_i}^{(2)}$ overestimated



(a) msPMHT

(b) msPLST

Figure 8.42: Measurement to target 1 assignments with both $\mathbf{R}_{t_i}^{(1)}$ and $\mathbf{R}_{t_i}^{(2)}$ overestimated

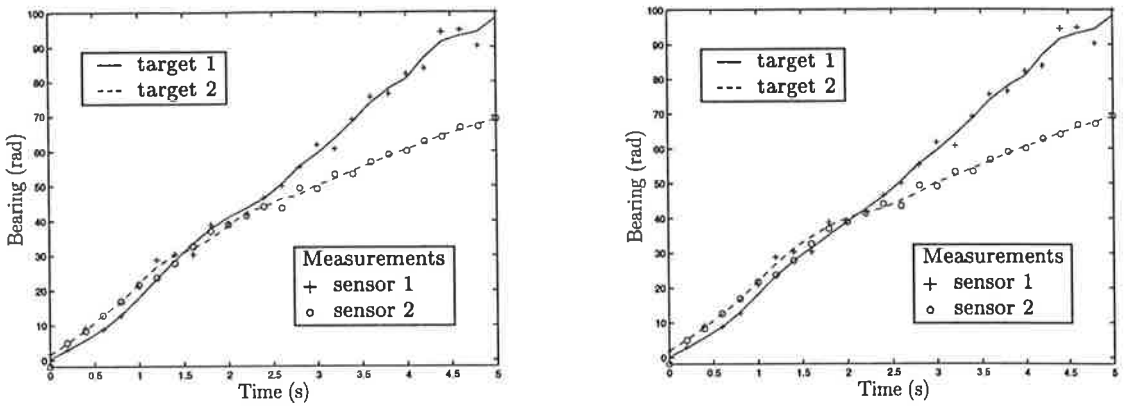
ure 8.42(a) shows how the measurements from both sensors are assigned to both targets before and immediately after divergence. The increased measurement noise covariance creates greater uncertainty in the assignments of both algorithms. As the measurements separate further, the measurements from sensor 2 are assigned to target 2, and gradually the nosier measurements from sensor 1 are assigned to target 1.

Careful observation of the target tracks from both algorithms shows that as the measurement noise covariance of the measurements from a sensor increases, the tracks appear straighter, indicating that the Kalman smoother decreases the significance of the measurements, relying more on the dynamic model. This behaviour is expected from the Kalman smoother.

The above results have shown that increasing the assumed measurement covariance of one or more sensors reduces the ability of the msPMHT to discriminate between closely spaced targets, and subsequently reduces its ability to identify all targets. The msPLST did not suffer from this problem to the same extent as the msPMHT.

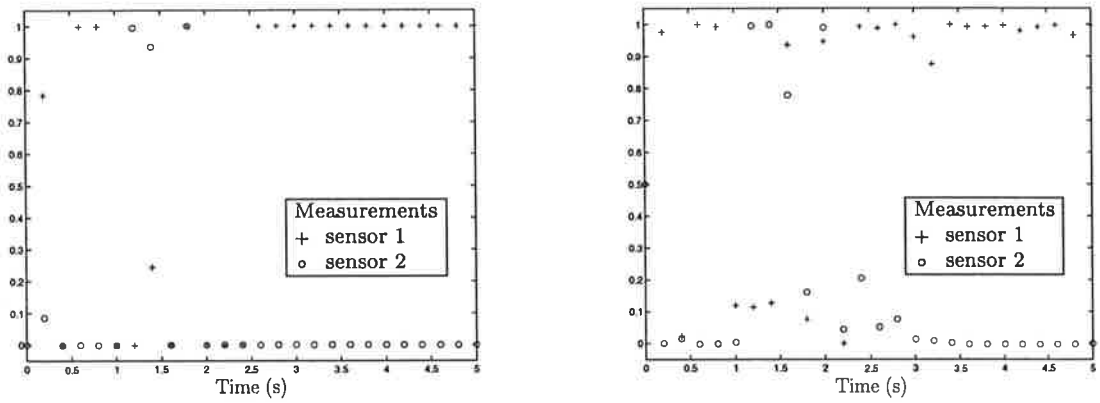
Underestimation of Covariance in Sensor 1

The effect of selecting assumed measurement covariances less than that of the data was also investigated. With the measurement noise covariance of sensor 1 reduced by a factor of ten from its matched value and sensor 2 remaining matched, the tracks of target 1 from both algorithms (figure 8.43) fluctuate more than they did under matched conditions. This indicates that they tend to follow the measurements more closely, as would be expected from a Kalman smoother with a lower measurement noise covariance. The assignments from both algorithms (figure 8.44) are similar to those under matched conditions, because the lower noise covariance allows the measurements to be assigned to both targets. Therefore the main symptom of this measurement covariance mismatch is the increased TEC, and this occurs in both algorithms.



(a) msPMHT (b) msPLST

Figure 8.43: Target tracks with $\mathbf{R}_{t_i}^{(1)}$ underestimated



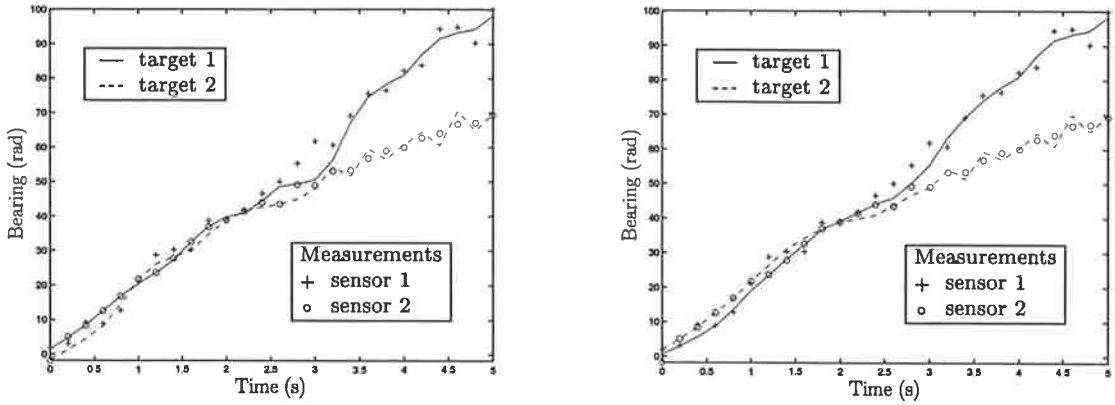
(a) msPMHT (b) msPLST

Figure 8.44: Measurement to target 1 assignments with $\mathbf{R}_{t_i}^{(1)}$ underestimated

Underestimation of Covariance in Both Sensors

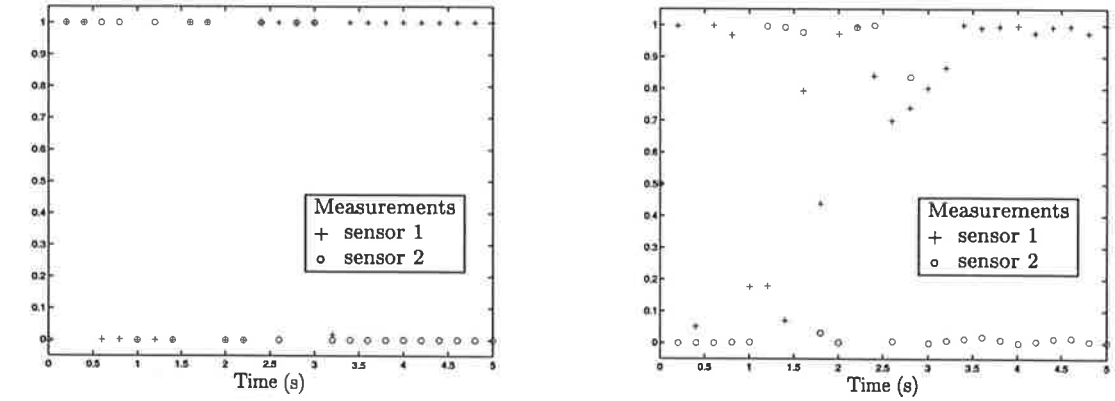
The measurement noise covariance of both sensors was reduced to a tenth of their matched values, and the resulting target tracks are shown in figure 8.45. The lower value of assumed measurement noise covariance causes both tracks from both algorithms to follow the individual measurements more closely, as illustrated by the increased track error in figure 8.45. The msPMHT track of target 1 follows the measurements from sensor 2 immediately after the targets diverge. This also occurs in the msPLST, but to a lesser extent because the msPLST assignments at this time (figure 8.46) are softer than those of the msPMHT.

For values of assumed measurement covariance that are less than that of the data, the target tracks have a greater tendency to follow the individual measurements, causing an increase in TEC. Also, both targets will initially tend to follow the measurements from the sensor with the lowest noise as they diverge or separate.



(a) msPMHT (b) msPLST

Figure 8.45: Target tracks with both $\mathbf{R}_{t_i}^{(1)}$ and $\mathbf{R}_{t_i}^{(2)}$ underestimated



(a) msPMHT (b) msPLST

Figure 8.46: Measurement to target 1 assignments with both $\mathbf{R}_{t_i}^{(1)}$ and $\mathbf{R}_{t_i}^{(2)}$ underestimated

8.7 Summary of Algorithm Evaluation

The linear Gaussian msPMHT and msPLST algorithms have been compared. Both produce *a posteriori* estimates of the target states and soft measurement to target assignments.

The msPMHT log likelihood and msPLST cost functions are similar, the main difference being the absence of terms associated with the probability mass function of the target assignments in the msPLST. This difference manifests itself in both the estimation of the the target assignments, and how the assignments influence the composite measurement and covariance used in the estimation of the target states.

The target assignment estimates of the msPMHT tend to be harder, i.e., closer to one or zero, than those of the msPLST when the targets are well separated, although this difference decreases as the target separation increases. When the targets are close, the msPMHT still tends to make harder assignments, whereas the msPLST is unable to

decide to which target to assign each measurement. These assignments affect the track error covariance through the calculation of the covariance of the composite measurements. When the targets are well separated, the composite measurement covariance, and therefore the TEC, is less than that of the msPLST, because the target assignments are harder in the msPMHT. For close targets, if most of the measurements are assigned to a particular target, the msPMHT's composite measurement covariance and TEC will be lower than that of the msPLST for that target. The converse applies if few measurements are assigned to the target.

Both algorithms have similar complexity within each iteration of the algorithm, but generally the msPMHT requires more iterations to converge.

Similar tracks were produced by both algorithms on simulated dual sensor measurements from crossing and diverging targets. Monte Carlo simulations showed that the msPMHT produced a wider range of tracks (in general all were practically feasible), particularly when the targets were close, i.e., the msPMHT was more dependent on the actual measurements than was the msPLST. Evaluation on real data showed that both algorithms produce satisfactory tracks, although in the case of crossing targets where one is not detected until it approaches the other, the msPMHT's performance was superior.

The msPMHT is more susceptible to errors in the measurement noise covariance than the msPLST when determining its target assignments, because it relies on the underlying pdf's. In particular, the msPMHT experiences difficulty discriminating between closely spaced targets if the measurement noise covariance is overestimated. Both algorithms experienced sub-optimal state estimation with incorrect measurement covariances, i.e., increased track error covariance. This was particularly evident when the measurement covariance was underestimated.

CHAPTER 9

Summary

The application of state space techniques for tracking using measurements from multiple sensors has been investigated from a track fusion and measurement association perspective. The knowledge of which measurements belong to a particular target, and can therefore be used for tracking, is usually not available. Solving this data association problem, in particular for measurements from multiple sensors, has been addressed.

The key contributions of this thesis have been the application of the asynchronous fused Kalman filter (AFKF) to tracking using real radar and optical data, and the development and evaluation of the multi-sensor Probabilistic Multi-Hypothesis Tracking (msPMHT) and multi-sensor Probabilistic Least Squares Tracking (msPLST) algorithms for tracking multiple targets using measurements from multiple sensors.

9.1 Summary of the AFKF

The AFKF is a variable update rate Kalman filter with a time variant measurement model that is determined by the sensor from which the measurement (or measurements) are received.

Its performance has been evaluated using both real and simulated data. The simulated data provided insight into the operation of the AFKF, and illustrated the potential improvement in tracking performance that can be achieved by using multiple sensors.

It was the application of the AFKF to real data collected from a radar and optical sensor that was of most interest. When tracking a single target in the absence of other targets and noise (clutter), the AFKF performed much as expected from the simulations. However, the optical sensor is passive and does not provide any range information. As a consequence, it receives measurements from all sources within its two-dimensional field of view over its entire detection range. Other targets with similar angular position, but separated in range, often seduced the optical sensor and, because the optical measurements were not as noisy as those from the radar, the AFKF often lost the target of interest and tracked the new interfering target. Similar problems arose when tracking in optical clutter.

This provided the motivation for developing new algorithms that simultaneously solve the data association and target state estimation problems.

The problem of model sensitivity in a dual sensor Kalman filter was investigated, and its tracking performance was compared to that of a single sensor Kalman filter. Errors in the assumed process noise covariance produced almost identical changes in the track error covariance of both Kalman filters. Mismatch in the measurement noise covariance also produced similar results in both filters if the ratio of assumed to actual measurement noise covariance (mismatch ratio) of the sensor common to both filters (original sensor) was substantially smaller than in the other (additional) sensor. However, if the mismatch ratio in the original sensor was larger than in the additional, the dual sensor filter was more sensitive than the single sensor filter to errors in the measurement noise covariance. If the mismatch ratio in the original sensor was equal to or marginally less than that of the additional sensor, the dual sensor filter was less sensitive than the single to measurement noise covariance errors.

9.2 Summary of msPMHT and msPLST Algorithms

Two multi-sensor multi-target tracking algorithms, the multi-sensor Probabilistic Multi-Hypothesis Tracking (msPMHT) and multi-sensor Probabilistic Least Squares Tracking (msPLST) algorithms, have been developed. The performance of each has been evaluated using simulated and real data, and comparisons between the two have been made.

Both algorithms exhibit the following similarities.

- Batch algorithms, operating on an entire batch of measurements.
- Introduce soft or probabilistic measurement to target assignments.
- Estimate the measurement to target assignments and target states simultaneously.
- Iterate until the likelihood or cost function converges. Each algorithm has similar complexity within each iteration.
- Converge to the nearest local maximum or minimum and therefore require careful initialisation.
- Limited to a predetermined maximum number of targets.
- Employ an extended observer containing the measurement to target assignments and the target states.
- Separate target models for each target, each comprising of a dynamic model and a separate measurement model for each sensor.

The following differences between the algorithms are noted.

- The msPMHT is a maximum *a posteriori* estimator, and the msPLST uses a minimum mean squared error technique.
- The msPMHT relies upon the observer probability functions, while the msPLST operates on squared error terms.
- Generally the msPMHT takes more iterations to reach convergence. The likelihood maxima are broader, and probably overlap more, than the msPLST minima.
- The msPMHT is more prone to model errors.
- The msPMHT measurement to target assignments tend to be *harder*.

Both algorithms produced acceptable results in multi-sensor multi-target scenarios, as indicated by testing on simulated and real data.

An extension of the msPMHT, the general msPMHT, allows multiple measurement models for each sensor. The measurements are assigned to these models by soft measurement to sensor assignments that are estimated along with the target assignments and states.

9.3 Future Research

Research in this area is by no means complete. Some possible areas of future research include the following.

Non-linear (Extended) AFKF

Each tracking coordinate, e.g., azimuth, elevation and range, was assumed to be independent of the others in the AFKF. The use of non-linear dynamic models, or Cartesian state variables with non-linear measurement models, will introduce the inter-dimensional dependencies into the target models, and this may improve the tracking performance under some conditions.

Recursive versions of the msPMHT and msPLST algorithms

Currently, both the msPMHT and msPLST are batch algorithms. In many tracking applications, a recursive solution where the algorithm is updated on receipt of a new measurement would be advantageous. Recursive versions of the EM and least squares algorithms could possibly be incorporated into the algorithms.

Analysis of msPMHT and msPLST in clutter

The evaluation in this thesis has concentrated on tracking in the presence of closely spaced targets. Evaluation of the algorithms in clutter, and the selection of suitable clutter models, provide the opportunity for further research.

Soft dynamic model assignments

Soft sensor assignments have been introduced in the general msPMHT for associating measurements to the sensor model that is most appropriate for the current operating conditions. A similar technique to assign the most appropriate dynamic model to a target may be useful, particularly for manoeuvring targets. This may have significant parallels with Interacting Multiple Model (IMM) filters.

Variable number of targets

The msPMHT and msPLST algorithms both assume a fixed maximum number of targets. Provision to change this value to the number of targets within the environment could be of interest.

Evaluation using non-linear systems and non-Gaussian distributions

Evaluation of the msPMHT and msPLST algorithms to date has assumed linear systems with Gaussian noise statistics. Their performance with non-linear dynamic or measurement models and non-Gaussian noise distributions, such as Ricean for radar, may be of interest in some applications.

9.4 Conclusions

This thesis has investigated the performance of the asynchronous fused Kalman filter for multi-sensor tracking. Although its performance is satisfactory under near ideal conditions, evaluation on real data has shown a strong tendency to lose track in the presence of other targets or clutter.

The msPMHT and msPLST algorithms were developed to overcome the data association problem when tracking in multi-sensor multi-target environments. Both performed successfully, particularly with crossing and diverging targets.

The general msPMHT algorithm has been developed to provide multiple sensor models for each sensor within each target model. The algorithm chooses the most appropriate for each measurement, allowing the algorithm to adapt to changing sensor characteristics.

APPENDIX A

Sensor Fusion Testbed

A multi-sensor testbed has been developed by the Tactical Surveillance Systems Division (TSSD), formerly the Microwave Radar Division (MRD), of the Defence Science and Technology Organisation (DSTO), Australia. It was built to collect track data from multiple sensors and to assist in the development and evaluation of multi-sensor tracking algorithms. It resides in the trailer housing the TSSD's *Generic Pulse Doppler Radar* (GPDR) and has two sensors, the GPDR and an optical tracking system, with provision for adding others. It may be operated on location at remote sites with a portable three phase generator, as shown in figure A.1.

The GPDR is an experimental X-band pulse Doppler radar that was developed by the TSSD. It is an amplitude monopulse radar operating at frequencies within the range

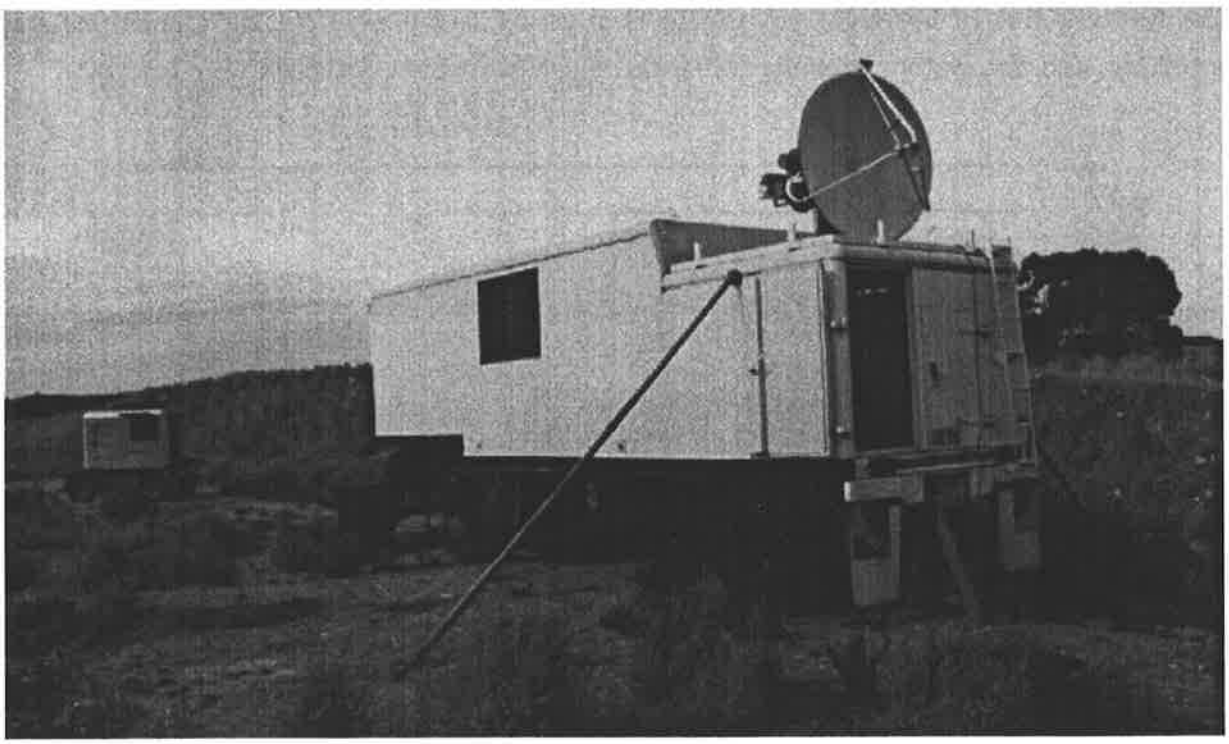


Figure A.1: Sensor fusion testbed operating at a remote site

9–10 GHz. It has a 3 dB beam-width of approximately 1.6° in both azimuth (bearing) and elevation, and at any time the antenna azimuth and elevation positions are known to an accuracy of 0.38 mrad. The radar has eight range cells, each being 1 μs (150 m) wide, and is able to resolve in range to a maximum accuracy of 62.5 ns (9.4 m). The two centre range cells act as a split gate discriminate, providing more precise measurements through interpolation. The four outermost range cells are used to estimate the signal to noise ratio (SNR). Each range cell contains 128 Doppler or frequency bins, with the two bins adjacent to the signal frequency forming a split gate velocity discriminant to measure the target's velocity. The radar provides azimuth, elevation, range and Doppler velocity[1] measurements approximately every 33.3 ms during target tracking. The radar is almost completely software controlled, and most parameters may be configured by the user.

The optical sensor is a colour video camera attached to the radar's antenna mount and positioned directly above the radar's antenna. The camera is fitted with a 75 mm lens through a focal length doubler, giving an effective focal length of 150 mm. This corresponds to a horizontal field of view (FOV) of approximately 2.4° and a vertical FOV of approximately 1.8° . This is comparable to the 3 dB beamwidth of the radar (1.6° in both directions), and the sensors are aligned to maximise their region of overlap. The camera's output images are processed by an Adept20 *automatic video tracker* (AVT) that is housed inside the GPDR's trailer. The AVT extracts the target from the video image and produces azimuth and elevation measurements of the target's position relative to the system's line of boresight. A choice of tracking and pre-processing algorithms are available, and the system is configurable through a software controlled user interface.

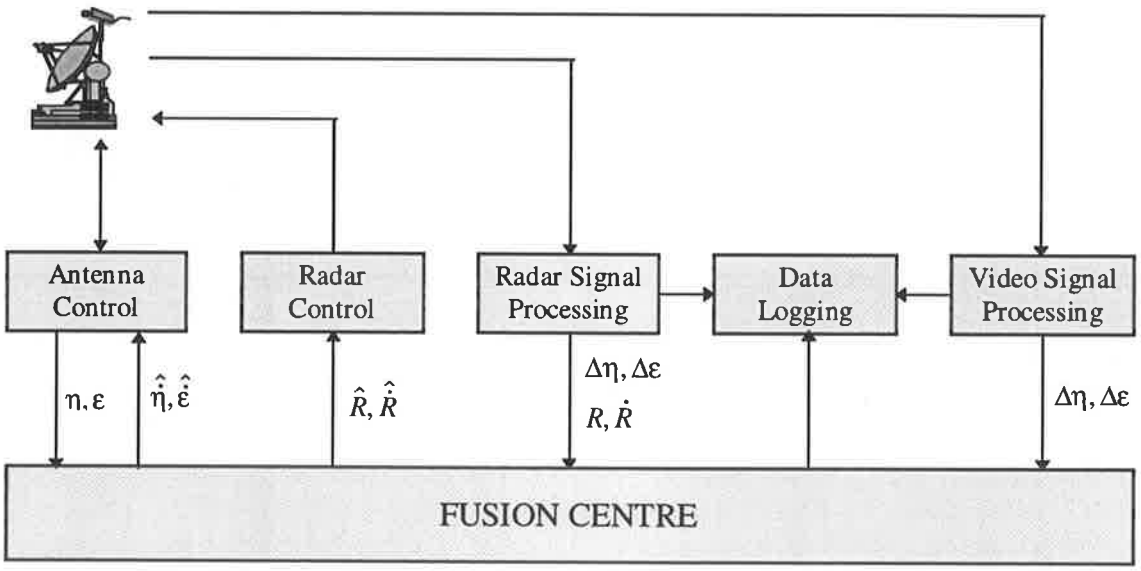


Figure A.2: Sensor fusion testbed block diagram

[1]Doppler velocity is the negative range rate, i.e., the radial velocity of the target towards the radar.

As illustrated in the functional diagram of figure A.2, the radar and optical system are connected to a separate computing platform, designated as the *fusion centre.* Both the radar and optical system provide azimuth and elevation measurements in the form of boresight errors ($\Delta\eta$ and $\Delta\varepsilon$ respectively). The radar also provides range (R) and Doppler velocity ($-\dot{R}$) measurements. To complete the azimuth and elevation measurements, the antenna boresight position (η and ε) is provided by the radar's antenna control. The fusion centre uses these measurements to produce estimates of the state of the target's dynamics. The azimuth and elevation rate estimates ($\hat{\dot{\eta}}$ and $\hat{\dot{\varepsilon}}$) are used to control the antenna mount, and the range and range rate estimates ($\hat{R}$ and $\hat{\dot{R}}$) are used to position the range and velocity gates. Other state parameter estimates, e.g., azimuth ($\hat{\eta}$) and elevation ($\hat{\varepsilon}$), are also available from the fusion centre. A data logging facility is capable of recording radar, optical and fusion centre parameters for post trial evaluation.

A detailed description of the testbed, including hardware specifications, is given in (Krieg 1997).

APPENDIX B

Kalman Smoother Derivation

For linear Gaussian systems, the state estimates for a target may be obtained by solving the tridiagonal system of $T+1$ equations (B.1) corresponding to that particular target. The linear Gaussian version of the msPMHT ((6.38), page 78) and the msPLST ((7.28), page 100) algorithms require this system of equations be solved for each target, i.e., for targets $m = 1, \ldots, M$ where M is the number of targets. In these algorithms, $\tilde{\mathbf{z}}_{t_i}^{(m)}$ and $\tilde{\mathbf{R}}_{t_i}^{(m)}$ represent the composite measurement and its covariance for target m at time t_i. The composite measurement is a weighted sum of all measurements generated at time t_i, where the values of the actual weights are algorithm dependent. The terms in (B.2) are defined in chapter 5 with the exception of the measurement model, in which $\tilde{}$ signifies variables associated with the composite measurement model.[1]

$$
\begin{aligned}
\left(\bar{\Sigma}_{t_0}^{(m)-1} + \mathbf{D}_{t_0}^{(m)}\right)\mathbf{x}_{t_0}^{(m)} - \mathbf{B}_{t_0}^{(m)}\mathbf{x}_{t_1}^{(m)} &= \bar{\Sigma}_{t_0}^{(m)-1}\bar{\mathbf{x}}_{t_0}^{(m)} && (i=0)\\
&\vdots\\
-\mathbf{B}_{t_{i-1}}^{(m)\mathsf{T}}\mathbf{x}_{t_{i-1}}^{(m)} + \left(\mathbf{A}_{t_i}^{(m)} + \mathbf{D}_{t_i}^{(m)}\right)\mathbf{x}_{t_i}^{(m)} - \mathbf{B}_{t_i}^{(m)}\mathbf{x}_{t_{i+1}}^{(m)} &= \mathbf{c}_{t_i}^{(m)} && i=1,2,\ldots,T-1\\
&\vdots\\
-\mathbf{B}_{t_{T-1}}^{(m)\mathsf{T}}\mathbf{x}_{t_{T-1}}^{(m)} + \mathbf{A}_{t_T}^{(m)}\mathbf{x}_{t_T}^{(m)} &= \mathbf{c}_{t_T}^{(m)} && (i=T)
\end{aligned}
\tag{B.1}
$$

where

$$
\begin{aligned}
\mathbf{A}_{t_i}^{(m)} &= \mathbf{Q}_{t_{i-1}}^{(m)-1} + \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}}\tilde{\mathbf{R}}_{t_i}^{(m)-1}\tilde{\mathbf{H}}_{t_i}^{(m)} && i=1,2,\ldots,T\\
\mathbf{B}_{t_i}^{(m)} &= \mathbf{F}_{t_i}^{(m)\mathsf{T}}\mathbf{Q}_{t_i}^{(m)-1} && i=0,1,\ldots,T-1\\
\mathbf{c}_{t_i}^{(m)} &= \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}}\tilde{\mathbf{R}}_{t_i}^{(m)-1}\tilde{\mathbf{z}}_{t_i}^{(m)} && i=1,2,\ldots,T\\
\mathbf{D}_{t_i}^{(m)} &= \mathbf{F}_{t_i}^{(m)\mathsf{T}}\mathbf{Q}_{t_i}^{(m)-1}\mathbf{F}_{t_i}^{(m)} && i=0,1,\ldots,T-1
\end{aligned}
\tag{B.2}
$$

Directly solving the system of equations (B.1) requires the inversion of a $T+1$ by $T+1$ matrix, where T is the length of the batch of (composite) measurements. This method

[1] The composite measurement model is defined in chapter 6, page 78.

becomes computationally expensive if, as often is the case, large batches of measurements are to be processed. Streit and Luginbuhl (1995) recognised the forward and backward recursive structure of the equations (B.1), and they found that the solution could be obtained by processing the measurements through a *fixed interval Kalman smoother.* The derivation of this bi-directional recursive algorithm from the equations (B.1) is included below for completeness.

B.1 State Estimates

The given known covariance matrices, i.e., $\bar{\Sigma}_{t_0}^{(m)}, \mathbf{Q}_{t_0}^{(m)}, \ldots, \mathbf{Q}_{t_{T-1}}^{(m)}, \tilde{\mathbf{R}}_{t_1}^{(m)}, \ldots, \tilde{\mathbf{R}}_{t_T}^{(m)}$, are assumed to be symmetric. This assumption is reasonable because, by definition, each of these covariance matrices is the expectation of the outer product of a vector with itself, e.g.,

$$\bar{\Sigma}_{t_0}^{(m)} = \mathrm{E}\left[\left(\bar{\mathbf{x}}_{t_0}^{(m)} - \mathbf{x}_{t_0}^{(m)}\right)\left(\bar{\mathbf{x}}_{t_0}^{(m)} - \mathbf{x}_{t_0}^{(m)}\right)^{\mathsf{T}}\right] \tag{B.3}$$

where $\mathbf{x}_{t_0}^{(m)}$ is the true state of target m at time t_0.

The derivation begins by replacing the target states, $\mathbf{x}_{t_i}^{(m)}$, in (B.1) with their estimates, $\hat{\mathbf{x}}_{t_i}^{(m)}$. Then, using the first equation from (B.1), the state estimate at time t_0 is expressed in terms of $\hat{\mathbf{x}}_{t_1}^{(m)}$, i.e.,

$$\begin{aligned}\hat{\mathbf{x}}_{t_0}^{(m)} &= \left(\bar{\Sigma}_{t_0}^{(m)-1} + \mathbf{F}_{t_0}^{(m)\mathsf{T}}\mathbf{Q}_{t_0}^{(m)-1}\mathbf{F}_{t_0}^{(m)}\right)^{-1}\left(\bar{\Sigma}_{t_0}^{(m)-1}\bar{\mathbf{x}}_{t_0}^{(m)} + \mathbf{F}_{t_0}^{(m)\mathsf{T}}\mathbf{Q}_{t_0}^{(m)-1}\hat{\mathbf{x}}_{t_1}^{(m)}\right) \\ &= \bar{\mathbf{x}}_{t_0}^{(m)} + \bar{\Sigma}_{t_0}^{(m)}\mathbf{F}_{t_0}^{(m)\mathsf{T}}\left(\mathbf{F}_{t_0}^{(m)}\bar{\Sigma}_{t_0}^{(m)}\mathbf{F}_{t_0}^{(m)\mathsf{T}} + \mathbf{Q}_{t_0}^{(m)}\right)^{-1}\left(\hat{\mathbf{x}}_{t_1}^{(m)} - \mathbf{F}_{t_0}^{(m)}\bar{\mathbf{x}}_{t_0}^{(m)}\right).\end{aligned} \tag{B.4}$$

The concept of a *forward state estimate*, $\hat{\mathbf{y}}_{t_i|t_j}^{(m)}$, is introduced as the state estimate for target m at time t_i, given all measurements up to and including time t_j for $t_j \leqslant t_i$. The covariance of this forward state estimate is

$$\mathbf{P}_{t_i|t_j}^{(m)} = \mathrm{E}\left[\left(\hat{\mathbf{y}}_{t_i|t_j}^{(m)} - \mathbf{x}_{t_i}^{(m)}\right)\left(\hat{\mathbf{y}}_{t_i|t_j}^{(m)} - \mathbf{x}_{t_i}^{(m)}\right)^{\mathsf{T}}\right] \tag{B.5}$$

where $\mathbf{x}_{t_i}^{(m)}$ represents the true state of target m at time t_i.

The initial forward state estimate and its covariance are taken as the *a priori* state estimate and its covariance, i.e.,

$$\begin{aligned}\hat{\mathbf{y}}_{t_0|t_0}^{(m)} &= \bar{\mathbf{x}}_{t_0}^{(m)} \\ \mathbf{P}_{t_0|t_0}^{(m)} &= \bar{\Sigma}_{t_0}^{(m)}.\end{aligned} \tag{B.6}$$

Using a linear Gaussian target dynamic model (5.13), the predicted forward state estimate

at time t_1 is defined as $\hat{\mathbf{y}}^{(m)}_{t_1|t_0} = \mathbf{F}^{(m)}_{t_0}\hat{\mathbf{y}}^{(m)}_{t_0|t_0}$, with covariance

$$\mathbf{P}^{(m)}_{t_1|t_0} = \mathbf{F}^{(m)}_{t_0}\mathbf{P}^{(m)}_{t_0|t_0}\mathbf{F}^{(m)\mathsf{T}}_{t_0} + \mathbf{Q}^{(m)}_{t_0}. \tag{B.7}$$

Substituting (B.6) and (B.7) into (B.4) gives

$$\hat{\mathbf{x}}^{(m)}_{t_0} = \hat{\mathbf{y}}^{(m)}_{t_0|t_0} + \mathbf{P}^{(m)}_{t_0|t_0}\mathbf{F}^{(m)\mathsf{T}}_{t_0}\mathbf{P}^{(m)-1}_{t_1|t_0}\left(\hat{\mathbf{x}}^{(m)}_{t_1} - \mathbf{F}^{(m)}_{t_0}\hat{\mathbf{y}}^{(m)}_{t_0|t_0}\right). \tag{B.8}$$

This is the state estimate at time t_0 expressed as a function of the forward state estimate at time t_0, $\hat{\mathbf{y}}^{(m)}_{t_0|t_0}$, and the state estimate at time t_1, $\hat{\mathbf{x}}^{(m)}_{t_1}$.

From the i^{th} equation in (B.1), the state estimate at time t_i is

$$\begin{aligned}\hat{\mathbf{x}}^{(m)}_{t_i} = &\left(\mathbf{Q}^{(m)-1}_{t_{i-1}} + \tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\tilde{\mathbf{R}}^{(m)-1}_{t_i}\tilde{\mathbf{H}}^{(m)}_{t_i} + \mathbf{F}^{(m)\mathsf{T}}_{t_i}\mathbf{Q}^{(m)-1}_{t_i}\mathbf{F}^{(m)}_{t_i}\right)^{-1} \\ &\times \left[\tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\tilde{\mathbf{R}}^{(m)-1}_{t_i}\tilde{\mathbf{z}}^{(m)}_{t_i} + \mathbf{Q}^{(m)-1}_{t_{i-1}}\mathbf{F}^{(m)}_{t_{i-1}}\hat{\mathbf{x}}^{(m)}_{t_{i-1}} + \mathbf{F}^{(m)\mathsf{T}}_{t_i}\mathbf{Q}^{(m)-1}_{t_i}\hat{\mathbf{x}}^{(m)}_{t_{i+1}}\right].\end{aligned} \tag{B.9}$$

Modifying (B.8) to represent $\hat{\mathbf{x}}^{(m)}_{t_{i-1}}$ and substituting into (B.9), the state estimate at t_i becomes

$$\begin{aligned}\hat{\mathbf{x}}^{(m)}_{t_i} = &\left(\mathbf{Q}^{(m)-1}_{t_{i-1}} - \mathbf{Q}^{(m)-1}_{t_{i-1}}\mathbf{F}^{(m)}_{t_{i-1}}\mathbf{P}^{(m)}_{t_{i-1}|t_{i-1}}\mathbf{F}^{(m)\mathsf{T}}_{t_{i-1}}\mathbf{P}^{(m)-1}_{t_i|t_{i-1}} + \tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\tilde{\mathbf{R}}^{(m)-1}_{t_i}\tilde{\mathbf{H}}^{(m)}_{t_i}\right. \\ &\left. + \mathbf{F}^{(m)\mathsf{T}}_{t_i}\mathbf{Q}^{(m)-1}_{t_i}\mathbf{F}^{(m)}_{t_i}\right)^{-1}\left[\mathbf{F}^{(m)\mathsf{T}}_{t_i}\mathbf{Q}^{(m)-1}_{t_i}\hat{\mathbf{x}}^{(m)}_{t_{i+1}} + \tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\tilde{\mathbf{R}}^{(m)-1}_{t_i}\tilde{\mathbf{z}}^{(m)}_{t_i}\right. \\ &\left. + \mathbf{Q}^{(m)-1}_{t_{i-1}}\mathbf{F}^{(m)}_{t_{i-1}}\left(\mathbf{1} - \mathbf{P}^{(m)}_{t_{i-1}|t_{i-1}}\mathbf{F}^{(m)\mathsf{T}}_{t_{i-1}}\mathbf{P}^{(m)-1}_{t_i|t_{i-1}}\mathbf{F}^{(m)}_{t_{i-1}}\right)\hat{\mathbf{y}}^{(m)}_{t_{i-1}|t_{i-1}}\right].\end{aligned} \tag{B.10}$$

Denoting the forward state estimate covariances as

$$\begin{aligned}\mathbf{P}^{(m)}_{t_i|t_i} &= \left(\mathbf{P}^{(m)-1}_{t_i|t_{i-1}} + \tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\tilde{\mathbf{R}}^{(m)-1}_{t_i}\tilde{\mathbf{H}}^{(m)}_{t_i}\right)^{-1} \\ &= \mathbf{P}^{(m)}_{t_i|t_{i-1}} - \mathbf{P}^{(m)}_{t_i|t_{i-1}}\tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\left(\tilde{\mathbf{H}}^{(m)}_{t_i}\mathbf{P}^{(m)}_{t_i|t_{i-1}}\tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i} + \tilde{\mathbf{R}}^{(m)}_{t_i}\right)^{-1}\tilde{\mathbf{H}}^{(m)}_{t_i}\mathbf{P}^{(m)}_{t_i|t_{i-1}} \\ \mathbf{P}^{(m)}_{t_{i+1}|t_i} &= \mathbf{F}^{(m)}_{t_i}\mathbf{P}^{(m)}_{t_i|t_i}\mathbf{F}^{(m)\mathsf{T}}_{t_i} + \mathbf{Q}^{(m)}_{t_i},\end{aligned} \tag{B.11}$$

(B.10) may be written

$$\begin{aligned}\hat{\mathbf{x}}^{(m)}_{t_i} = &\mathbf{P}^{(m)}_{t_i|t_i}\mathbf{F}^{(m)\mathsf{T}}_{t_i}\mathbf{P}^{(m)-1}_{t_{i+1}|t_i}\hat{\mathbf{x}}^{(m)}_{t_{i+1}} + \left(\mathbf{1} - \mathbf{P}^{(m)}_{t_i|t_i}\mathbf{F}^{(m)\mathsf{T}}_{t_i}\mathbf{P}^{(m)-1}_{t_{i+1}|t_i}\mathbf{F}^{(m)}_{t_i}\right) \\ &\times \left[\mathbf{F}^{(m)}_{t_{i-1}}\hat{\mathbf{y}}^{(m)}_{t_{i-1}|t_{i-1}} + \mathbf{P}^{(m)}_{t_i|t_i}\tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\tilde{\mathbf{R}}^{(m)-1}_{t_i}\left(\tilde{\mathbf{z}}^{(m)}_{t_i} - \tilde{\mathbf{H}}^{(m)}_{t_i}\mathbf{F}^{(m)}_{t_{i-1}}\hat{\mathbf{y}}^{(m)}_{t_{i-1}|t_{i-1}}\right)\right].\end{aligned} \tag{B.12}$$

Defining the forward state estimate at time t_i as

$$\hat{\mathbf{y}}^{(m)}_{t_i|t_i} = \mathbf{F}^{(m)}_{t_{i-1}}\hat{\mathbf{y}}^{(m)}_{t_{i-1}|t_{i-1}} + \mathbf{P}^{(m)}_{t_i|t_i}\tilde{\mathbf{H}}^{(m)\mathsf{T}}_{t_i}\tilde{\mathbf{R}}^{(m)-1}_{t_i}\left(\tilde{\mathbf{z}}^{(m)}_{t_i} - \tilde{\mathbf{H}}^{(m)}_{t_i}\mathbf{F}^{(m)}_{t_{i-1}}\hat{\mathbf{y}}^{(m)}_{t_{i-1}|t_{i-1}}\right), \tag{B.13}$$

i.e., a Kalman filter, the assumed forward state estimate covariance definitions in (B.11)

are correct, and substituting (B.13) into (B.12), the state estimate at time t_i becomes

$$\hat{\mathbf{x}}_{t_i}^{(m)} = \hat{\mathbf{y}}_{t_i|t_i}^{(m)} + \mathbf{P}_{t_i|t_i}^{(m)} \mathbf{F}_{t_i}^{(m)\mathsf{T}} \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \left(\hat{\mathbf{x}}_{t_{i+1}}^{(m)} - \mathbf{F}_{t_i}^{(m)} \hat{\mathbf{y}}_{t_i|t_i}^{(m)} \right). \tag{B.14}$$

The state estimate at time t_i (B.14) is a function of the forward state estimate at time t_i, $\hat{\mathbf{y}}_{t_i|t_i}^{(m)}$, and the state estimate at time t_{i+1}, $\hat{\mathbf{x}}_{t_{i+1}}^{(m)}$. The forward state estimate (B.13) is a function of the previous forward state estimate at time t_{i-1} and the measurement at time t_i, $\tilde{\mathbf{z}}_{t_i}^{(m)}$. Similarly, the forward state estimate covariances (B.11) are functions of the forward state estimate covariances from previous times and the measurement covariance at the present time. Therefore the forward state estimates and their covariances may be calculated using forward recursion, which may then be followed by backward recursion to calculate the final smoothed state estimates.

All that is now required is some way of terminating the forward recursion to allow the backward recursion to commence. Looking to the final equation in (B.1), the state estimate at the end of the batch of measurements is

$$\hat{\mathbf{x}}_{t_T}^{(m)} = \left(\mathbf{Q}_{t_{T-1}}^{(m)-1} + \tilde{\mathbf{H}}_{t_T}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_T}^{(m)-1} \tilde{\mathbf{H}}_{t_T}^{(m)} \right)^{-1} \left[\tilde{\mathbf{H}}_{t_T}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_T}^{(m)-1} \tilde{\mathbf{z}}_{t_T}^{(m)} + \mathbf{Q}_{t_{T-1}}^{(m)-1} \mathbf{F}_{t_{T-1}}^{(m)} \hat{\mathbf{x}}_{t_{T-1}}^{(m)} \right]. \tag{B.15}$$

Modifying (B.14) for $\hat{\mathbf{x}}_{t_{Ti-1}}^{(m)}$ and substituting it into (B.15)

$$\begin{aligned} \hat{\mathbf{x}}_{t_T}^{(m)} = & \left(\mathbf{Q}_{t_{T-1}}^{(m)-1} - \mathbf{Q}_{t_{T-1}}^{(m)-1} \mathbf{F}_{t_{T-1}}^{(m)} \mathbf{P}_{t_{T-1}|t_{T-1}}^{(m)} \mathbf{F}_{t_{T-1}}^{(m)\mathsf{T}} \mathbf{P}_{t_T|t_{T-1}}^{(m)-1} + \tilde{\mathbf{H}}_{t_T}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_T}^{(m)-1} \tilde{\mathbf{H}}_{t_T}^{(m)} \right)^{-1} \\ & \times \left[\tilde{\mathbf{H}}_{t_T}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_T}^{(m)-1} \tilde{\mathbf{z}}_{t_T}^{(m)} + \mathbf{Q}_{t_{T-1}}^{(m)-1} \mathbf{F}_{t_{T-1}}^{(m)} \left(\mathbf{1} - \mathbf{P}_{t_{T-1}|t_{T-1}}^{(m)} \mathbf{F}_{t_{T-1}}^{(m)\mathsf{T}} \mathbf{P}_{t_T|t_{T-1}}^{(m)-1} \mathbf{F}_{t_{T-1}}^{(m)} \right) \hat{\mathbf{y}}_{t_{T-1}|t_{T-1}}^{(m)} \right]. \end{aligned} \tag{B.16}$$

Using forward state estimate covariance definitions for $\mathbf{P}_{t_{T-1}|t_{T-1}}^{(m)}$, $\mathbf{P}_{t_T|t_{T-1}}^{(m)}$ and $\mathbf{P}_{t_T|t_T}^{(m)}$ that are similar to those in (B.11),

$$\begin{aligned} \hat{\mathbf{x}}_{t_T}^{(m)} &= \mathbf{F}_{t_{T-1}}^{(m)} \hat{\mathbf{y}}_{t_{T-1}|t_{T-1}}^{(m)} + \mathbf{P}_{t_T|t_T}^{(m)} \tilde{\mathbf{H}}_{t_T}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_T}^{(m)-1} \left(\tilde{\mathbf{z}}_{t_T}^{(m)} - \tilde{\mathbf{H}}_{t_T}^{(m)} \mathbf{F}_{t_{T-1}}^{(m)} \hat{\mathbf{y}}_{t_{T-1}|t_{T-1}}^{(m)} \right) \Big] \\ &= \hat{\mathbf{y}}_{t_T|t_T}^{(m)}. \end{aligned} \tag{B.17}$$

Therefore at time t_T, the state estimate is equal to the forward state estimate at time t_T. The forward recursion ceases at this point because no further measurements are available and, because the state estimate at this time is known, the backward recursion commences.

B.2 Summary of the Kalman Smoother Algorithm

The fixed interval Kalman smoother algorithm may be summarised as follows.

1. Initialise the forward state estimates using

$$\begin{aligned} \hat{\mathbf{y}}_{t_0|t_0}^{(m)} &= \bar{\mathbf{x}}_{t_0}^{(m)} \\ \mathbf{P}_{t_0|t_0}^{(m)} &= \bar{\Sigma}_{t_0}^{(m)}. \end{aligned} \tag{B.18}$$

2. Perform the forward recursion, $i = 1, \ldots, T$, with

$$\begin{aligned} \mathbf{P}_{t_i|t_{i-1}}^{(m)} &= \mathbf{F}_{t_{i-1}}^{(m)} \mathbf{P}_{t_{i-1}|t_{i-1}}^{(m)} \mathbf{F}_{t_{i-1}}^{(m)\mathsf{T}} + \mathbf{Q}_{t_{i-1}}^{(m)} \\ \mathbf{P}_{t_i|t_i}^{(m)} &= \left(\mathbf{P}_{t_i|t_{i-1}}^{(m)-1} + \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_i}^{(m)-1} \tilde{\mathbf{H}}_{t_i}^{(m)} \right)^{-1} \\ \hat{\mathbf{y}}_{t_i|t_i}^{(m)} &= \mathbf{F}_{t_{i-1}}^{(m)} \hat{\mathbf{y}}_{t_{i-1}|t_{i-1}}^{(m)} + \mathbf{P}_{t_i|t_i}^{(m)} \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_i}^{(m)-1} \left(\tilde{\mathbf{z}}_{t_i}^{(m)} - \tilde{\mathbf{H}}_{t_i}^{(m)} \mathbf{F}_{t_{i-1}}^{(m)} \hat{\mathbf{y}}_{t_{i-1}|t_{i-1}}^{(m)} \right). \end{aligned} \tag{B.19}$$

3. Determine the state estimate at time t_T, i.e.,

$$\hat{\mathbf{x}}_{t_T} = \hat{\mathbf{y}}_{t_T|t_T}^{(m)}. \tag{B.20}$$

4. Calculate the final state estimates using backward recursion, $i = T-1, \ldots, 0$, i.e.,

$$\hat{\mathbf{x}}_{t_i}^{(m)} = \hat{\mathbf{y}}_{t_i|t_i}^{(m)} + \mathbf{P}_{t_i|t_i}^{(m)} \mathbf{F}_{t_i}^{(m)\mathsf{T}} \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \left(\hat{\mathbf{x}}_{t_{i+1}}^{(m)} - \mathbf{F}_{t_i}^{(m)} \hat{\mathbf{y}}_{t_i|t_i}^{(m)} \right). \tag{B.21}$$

The equations (B.18) to (B.21) represent the Rauch-Tung-Striebel form of the fixed interval Kalman smoother (Gelb 1992, p 164), as recognised by Streit and Luginbuhl (1995). Also, the forward state estimates $\hat{\mathbf{y}}_{t_i|t_i}^{(m)}$ are equivalent to the state estimates from a Kalman filter.

B.3 State Estimate Error Covariance

Although not required to estimate the target states, the state estimate error covariance may prove useful in further processing. Therefore its derivation is given below.

Using (B.14), (5.13) and (5.15), the state estimate error is defined as

$$\begin{aligned} \tilde{\mathbf{x}}_{t_i}^{(m)} &= \hat{\mathbf{x}}_{t_i}^{(m)} - \mathbf{x}_{t_i}^{(m)} \\ &= \tilde{\mathbf{y}}_{t_i|t_i}^{(m)} + \mathbf{P}_{t_i|t_i}^{(m)} \mathbf{F}_{t_i}^{(m)\mathsf{T}} \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \left(\tilde{\mathbf{x}}_{t_{i+1}}^{(m)} - \left(\mathbf{F}_{t_i}^{(m)} \tilde{\mathbf{y}}_{t_i|t_i}^{(m)} - \mathbf{w}_{t_i}^{(m)} \right) \right) \end{aligned} \tag{B.22}$$

where, from (B.13), the forward state error is given as

$$\begin{aligned} \tilde{\mathbf{y}}_{t_i|t_i}^{(m)} &= \hat{\mathbf{y}}_{t_i|t_i}^{(m)} - \mathbf{x}_{t_i}^{(m)} \\ &= \left(\mathbf{1} - \mathbf{P}_{t_i|t_i}^{(m)} \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_i}^{(m)-1} \tilde{\mathbf{H}}_{t_i}^{(m)} \right) \left(\mathbf{F}_{t_{i-1}}^{(m)} \tilde{\mathbf{y}}_{t_{i-1}|t_{i-1}}^{(m)} - \mathbf{w}_{t_{i-1}}^{(m)} \right) \\ &\quad + \mathbf{P}_{t_i|t_i}^{(m)} \tilde{\mathbf{H}}_{t_i}^{(m)\mathsf{T}} \tilde{\mathbf{R}}_{t_i}^{(m)-1} \tilde{\mathbf{v}}_{t_i}^{(m)}. \end{aligned} \tag{B.23}$$

The covariance of the state estimate error at time t_i is defined as

$$\boldsymbol{\Sigma}_{t_i}^{(m)} = \mathrm{E}\left[\tilde{\mathbf{x}}_{t_i}^{(m)} \tilde{\mathbf{x}}_{t_i}^{(m)\mathsf{T}}\right] \tag{B.24}$$

and, given the definition of the forward state estimate covariance $\mathbf{P}_{t_i|t_j}^{(m)}$ in (B.5), it is written

$$\boldsymbol{\Sigma}_{t_i}^{(m)} = \mathbf{P}_{t_i|t_i}^{(m)} + \mathbf{P}_{t_i|t_i}^{(m)} \mathbf{F}_{t_i}^{(m)\mathsf{T}} \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \left(\boldsymbol{\Sigma}_{t_{i+1}}^{(m)} - \mathbf{P}_{t_{i+1}|t_i}^{(m)}\right) \mathbf{P}_{t_{i+1}|t_i}^{(m)-1} \mathbf{F}_{t_i}^{(m)} \mathbf{P}_{t_i|t_i}^{(m)}. \tag{B.25}$$

The calculation of the state estimate error covariance may be performed with the state estimation during the backward recursion. Similarly to the state estimate, the state estimate error covariance at time t_T is defined as the forward state error covariance at that time, i.e., $\boldsymbol{\Sigma}_{t_T}^{(m)} = \mathbf{P}_{t_T|t_T}^{(m)}$.

Bibliography

Ahmeda, S. S., Harrison, I. and Woolfson, M. S.: 1996, Adaptive probabilistic data-association algorithm for tracking in a cluttered environment, *IEE Proc. Radar, Sonar and Navigation* **143**(1), 17–22.

Avitzour, D.: 1992, A maximum likelihood approach to data association, *IEEE Trans Aerospace and Electronic Systems* **28**(2), 560–5.

Bar-Shalom, Y.: 1981, On the track-to-track correlation problem, *IEEE Trans Automatic Control* **26**, 571–2.

Bar-Shalom, Y. and Campo, L.: 1986, The effect of the common process noise on the two-sensor fused-track covariance, *IEEE Trans Aerospace and Electronic Systems* **22**, 803–5.

Bar-Shalom, Y. (ed.): 1990, *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, Norwood, MA, USA.

Bar-Shalom, Y. (ed.): 1992, *Multitarget–Multisensor Tracking: Applications and Advances Volume II*, Artech House, Norwood, MA, USA.

Bar-Shalom, Y. and Fortmann, T. E.: 1988, *Tracking and Data Association*, Academic Press, Orlando, FL, USA.

Bertsekas, D. P.: 1981, A new algorithm for the assignment problem, *Mathematical Programming* **21**, 152–71.

Bertsekas, D. P.: 1991, *Linear Network Optimization: Algorithms and Codes*, MIT Press, Cambridge, MA, USA.

Bertsekas, D. P., Castanon, D. A. and Tsaknakis, H.: 1993, Reverse auction and the solution of inequality constrained assignment problems, *SIAM J. Optimization* **3**, 268–97.

Blair, W. D., Rice, T. R., Alouani, A. T. and Xia, P.: 1991, Asynchronous data fusion for target tracking with a multi-tasking radar and optical sensor, *Acquisition, Tracking and Pointing V*, Vol. 1482 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 234–45.

Castanon, D. A.: 1992, New assignment algorithms for data association, *Signal and Data Processing of Small Targets 1992*, Vol. 1698 of *Proceeding of the SPIE*, Orlando, FL, USA, pp. 313–23.

Deb, S., Mallubhatla, R., Pattipati, K. R. and Bar-Shalom, Y.: 1990, A multisensor multitarget data association algorithm, *Proc. IEEE Int. Conf. on Systems Engineering*, Pittsburgh, PA, USA, pp. 320–3.

Deb, S., Pattipati, K. R. and Bar-Shalom, Y.: 1993, A multisensor multitarget data association algorithm for heterogeneous sensors, *IEEE Trans Aerospace and Electronic Systems* **29**(2), 562–8.

Deb, S., Yeddanapudi, M., Pattipati, K. and Bar-Shalom, Y.: 1997, A generalized s-d assignment algorithm for multisensor-multitarget state estimation, *IEEE Trans Aerospace and Electronic Systems* **33**(2), 523–38.

Dempster, A. P., Laird, N. M. and Rubin, D. B.: 1977, Maximum likelihood from incomplete data via the em algorithm, *J. Royal Statistical Society Ser. B* **39**(1), 1–38.

Dufour, F. and Mariton, M.: 1991a, Passive data fusion and tracking with target manoeuvers, *Proc. 1991 American Control Conference*, Vol. 1, Boston, MA, USA, pp. 852–3.

Dufour, F. and Mariton, M.: 1991b, Tracking a 3d manoeuvring target with passive sensors, *IEEE Trans Aerospace and Electronic Systems* **27**(4), 725–39.

Dunham, D. T. and Hutchins, R. G.: 1997, Tracking multiple targets in cluttered environments with a probabilistic multi-hypothesis tracker, *Acquisition, Tracking and Pointing XI*, Vol. 3086 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 284–95.

Filippidis, A. and Bogner, R. E.: 1992, Track-to-track correlation for sensor level tracking, *Proc. Int. Symp. on Signal Processing and its Applications (ISSPA-92)*, Vol. 2, Gold Coast, Qld, Australia, pp. 666–9.

Gauvrit, H., Jauffret, C. and le Cadre, J. P.: 1997, A general formulation of multitarget tracking in the viewpoint of an incomplete data problem, accepted *IEEE Trans Aerospace and Electronic Systems.*

Gelb, A.: 1992, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, USA.

Giannopoulos, E., Streit, R. and Swaszek, P.: 1996, Probabilistic multi-hypothesis tracking in a multi-sensor, multi-target environment, *Proc. 1st Australian Data Fusion Symposium (ADFS-96)*, Adelaide, SA, Australia, pp. 184–9.

Haimovich, A. M., Yosko, J., Greenberg, R. J., Parisi, M. A. and Becker, D.: 1993, Fusion of sensors with dissimilar measurement/tracking accuracies, *IEEE Trans Aerospace and Electronic Systems* **29**(1), 245–50.

Horton, M. J. and Jones, R. A.: 1995, Fuzzy logic extended rule set for multitarget tracking, *Acquisition, Tracking and Pointing IX*, Vol. 2468 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 106–17.

Houles, A. and Bar-Shalom, Y.: 1989, Multisensor tracking of a maneuvering target in clutter, *IEEE Trans Aerospace and Electronic Systems* **25**(2), 176–89.

Jazwinski, A. H.: 1970, *Stochastic Processes and Filtering Theory*, Academic, San Diego, CA, USA.

Korona, Z. and Kokar, M. M.: 1995, Multiple scan joint probabilistic data association, *IEEE Trans Aerospace and Electronic Systems* **31**(3), 1204–1215.

Krieg, M. L.: 1997, Asynchronous single platform sensor fusion, *Technical Note DSTO-TN-0084*, Defence Science and Technology Organisation, Australia, Salisbury, SA, Australia.

Krieg, M. L. and Gray, D. A.: 1996a, Multi-sensor probabilistic multi-hypothesis tracking, *Proc. 1st Australian Data Fusion Symposium (ADFS-96)*, Adelaide, SA, Australia, pp. 153–8.

Krieg, M. L. and Gray, D. A.: 1996b, Performance of state space multi-sensor track fusion with model mismatch, *Proc. 1st Australian Data Fusion Symposium (ADFS-96)*, Adelaide, SA, Australia, pp. 1–6.

Krieg, M. L. and Gray, D. A.: 1996c, Radar and optical track fusion using real data, *Proc. 1st Australian Data Fusion Symposium (ADFS-96)*, Adelaide, SA, Australia, pp. 25–30.

Krieg, M. L. and Gray, D. A.: 1996d, Track fusion in the presence of an interference, *Proc. 4th Int. Symp. on Signal Processing and its Applications (ISSPA-96)*, Vol. 1, Gold Coast, Qld, Australia, pp. 192–5.

Krieg, M. L. and Gray, D. A.: 1997a, Comparison of probabilistic least squares and probabilistic multi-hypothesis tracking algorithms for multi-sensor tracking, *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP-97)*, Vol. 1, Munich, Germany, pp. 515–8.

Krieg, M. L. and Gray, D. A.: 1997b, Comparisons of pmh and pls trackers on real and simulated data, accepted *EURASIP Signal Processing.*

Krieg, M. L. and Gray, D. A.: 1997c, Multi-sensor probabilistic multi-hypothesis tracking using dissimilar sensors, *Acquisition, Tracking and Pointing XI*, Vol. 3086 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 129–138.

Malkoff, D. B.: 1997, Evaluation of the jonker-volgenant-castanon (jvc) assignment algorithm for track association,, *Signal Processing, Sensor Fusion and Target Recognition VI*, Vol. 3068 of *Proceedings of the SPIE*, Orlando, FL.

Martinerie, F. and Forster, P.: 1992, Data association and tracking using hidden markov models and dynamic programming, *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP-92)*, Vol. 2, San Francisco, CA, USA, pp. 449–52.

Moon, T. K.: 1996, The expectation-maximisation algorithm, *IEEE Signal Processing* pp. 47–60.

Munkres, J.: 1957, Algorithms for the assignment and transportation problems, *J. Soc. Indust. Applied Mathematics* **5**(1), 32–8.

Musicki, D. and Evans, R. J.: 1992, Tracking in clutter using probabilistic data association, *Proc. IEE Int. Conf. Radar 92*, Brighton, UK, pp. 82–5.

Pattipati, K. R., Deb, S., Bar-Shalom, Y. and Washburn, R. B.: 1992, Passive multisensor data association using a new relaxation algorithm, *IEEE Trans Automatic Control* **37**(2), 198–213.

Poore, A. B. and Rijavec, N.: 1991, Multi-target tracking and multi-dimensional assignment problems, *Signal and Data Processing of Small Targets 1991*, Vol. 1481 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 345–56.

Poore, A. B. and Rijavec, N.: 1993, A lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking, *SIAM J. Optimization* **3**(3), 544–63.

Poore, A. B., Robertson III, A. J. and Shea, P. J.: 1995, A new class of lagrangian relaxation based algorithms for fast data association in multiple hypothesis tracking applications, *Signal Processing, Sensor Fusion and Target Recognition IV*, Vol. 2484 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 184–94.

Pulford, G. W. and Evans, R. J.: 1996, Probabilistic data association for systems with multiple simultaneous measurements, *Automatica* **32**(9), 1311–6.

Rago, C., Willett, P. and Streit, R.: 1995a, A comparison of the jpdaf and pmht tracking algorithms, *Proc. IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP-95)*, Vol. 5, Detroit, MI, USA, pp. 3571–4.

Rago, C., Willett, P. and Streit, R.: 1995b, Direct data fusion using the pmht, *Proc. 1995 American Control Conference*, Vol. 3, Seattle, WA, USA, pp. 1698–702.

Rago, C., Willett, P. and Streit, R.: 1995c, A modified pmht, *Proc. 29th Annual Conf. on Information Sciences and Systems*, Baltimore, MD, USA, pp. 647–52.

Reid, D. B.: 1979, An algorithm for tracking multiple targets, *IEEE Trans Automatic Control* **24**(6), 843–54.

Roecker, J. A. and McGillem, C. D.: 1988, Comparison of two-sensor tracking methods based on state fusion and measurement fusion, *IEEE Trans Aerospace and Electronic Systems* **24**(4), 447–9.

Romine, J. B., Kamen, E. W. and Sastry, C. R.: 1994, Fusion of radar and imaging sensor data for target tracking, *Signal and Data Processing of Small Targets 1994*, Vol. 2235 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 558–68.

Saha, R. K.: 1994, Effect of cross-correlation on track-to-track fusion, *Signal and Data Processing of Small Targets 1994*, Vol. 2235 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 474–85.

Streit, R. L. and Luginbuhl, T. E.: 1993, A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning, *Proc. Sixth Joint Service Data Fusion Symposium*, Laurel, MD, USA, pp. 1015–24.

Streit, R. L. and Luginbuhl, T. E.: 1994, Maximum likelihood method for probabilistic multi-hypothesis tracking, *Signal and Data Processing of Small Targets 1994*, Vol. 2235 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 394–405.

Streit, R. L. and Luginbuhl, T. E.: 1995, Probabilistic multi-hypothesis tracking, *Technical Report TR 10,428*, Naval Undersea Warfare Center Division, Newport, RI, USA.

Thompson, W. E., Parra-Loera, R. and Tao, C.: 1991, A pseudo k-means approach to the multisensor multitarget tracking problem, *Data Structures and Target Classification*, Vol. 1470 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 48–58.

van Keuk, G.: 1995, Multihypothesis tracking with electronically scanned radar, *IEEE Trans Aerospace and Electronic Systems* **31**(3), 916–27.

Wang, F., Litva, J., Lo, T. and Bosse, E.: 1996, Performance of neural data associator, *IEE Proc. Radar, Sonar and Navigation* **143**(2), 71–8.

Werthmann, J. R.: 1992, A step-by-step description of a computationally efficient version of multiple hypothesis tracking, *Signal and Data Processing of Small Targets 1992*, Vol. 1698 of *Proceedings of the SPIE*, Orlando, FL, USA, pp. 288–300.

Willner, D., Chang, C. B. and Dunn, K. P.: 1976, Kalman filter algorithms for a multi-sensor system, *Proc. 15th IEEE Conf. Decision and Control*, Clearwater Beach, FL, USA, pp. 570–4.

Wu, C. F. J.: 1983, On the convergence properties of the em algorithm, *Annals of Statistics* **11**(1), 95–103.

Zhou, B. and Bose, N. K.: 1995, An efficient algorithm for data association in multitarget tracking, *IEEE Trans Aerospace and Electronic Systems* **31**(1), 458–68.