Durham E-Theses

Measures of effectiveness for data fusion based on information entropy

Noonan, Colin Anthony

Measures of effectiveness for Data Fusion based on Information Entropy

Colin Anthony Noonan

Abstract

This thesis is concerned with measuring and predicting the performance and effectiveness of a data fusion process. Its central proposition is that information entropy may be used to quantify concisely the effectiveness of the process. The personal and original contribution to that subject which is contained in this thesis is summarised as follows:

- The mixture of performance behaviours that occur in a data fusion system are described and modelled as the states of an ergodic Markov process.

- An new analytic approach to combining the entropy of discrete and continuous information is defined.

- A new simple and accurate model of data association performance is proposed.

- A new model is proposed for the propagation of information entropy in an minimum mean square combination of track estimates.

- A new model is proposed for the propagation of the information entropy of object classification belief as new observations are incorporated in a recursive Bayesian classifier.

- A new model to quantify the information entropy of the penalty of ignorance is proposed.

- New formulations of the steady state solution of the matrix Riccati equation to model tracker performance are proposed.

Measures of effectiveness for Data Fusion

based on Information Entropy

Colin Anthony Noonan

PhD

University of Durham

Department of Physics

2000

Table of Contents

List of Tables and Illustrations

Declaration

No part of the material offered in this thesis has been submitted by the author for a degree in this or in any other University. If material has been generated through joint work, the author's independent contribution has been clearly indicated. In all other cases, material from the work of others has been acknowledged and quotations and paraphrases suitably indicated.

Statement of Copyright

The copyright of this thesis rests with the author. No quotation from it should be published without their prior written consent and information derived from it should be acknowledged.

Acknowledgements

Thanks to Dr. Keith Orford, my supervisor at the University of Durham, for guiding and encouraging me and for maintaining an interest in and enthusiasm for this work over nearly six years of intermittent progress.

To BAE SYSTEMS for providing the funding for the work.

To Mike Everett for encouraging me to start and for always asking very good questions.

To Rob Deaves and the BAE SYSTEMS Sowerby research centre for collecting the data for the ICIDS system case study and for finding time to assist me when I needed detailed access to the source data.

To Steve and Molly, my parents, for supporting me during my first degree studies, even when I was unsure of what I wanted.

Thanks to Sharon, my wife, for her love and support and her patience when my free time was taken up by this work and to Ann, Susan and Alison, my daughters, for (apparently) preferring it that way.

1   Introduction

Data fusion is a relatively new term which has been coined in the last decade or so to describe processes which take place inside information systems whose purpose is to provide situation awareness [82]. The situation in question can be anything from the trends in equity prices on the Stock Market to the location and status of each aircraft which is present in the airspace around a busy airport. In either case the data fusion process receives information in small data updates and must build and maintain the bigger picture. With the advantage of the big picture the market analyst can decide, in real time, what and when to buy and sell and the air traffic controller can decide, also in real time, to allocate landing slots and holding patterns. In both cases, the success of the task depends on the timely availability and the accuracy of the picture.

One of the reasons for employing a data fusion system can be a need to process quickly a large amount of information. In general, the amounts of available information and the means of access to it have increased over recent years and appear set to continue to increase. A good example of a general, public information system where this is seen is the Internet [68] [79]. In situation awareness systems, a similar increase in information availability, and the communications capacity and connectivity necessary to gather information over a wide geographical region contribute to increased processing demands.

In modern situation awareness systems, data fusion is a complex process carried out by powerful computers. The large amount of information and the complexity of the process can mean that, using human experience and knowledge, it is difficult to predict or judge objectively the effectiveness of the system and that difficulty may increase as new and more automated systems become available. This thesis is concerned with enabling a systematic approach to measuring the effectiveness of data fusion processes for the purposes of objective prediction and evaluation.

Any information system needs to be effective [15]. Whilst access to the Internet is cheap, it would be easy to waste large amounts of time and effort searching ineffectively for information which turns out to be of little value. In contrast, situation awareness systems are extremely expensive and they must make a positive and significant contribution to their owners' operations to warrant their use. When key decisions are made on the strength of the information that situation awareness systems provide, a failing system could jeopardise an entire operation with dire consequences, be they crashed investments or crashed aeroplanes. It would be useful to know in advance whether a system can be used effectively and desirable to monitor effectiveness once a system goes into operational use.

Having stated these objectives, it must be decided how effectiveness should be quantified. Familiar characteristics of data fusion performance such as precision, accuracy, timeliness and so on may be quantified, each in its own way. But knowledge of these quantities alone does not answer the question - how effective? The answer lies in a comparison of expected or achieved performance levels with those levels required to allow the tasks and decisions of the supported operation to proceed.

If knowledge of the required performance level for every aspect of information quality is available, each can be examined in turn and the system declared effective or not in that respect. However, each might be subject to different uncertainty characteristics resulting in an effectiveness statement which is complex and subject to numerous caveats.

An alternative approach could examine the information quality obtained from a data fusion process through a single quantity into which all the aspects of information quality and the corresponding uncertainties would be combined. Such an approach would not be as precise as one which examined separately all the individual aspects of effectiveness but it would have the advantage of producing an indicator of effectiveness

which would be simple and concise even when the subject system is very large and complex.

Information entropy is one means of obtaining such a single quantity and Measures of Effectiveness for data fusion based on information entropy are the subject of this thesis.

1.1 Route Map

Chapter 1 introduces the subject of this thesis and provides this route map. The purpose of the route map is to clarify, for the reader, the additions to established work which lead to the ability to measure and predict data fusion effectiveness.

The background to this thesis is given in Chapter 2 where the nature of information systems and situation awareness are discussed, and the role of the embedded data fusion process is considered. There, the central proposition of this thesis is introduced, that *information entropy may be used to measure the effectiveness of a data fusion process*. The uses to which such measures might be put are discussed.

Chapter 3 considers the data fusion problem and numerous documented approaches to its solution. It is offered as a survey.

In Chapter 4 information and entropy as a measure of information error are considered. Most of the chapter is concerned with history and the progression of ideas that lead a measure of physical disorder in an ensemble of particles to be used to represent the value of stored, transmitted and computed information. At the end of the chapter *a new analytic approach to the calculation of combined entropy of continuous and discrete information is defined*.

Chapter 5 considers the properties of entropy in the context of data fusion and situation awareness. Spatial and temporal changes in "point-of-view" are discussed and the ability of the entropy measure to indicate shape as well as spread of the distribution

of information error is pointed out. This property has potential as the basis of a test of assumptions of Gaussian statistics.

Chapter 6 is concerned with simulations and models of data fusion. Most of the novel work in this thesis is described there.

- *A new Data Association model* is proposed which is novel because it operates in a space where distances are normalised with respect to the distribution of relative errors between new information and information already accumulated. The main advantages of this compared with known approaches are greater accuracy and applicability to a wider range of situations.

- Tracking is represented by the well known steady-state solution of the matrix Riccati equation [2]. *New formulations of the solution as a series expansion and a nested computation are offered.*

- *A new model is offered by which track fusion entropy may be propagated directly.*

- *A new model is offered for classification that predicts the evolution of the entropy of the class belief distribution* when class observations are combined using a recursive Bayesian process.

- The concept of *a quantifiable information penalty of ignorance is proposed* and a model is offered.

- *The new concept of fusion states to account for step changes in information quality is proposed and a model to predict their influence on fused information quality is offered.*

Chapter 7 describes a series of case studies performed as part of this work and discusses the results.

Finally, the conclusions of this thesis and recommendations for future work are given in Chapter 8.

2   Information Systems and Measures of Effectiveness

The effectiveness of situation awareness systems employing data fusion can be defined in various ways. Whilst factors such as access, usage and productivity are key in any definition of effectiveness of the situation awareness system, the effectiveness of a data fusion process is concerned with information quality. Simply put, an effective data fusion process delivers required information, accurately, to the right place, at the right time to allow correct decisions to be made by the systems' owners.

Accuracy with respect to required information may be characterised in the information error distribution produced by the data fusion process, where information error is regarded as the capacity of the fused information to support conclusions which are not true and decisions which are wrong. This distribution includes error in the estimation-theoretic sense, whereby the true value of a parameter is different from the estimate, and in an evidential sense where the information indicates an object or situation belonging to a class that is not the true class. It is proposed in this thesis is that the entropy of the information error distribution may be used to measure data fusion effectiveness.

The ability to measure effectiveness can then be put to use. When a new situation awareness system is designed and developed, entropy measures of effectiveness can be applied to simulations and prototype systems to establish the validity of the underlying design assumptions and confirm the suitability of the data fusion process to perform its intended task. Analytic models of effectiveness can be applied to the earliest stages of system concept definition to provide evidence of validity and suitability at the outset. Together, accurate measures and models can be used to ensure that the new situation awareness system is created with its effectiveness assured.

Much of the novel work in this thesis deals with modelling and measurement for the purposes of evaluation. In the future, similar measures and models might be applied

to system automation. For example, a situation awareness system might have sensors under automatic control which it will allocate to improve information where it is weakest. Further in the future, these information measures might be applied to the self-supervision of automatic systems.

This chapter addresses these background issues in detail. It starts with an examination of the origins of such systems and the nature of the information they hold.

2.1 Information Systems

Some authors suggest that prehistoric cave painting had as much to do with recording information as it had with art [28]. Cave painting is a relatively recent phenomenon and early examples (from Altamira in Spain) date from about 16 thousand years ago [6] [17]. In comparison with the earliest stone tools which are said to date from 2 million years ago that is very recent. In comparison with recorded history it is a long time and suggests that humanity has been treating information systematically for just that long.

The earliest writing was concerned with keeping records [25]. Clay tablets from Uruk in Sumer are said to date from 3000 BC and hold lists of objects [31]. It appears that the list was some kind of inventory and that an information system, in the true sense of Information Science, was operating.

The use of writing for scientific, religious, legal, narrative and historical works developed later. But Ashurbanipal's library in Nineveh in the 7th century BC contained thousands of clay tablets and examples of works of all these types [25].

Half a century ago, most commonplace information systems would have been recognisable to the educated people of ancient times. Half a century ago modern exceptions were exotic and unusual. Change and development had apparently been a slow and gradual process over several millennia until the sudden explosion of the last few decades produced the very large, diverse and widespread systems of today.

We still have inventories and ledgers and still collect information together into libraries and archives. In addition to these traditional information systems we have databases that maintain large amounts of detailed information in highly organised structures for fast and flexible access; we have the Internet that provides an information network to connect us to huge amounts of information; and we have situation awareness systems to create and maintain a picture of the world (or some aspect of it) as it is now in real time, keeping pace as the world changes.

2.1.1 Immutable Information

Some information systems exist to provide access to a store of information. Filing cabinets, libraries and computer databases are examples of this. New items may be added to the store and out of date ones may be removed but the information content of an item need not change significantly during its time within the store. Items can be gathered, changed and removed by human choice and at a human pace. The information might be regarded as immutable. A fact is a fact and an author's contribution to a subject does not change when subsequent works throw new light. But not all information is like that.

In some circumstances, and increasingly in the systems of today, information storage is under computer control. This may be because the volume of the information is very large or because the medium of storage makes access via computers inevitable. Another, more compelling, reason for an information storage system to be under computer control is when the information is the product of a data fusion process, gathered and created by computers, in parallel and simultaneously with being available for use.

2.1.2 Ephemeral Information and Situation Awareness

The information provided by a situation awareness system may change constantly. It may be subject to uncertainty and undergo continual update and revision. If

stable, well-known, factual information is regarded as immutable, the information from a situation awareness system can be rapidly-changing, partially-known and estimated, and can be regarded as ephemeral.

A ledger is a dynamic information system with update and revision that happens at a relatively slow rate. It is possibly the oldest type of information system and has existed for a very long time with little fundamental change [25]. The ledger is concerned with situation awareness in the sense that the owner of the ledger wishes to be aware of his or her financial situation and the information is ephemeral because each new entry changes the result. Whilst it is possible and convenient to keep a ledger using a computer, it is not essential and paper and pens can be equally effective. Also, the Performance and Effectiveness of a ledger is intuitively obvious and relates to timely and accurate recording of information and error free arithmetic.

Other things which might be the subject of a situation awareness system provide a greater challenge than that posed by the ledger and do not lend themselves to simple, intuitive effectiveness judgements. Examples of these include the airspace around a busy airport, a space mission, a battlefield, the stock markets, an automated factory and a complex industrial plant or machine.

Air traffic control, (space) mission control and battlefield systems might be classed as Command and Control systems (C^2 to borrow military terminology) because the purpose of the information system is to allow some complex operation to be directed in real time. C^2 systems began to emerge in their present form when high-speed long-distance communications came into being. Long-distance observation has been used since prehistory when fortifications were built on the tops of hills. Airborne surveillance was used in the French revolution when spotters ascended in hot air balloons and used telescopes to improve their powers of observation. But conventional means of recording and conveying the information were used.

During the first world war the aeroplane, telephone and telegraph meant that the observers were mobile and the observations of many could be relayed quickly to a central point. The range and reaction times of these information systems were pushed well beyond the limits of human perception and communication. By the time of the second world war, Radar had been invented and large numbers of human operators were employed continually updating dynamic maps of the theatre of battle in order to keep pace with the rapid and continuous stream of fresh information.

Workplace and plant Condition Monitoring systems emerged when the complexity of the workplace and plant and the degree of instrumentation required to monitor its state and confirm its error free running increased to the point where teams of operatives were required to carry out the task. A metropolitan railway network or the monitoring systems on a modern passenger jet are candidate applications for this type of system.

In a similar manner, Condition Monitoring systems to monitor the trends in Stocks and Shares have expanded as markets have grown, and with them the number of equities traded. The blackboards and the teams of clerks which once conveyed the state of the markets have been replaced with banks of computer screens.

For C^2 and Condition Monitoring systems alike, the impact of the recent and rapid rise of computers has been great. In modern examples of situation awareness systems much information is gathered by automated sensors and consolidated by computers into high level summaries for external consumption. This process of information gathering and consolidation is referred to as data fusion [82]. The emergence of automated, computer-based data fusion systems to do this is a relatively new phenomenon. Up to 1991 Linn, Hall and Llinas cited 54 documented examples of military application of such systems with various degrees of automation, mostly rudimentary [47]. Only three systems were operational; the rest were experimental. Since

then the rapid increases in computer processing speed and storage capacity have meant that progressively more complex data fusion tasks potentially may be subject to full automation. However, the investment required to implement each potential new system is large and progress has been measured and gradual.

The nature of the information in a data fusion system needs further explanation. Like a ledger, data fusion brings together many items, each with a little information, to yield higher information. A single transaction in the ledger does not reveal the bank balance but all transactions together do.

2.1.3 Uncertain Information and the Use of Hypotheses

Unlike a ledger, within a data fusion process, it is common for the source information to be subject to uncertainty. Using the ledger metaphor again, imagine a situation where not all the transactions are known and those that are known are uncertain. Imagine further that it is still required to estimate the bank balance despite this uncertainty. That is then the situation in which a data fusion process operates.

One way to deal with uncertainties such as these is to treat the items of information as hypotheses with levels of confidence or belief. An hypothesis that "object A is present" will have a belief associated with it which might be expressed as a probability of truth and an hypothesis of the form "the location of object A is x" will have a belief defined in terms of the probable error in x. The formation of hypotheses serves to build the required picture of the world and the statement of belief tells the user how reliable a basis for decisions that picture is.

When new information is gathered, it can result in changes to or removal of the existing hypotheses. We might change the value of x to y, or adopt the new hypothesis that it is object B present rather than A. The new information could result in changes to belief so that the error bounds on x change or the probability of object A changes. The

new information could result in creation of new hypotheses and beliefs so that we believe that objects A and B are both present.

Change can even happen when no new information is gathered. When we deal with dynamic information, beliefs degrade during any period when no new, relevant information is gathered. This can lead to information removal because the hypotheses are no longer believed. If the object A were moving but we were uncertain about its velocity, during any period when the object is unobserved, the error bounds on x would increase. After a certain time the estimate of x would be no better than a wild guess.

So, our situation awareness system may hold a mixture of immutable facts and ephemeral hypotheses. For example, an air traffic control system holds flight plans, air lanes and background details as if they were immutable facts. In the time scale of air traffic control they are approximately so. The system also holds other items such as the current position and status of each aircraft and these are ephemeral and subject to the uncertainties described above.

The detailed discussion so far has considered situation awareness systems and the information they provide. It has been implied that a trend toward complexity and the projection of situation awareness beyond the bounds of unaided human perception has led to systems for which an intuitive, objective judgement of effectiveness is difficult to achieve. To examine this further it is necessary to define effectiveness.

2.2 Definitions of Effectiveness for Information Systems

There is no obvious, single measure of the effectiveness of an information system For the purposes of this discussion four effectiveness factors are offered: access, quality of information, cost and benefit.

These factors, in some combination, are proposed as a basis for a definition of information system effectiveness. Each individual system will require a different

combination, in which the factors take on different weightings depending on the context in which the system operates.

In addition to these, feasibility is an issue. Whilst feasibility is not an effectiveness factor in the same way as those given above, a system which cannot be manufactured or one which attempts to gather information which is not available will never be effective. As an example consider a system to search space for extraterrestrial communications. If the transmissions exist in the form in which they are sought, it is a feasible system and there is a finite probability that the system will look in the right place at the right time and that probability is one indicator of the effectiveness of the system. On the other hand, if such transmissions do not exist, however long the system searches, it will never discover the fact of their non-existence with certainty.

2.2.1 Effective Access to Information

To a system's users, quality of access is extremely important. Availability, timeliness and user friendliness may be deciding factors in any measurement of effectiveness. A system which is not easy to access and not user friendly may fall into disuse unless it is the sole source of the information it offers. Similarly, if information is frequently unavailable or takes too long to arrive this can have the same effect. For situation awareness systems, the user might have no alternative source of information and poor quality of access can lead to important decisions neglected or delayed so that the safety of the aircraft is put at risk. In the air traffic control example, the timeliness of a decision, and hence the timeliness of the information to support that decision, can be critical.

If an information system serves an automated process, access remains an important issue. The auto pilot on an aircraft needs to know the current flight plan and the position, speed, heading and attitude (orientation) of the aircraft. If any of this information is out of date or unavailable, the consequences can be dire.

2.2.2 Effectiveness and Quality of Information

Quality of information is concerned with completeness, accuracy and relevance. An air traffic control system which routinely indicated the presence of aircraft without providing the customary background details would be of limited usefulness because the information would be incomplete. Similarly, an automated billing system which sent inaccurate bills or sent them to the wrong customers would be a source of embarrassment and loss of business to its owners. But, quality of information can be relative.

If the incomplete air traffic control record was all that was known about some off-course aircraft, that information would be of great value. Likewise, accuracy to the smallest unit of currency is required of the billing system whilst the air traffic control system might not measure an aircraft's position to an accuracy better than the size of the aircraft. With safe operating practises, occasional incomplete records and limited accuracy need not render the air traffic control system ineffective.

2.2.3 Cost-effectiveness of Information

The term cost-effective is in common use and usually implies that some comparison of cost versus benefit has been carried out. Affordability is an increasingly common effectiveness criterion which implies that cost has been compared with benefit and budget.

Waltz and Llinas classify cost as a Measure of Force Effectiveness (MoFE) for military C^2 systems [82]. We will see in section 2.3.4 that such measures are applied to the system and its operators. This is an important distinction since modern systems can provide very large amounts of information at very low cost. The cost of an operators time can be the most significant component.

In Decision Theory, cost is equated to the risk of making a decision [45]. The cost of a decision $d(Y)$ made on the basis of observations Y when the state of the system is X is written:

$$C = C(X, d(Y)).$$

The expectation of the cost is then minimised.

2.2.4 Effectiveness and the Benefits of Information

Benefit is concerned with enrichment, productivity and efficiency. A civic library may be said to enrich a community whereas a computerised business management system may allow its owners to do more business with less staff and so enrich themselves materially. The latter is more tangible and readily measurable but both contribute to the effectiveness of their respective systems.

Benefit is a higher level measure similar to a Measure of Force Effectiveness for military C^2 systems. The issues it addresses are concerned with the organisation which owns the system and the benefits that organisation receives in conducting its business or operation. It is possible for a system to give good access to high quality information at a low cost and still not benefit its owners because it is not relevant to the needs of their operation.

2.3 Measuring the Effectiveness of Information Systems

Given a set of factors contributing to the effectiveness of an information system, the next step is to measure them.

Various classifications of measures have been proposed. The NATO Code of Best Practice (COBP) on the assessment of C^2 takes a hierarchical view, that it attributes to the Military Operations Research Society, and which is supported by authors such as Waltz & Llinas, Vantrees and Girard [29] [70] [81] [82]. The hierarchy is given in the NATO COBP:

"... (1) Measures of Force Effectiveness (MoFE) which focus on how a force performs its mission or the degree to which it meets its objectives... ...

(2) Measures of C² Effectiveness (MoE) which focus on the impact of the C² System within the operational context... ...

(3) Measures of C² Performance (MoP) which focus on the internal system structure, characteristic and behaviour... ...

(4) Dimensional Parameters (DP) which are the properties or characteristics inherent in the physical C² systems..." [70].

Cyrus classified measures by their means of measurement rather than the level or importance of the features being measured [15]. Six types of measure were identified in current use. They were: user satisfaction, system usage, performance/usefulness, productivity, cost/benefit analysis (CBA) and value analysis. Figure 2.1 represents the approximate relationships between the types of measure identified by Cyrus and the effectiveness factors of section 2.2.
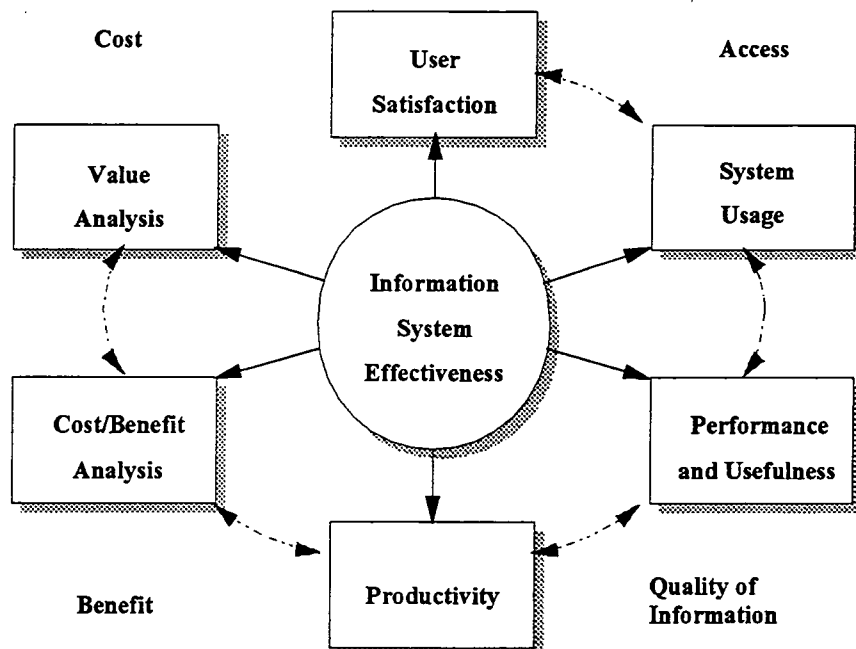
Figure 2.1: Information System Effectiveness Measures

The different types of Measures of Effectiveness are related to Effectiveness Factors.

2.3.1 User Satisfaction Measurement

User satisfaction measurements are obtained through opinion surveys. Users are commonly asked to complete questionnaires and the results are analysed to provide an indication of effectiveness. This approach to measurement is particularly common when the effectiveness of human organisations is evaluated. Cyrus cites Galletta and Lederer as the source of an evaluation of this approach to measuring effectiveness [15] [27].

User Satisfaction measures the quality of access and quality of information jointly. It attempts to measure the users opinion of the system in question and, to have a high opinion of a system, a user must be satisfied with its access mechanisms and feel that the information provided is worthwhile. A failure in either would cause the system to be deemed ineffective. Whilst these measures might be applied to data fusion processes, they could only be applied in the context of the higher information system.

2.3.2 System Usage Measurement

System Usage measures infer a system's effectiveness from how heavily it is used. It can be measured directly and thus may be regarded as more objective than User Satisfaction. However, its applicability is limited to systems where use is optional. Baroudi et. al. give an account of an empirical study of System Usage and Information Satisfaction [4].

Like User Satisfaction, System Usage measures of quality of access and quality of information jointly. Also like User Satisfaction, System Usage measures can be applied to data fusion processes, but only in the context of the higher information system.

2.3.3 Performance and Usefulness Measurement

Performance and Usefulness measures may be used to examine any quantifiable aspect of the system. A Measure of Performance (MoP) may be obtained from a parameter of the behaviour of the system in question. For our air traffic control example

we might measure response time from first detection of a new aircraft to availability of full information, or aircraft position and velocity estimation accuracy.

Usefulness (effectiveness) is measured in the presence of Standards of Performance (SoP). SoPs define levels of performance at which the system is deemed to be operating effectively. For instance, safe operation may demand that an air traffic controller is aware of all aircraft positions within a tolerance of x metres in any direction. Usefulness is concerned with whether or not this SoP is achieved. It may be expressed as a Boolean parameter (SoP achieved: TRUE or FALSE) or probabilistically (probability of SoP achieved $= p$). So, whilst the response time of our air traffic control system measures performance, timeliness measures usefulness.

Performance/Usefulness measures have been applied to the assessment and evaluation of military Command and Control (C^2) systems [1] [29] [78] [81] [82]. Van Trees, Girard and Waltz & Llinas, writing about such systems, use very similar definitions for their Measures of Performance (MoPs) and Measures of Effectiveness (MoEs) to those in the NATO COBP [29] [70] [81] [82]. MoPs are defined as parameters of performance which measure how well a system performs its internal functions. Effectively, the same definition used above. MoEs are defined respectively by Van Trees as the capability of a system to "... enhance the combat mission", by Girard as the: "... probability of successful accomplishment of a function", and by Waltz and Llinas as a means of quantifying: "... how a C^3 system performs its functions within an operational environment".

Whilst these definitions are somewhat broader than the definition of usefulness provided above, they are clearly based on similar concepts. Namely, a desired property of the system in question is defined and the measure is the result of some test which determines whether or not the property is present. We will see that this approach will

allow data fusion effectiveness to be examined in isolation from any higher information system.

2.3.4 The Measurement of Productivity

Productivity measures are familiar and traditional. They measure aspects of benefit and are concerned with whether or not a user of the system in question will contribute more because the system is there. This may be measured directly in terms of the output of an individual (work units/unit time) or indirectly in terms of the size of the team required to complete a particular task.

Cyrus cites Mason as a source of information on this subject [15] [52]. Whilst benefit is addressed by Van Trees, Girard and Waltz and Llinas in terms of Measures of Force Effectiveness (MoFE) for military C^2 systems which, according to Van Trees, measure how well: "...the force and the C^3 system perform their mission" which is again very similar to the definition in the NATO COBP [29] [70] [81] [82].

To obtain a measure of Productivity, the information system and its operators are assessed together. Thus, a data fusion process cannot be evaluated in isolation and cannot be effective in this sense. But, a failure in data fusion would cause the system to be deemed ineffective.

2.3.5 Cost/Benefit Analysis

Cost/Benefit Analysis (CBA) is a well established means of assessing whether a project or proposal is worthwhile. Its applications extend beyond the bounds of information systems assessment and it may be applied to (e.g.) a civil engineering project just as readily. The costs of procuring and operating a system are set against the anticipated benefits. Benefits are expressed in financial terms and the objective is to make a profit. The aim of cost/benefit analysis is to determine whether or not a project is likely to be cost-effective. Burke gives an account of CBA in the context of project management [12].

2.3.6 Value Analysis

Cyrus cites Keen who writes about Value Analysis as a less common variant of cost/benefit analysis [15] [39]. It attempts to determine the Affordability of a project and if it is affordable, minimise the cost. A model of the simplest, cheapest system which might satisfy the stated requirements is constructed and the costs and benefits assessed. If a fixed budget is exceeded the project is abandoned and the process ends or, if the benefits are deemed sufficient, the process ends. If the cost is acceptable but the benefits are insufficient, the cycle is repeated with a model of a more sophisticated and expensive system. Value analysis is iterated until one of the end conditions are reached.

Burke suggests that Affordability is a component of any CBA and as such does not regard Value Analysis as a separate approach [12]. Cost/benefit and Value Analysis do not measure effectiveness directly, rather they address the question of cost of effectiveness.

So far, this discussion has dealt with the systems and the information they provide, with the meaning of effectiveness and the ways in which effectiveness is commonly measured. In the next section, the application of Measures of Effectiveness to development of concepts and designs and the creation and deployment of new information systems will be examined.

2.4 MoEs and System Development and Deployment

A summary of the information given in this section was published at the 1997 Avionics conference in London [61]. A system is more likely to be effective if it is built to be so and there is an emerging consensus among system developers that this may be achieved through repeated assessment and evaluation throughout the development and deployment process.

Van Trees produced a table which he referred to as the "spectrum of evaluation techniques" when writing about the evaluation of C^3 systems and a very similar table

appears in the NATO COBP on the Assessment of C^2 [70] [81]. A generalised version is given in figure 2.2.
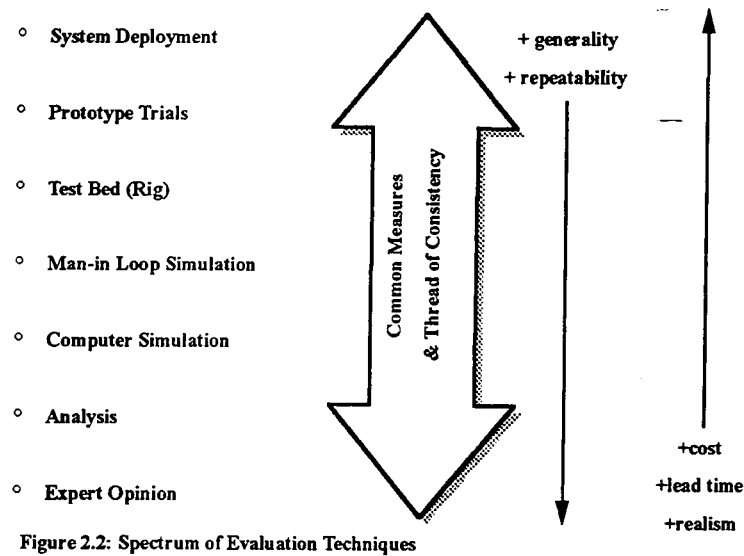
Figure 2.2: Spectrum of Evaluation Techniques

At the top of the diagram, the operational system is evaluated at high cost and long lead time to produce a highly accurate and specific result. At the bottom of the diagram, evaluation by expert opinion is relatively cheap, quick and general.

The techniques are listed in order of decreasing detail and complexity down the left hand side of the diagram. On the right hand side trends are shown in generality and repeatability, which increase down the page, and in cost, lead-time and realism, which increase up the page. Common measures and a thread of consistency are shown as central to the process of evaluation. Van Trees wrote: "The arrow ... conveys two important points. First, it is essential to have a common set of measures that are applicable and measurable across the spectrum of techniques. Second, these measures must give consistent results so there is a thread of consistency across the techniques."

Atkinson wrote about the integrated use of simulation and testing through a product life cycle [1]. The life cycle was defined:

concept → design → development → production → deployment

The stages of the cycle were:

• Concept, during which the nature, purpose, scope etc. of the system are defined;

•Design, during which the components and internal processes are defined;

•Development, during which the details of system manufacture are established;

•Production, when the components of the system are manufactured and assembled; and

•Deployment, when the system enters use.

Atkinson described initial simulations, developed solely from analytical work, used during the concept stage. These simulations were refined as initial technical data became available. At the concept stage, the importance of simulation outweighed that of testing.

As the product moved through design and development, prototype testing took over as the major activity. However, the simulations were updated as the test results became available. When the product moved into production and deployment, simulation again took a dominating role. It was used to assist in operations training and as a test bed for new ideas and performance queries.

Atkinson defined a hierarchy of simulator levels and corresponding scenario types and evaluations which possessed characteristics similar to Van Trees' spectrum of evaluation techniques [1] [81]. Scope and aggregation increased as upwards in the hierarchy. Fidelity and detail increase downwards. The critical characteristics of simulations were considered to be:

•Degree of representation, which refers to the level of fidelity and detail necessary to evaluate a particular feature;

•Practicality, which refers to the fidelity and detail which was achievable;

•Validation, which refers to the need to compare the simulation with any and as much information from the real world as is available; and

•Configuration control, which refers to the need to keep the simulation up to date as the product development progresses.

To manage the contribution that simulation made to C² system development, it was

proposed that all of the critical factors had to be managed.

A similar set of critical factors for the usefulness of a situation awareness system

Measure of Effectiveness (MoE) can be stated. Degree of representation refers to the

level at which system components are considered. Practicality refers to the level at which

components are identifiable and manageable. Validation is achieved by comparing MoEs

with observed outcomes. Configuration control means measuring effectiveness against

up-to-date criteria.

An approximate mapping of evaluation techniques to the system development

and deployment process is shown in figure 2.3. We have dispensed with the production

stage for the sake of simplicity. It is assumed to occur in parallel with product

development and deployment. Whilst this is a simple model of the process, it will serve

as a basis for discussion.

- Field Trials
- Prototype Trials
- Test Bed
- Man-in Loop Simulation
- Computer Simulation
- Analysis
- Expert Opinion

deployment
development
design
concept
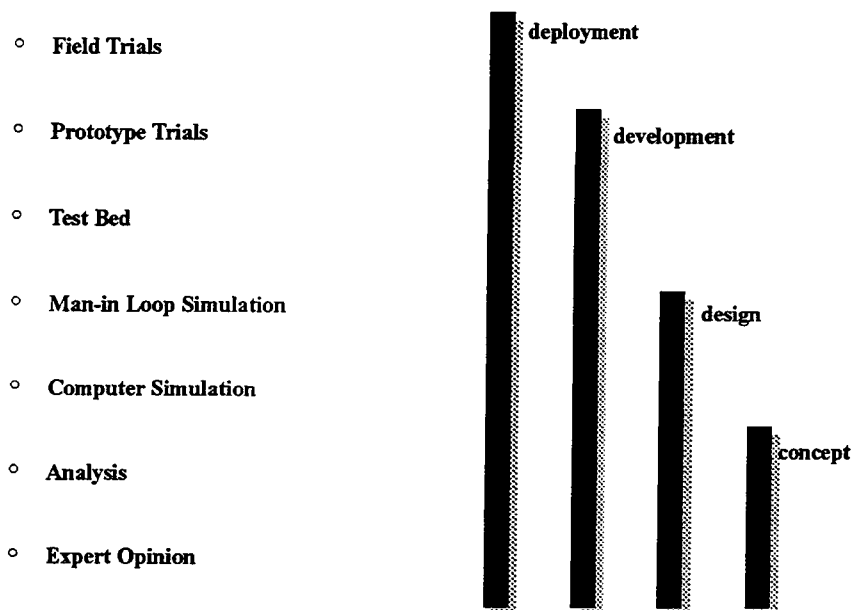
Figure 2.3: Evaluation Techniques vs. Life-Cycle Stage

Different approaches to evaluation are appropriate at different stages in the product life cycle.

At the concept stage, no detailed design exists and expert opinion and analysis

are the predominant evaluation techniques. These techniques remain available to the later

stages of the process. When a detailed design becomes available computer and man-in-the loop simulations are produced. These demand greater resources than analytical techniques and provide, in return, improved detail and realism. Simulations remain available to subsequent stages in the cycle.

During development the components of the system become available and a test bed system (an instrumented, laboratory-based, partial prototype) and/or a prototype system may be produced. Again the required resources and degree of realism increase. On completion of the development stage, the deployed system may be subjected to field trials if it is practical. In some applications field trials are not an option; a satellite-based system for instance.

The same MoEs are obtained at each stage in the life-cycle (use of common measures) of the system and the same outcome is demanded of each stage (thread of consistency). Given these properties, as the investment in a system increases, it is possible to confirm the design assumptions. Where design assumptions are ill founded, the fact is discovered and remedial action taken at the first opportunity. The "Smart Procurement" approach to military systems procurement reinforces this approach, ensuring early testing is focused on those assumptions deemed to hold the greatest risk.

This approach imposes constraints on the MoEs. Firstly, they must be tractable when analytic techniques are applied to them. This will allow analysis to be carried out in an objective and verifiable way. Secondly, MoEs which depend on knowledge of the evaluation scenario will present practical difficulties for the analyst. They could also be extremely expensive applied to field trials, demanding highly accurate instrumentation.

2.5 The Generic Properties of Measures

The NATO COBP for the Assessment of C^2 cites the "Modular Command and Control Evaluation Structure" (MCES) as an example of an attempt to establish an evaluation methodology for C^2 [70]. A shortcoming of MCES, pointed out in the COBP

is its lack of defined measures. Sweet wrote about the specification of measures for MCES and considered the generic properties of measures [78]. The steps comprising the MCES methodology are given in table 2.1.

Table 2.1: The Modular Command and Control Evaluation Structure [70]

Step	Procedure	Product
1	Formulate the problem	Problem statement and scenario
2	Bound the C^2 system	System elements
3	Define the C^2 process	System functions
4	Integrate the elements and functions	System architecture
5	Specify measures at boundaries	Functional measures
6	Generate data for measures	Values
7	Aggregate and integrate results	Analysis of results

Whilst MCES is interesting, it does not have a direct bearing on this work. This work might fit inside an evaluation performed using the MCES methodology. Of more interest are Sweet's generic properties of measures. It was proposed that measures should be selected to be: Mission oriented; Discriminatory, identifying real differences between entities; Measurable; Quantitative, assigned numbers or ranks; Realistic; Objective; Appropriate; Sensitive; Inclusive, reflecting the standards required by the objectives of the analysis; Independent, mutually exclusive with other measures; and, finally, Simple.

Clearly the properties are not independent and orthogonal. For instance, there is overlap between the criteria Measurable and Quantitative and similarly between the criteria Sensitive and Discriminatory. Other issues, such as conciseness are not included explicitly.

The NATO COBP gives a similar list properties referred to as criteria for validity and reliability [70]. For validity they are: mission oriented, realistic, appropriate inclusive and simple. For reliability: discriminatory, quantitative, measurable, objective and sensitive.

2.6 Effectiveness and Data Fusion

In present-day systems, the user need not be involved in the data fusion process nor interested in the detail of what is taking place. The user's interest is in the information provided and its quality. That information must be complete, accurate and relevant and any measure of data fusion effectiveness will take these characteristics into account. The timeliness and availability of information are also important issues and may also be regarded as features of data fusion effectiveness.

This work will define the performance and effectiveness (usefulness) of data fusion in terms of its information output. It will show that the key issues: availability, timeliness, completeness, accuracy and relevance can be addressed in this way.

Some of the issues, identified in this chapter, will not be addressed. Ease of access, "user friendliness", productivity and cost may only be evaluated in the context of the parent situation awareness system and its users. Techniques and methodologies exist to evaluate these issues and there is no reason why the system employing data fusion should require new or special treatment in these respects. Poor data fusion might cause a system to fail such an assessment but good fusion processes cannot guarantee success. The issues cannot be addressed by considering data fusion effectiveness alone.

2.7 Documented Data Fusion MoPs and MoEs

There are relatively few documented MoPs and MoEs for data fusion. Most documented examples warrant only simple MoPs expressed as graphs of error behaviour versus time to demonstrate some desired point [2] [8] [45] [51] [82]. For applications such as observation-to-object allocation or object classification, MoEs based on the

probability of a successful (or unsuccessful) outcome are common [3] [8] [51] [55] [60]. These approaches are adequate to illuminate issues surrounding algorithm design and behaviour but would be extremely cumbersome applied to the effectiveness of large-scale, complex systems.

Examples of more sophisticated MoPs and MoEs can be found. Blackman has developed an heuristic, rule-based, scoring algorithm for tracking filter assessment [8]. This provides a useful tool for confirming the appropriateness of tracking filter design but is specific to that application.

Kharbouch and Atherton proposed measuring the performance of data fusion using a scoring function which combined accuracy, time to acquisition and duration of sustained, uninterrupted tracking [44]. Waltz and Llinas produced a list of MoPs comprising: "... detection probability, false alarm rate, location estimation accuracy, identification probability, identification range, time from detect to transmission, communication time delay, sensor spatial coverage and classification accuracy..." [82].

The NATO COBP on the assessment of C2 systems gives examples of time and accuracy based measures of related system characteristics [70]. Time based measures were: time to perform task, response time to event, time to achieve target state, time on target, number of events pending (not yet processed) and timeliness of actions. Corresponding accuracy based measures were: accuracy of task, sensitivity to events, probability of error, time to recognise error, time to rectify error, knowledge of own status and quality of decisions.

MoEs for parameter estimation based on normalised error or on the negative exponential of its square have been used [2] [7] [8] [60]. The normalising factor in each case is an SoP and may be defined as the desired maximum error standard deviation. Whilst these measures are fully adequate for their documented applications, they demand knowledge of the evaluation scenario before they can be calculated.

Measures based in information theory have been used both as MoPs to characterise error behaviour and to indicate the utility of action within Sensor and Information Management processes [19] [51]. These measures offer the prospect of making statements about information quality which are both concise and informative.

2.8 An Approach to Data Fusion MoPs and MoEs

Information Theory and the theory of Stochastic Processes will form the bases to the approach to measuring effectiveness which is used in this work. This approach was presented in summary in the IEE Data Fusion Conference in Malvern in 1996 [59].

One may regard information quality as the degree of similarity between the reported situation and the true situation. Perfect information would allow decisions to be made which are not constrained by information error but, in reality, perfect information is not available. The error in the information, incorporating errors of inaccuracy, incompleteness, untimeliness etc., is the complement of similarity. So, if we measure the error we measure also the similarity provided that the definition of error incorporates all the elements of similarity that are sought.

However, we need to qualify these assertions. The reported or estimated situation must be defined in terms of relevant information only. Irrelevant parameters and variables must be excluded from any assessment and no relevant parameter may be omitted. Only then can we guarantee to reflect errors of incompleteness.

In terms of the NATO COBP on the assessment of C2 systems, the information error is our Measure of Performance which focuses "...on the internal system structure, characteristics and behaviour" [70]. In order to address the effectiveness of the data fusion process, we compare the information error with the information needs of the task to be performed. We refer to the level of information error which is tolerable, without disruption to the task, as the Standard of Performance for that task. When we compare the MoP and the SoP we are able to measure effectiveness which, according to the

NATO COBP focuses "... on the impact of the C2 systems within the operational context" [70].

The measures we use must be timely and located. They must be timely in the sense that if a task must be carried out at time t the MoE must refer specifically to that time. They must be located in the sense that if a decision must be made by an observer at location x the MoE must refer specifically to the information available at that location. If we take a prototype system and measure its performance, these issues concern only the instrumentation of the trial. If we require to produce a model of a system concept, we will see in Chapter 6 that these issues demand close attention.

When we define a measure of performance, the following points must be addressed. Firstly, it is axiomatic that the truth cannot be known in any realistic scenario independently of the reported information, otherwise there would be no need for a fusion process. There may be cases of highly instrumented trials when this is not so and when simulations are conducted it is clearly not so, but in general it holds. If the truth cannot be known, likewise the error cannot be known. However, using well-established theory we may predict the log-likelihood of error in such estimates [41].

Secondly, in many applications, information will be reported regarding many real-world objects simultaneously. Conveniently, log-likelihoods of error may be combined by summation and increase as the likelihood of error decreases (i.e. as the degree of similarity increases).

Finally, the log-likelihood of error observed at any instant, in a dynamic situation, may be subject to significant variation. For this reason, the expectation of the log-likelihood is more indicative of the capabilities of the data fusion process than any instantaneous measurement. We note that information entropy may be defined [75]:

$H = -E[\text{log-likelihood}]$.

Information entropy is proposed as the basis for MoPs, SoPs and MoEs for data fusion. It will be shown in chapters 4 and 5 that MoEs of this form are amenable to analytic techniques and may be obtained with or without reference to the true situation. In Chapter 6 models will be proposed to predict the entropies of various data fusion processes.

However, we will discover in Chapter 6 that the entropy of a data fusion system is predictable only when the system's behaviour is likewise predictable. It will be seen that systems are capable of displaying different behaviours in identical situations and that the switching and mixture of behaviours may be described in terms of a particular type of Stochastic Process known as an ergodic Markov process. The behaviours of the system correspond to the states of the ergodic Markov process and will be referred to as the Fusion States. It will also be seen that a data fusion system is fully predictable only when one Fusion State dominates its behaviour. When more than one Fusion State is present in the system behaviour the proportions of the mixture of states will be predicted.

This is analogous to the mixed models approach described by West and Harrison [83]. When forecasting future trends and events in a system capable displaying multiple behaviours, West and Harrison estimated the most likely current behaviour or mixture of behaviours to account for the observations to date. The models in Chapter 6 of this thesis will predict only the quality of the forecast and will base this prediction on assumed knowledge of the statistics of observations.

2.9   Emerging Uses of Data Fusion MoEs

So far we have considered the use of MoPs and MoEs as indicators in an evaluation process. The prospect exists to use MoPs and MoEs within a system to optimise some aspect of its own behaviour. The system would be constructed to incorporate a model of effective behaviour (like SoPs) and a model of the processes it

executes (MoP modelling). Any actions under internal control would then be selected and initiated to maximise effectiveness predicted by a comparison of these models.

2.9.1 Sensor and Information Management

Managed sensor systems allocate tasks to sensors and control the sensors operating modes to optimise the information output of situation awareness systems [8] [18] [19] [35] [48] [51] [64] [80] [82] [85].

The subject of sensor management was surveyed in the well-known paper of Popoli [64]. Popoli divided sensor management into two levels. The μ-level was concerned with the physical sensing environment and optimising the operating modes of the sensing devices to that environment. Woodward was among the first to recognise the possibilities of an optimised sensor device in the 1950's [85]. Whilst Woodward could not envisage using a computer to adapt the response of the sensor to the changing environment, his proposition that the transmissions of a Radar be designed to optimise the information return, anticipated μ-level sensor management.

The other level of sensor management, described by Popoli is the M-level, where the sensor tasks are allocated and scheduled to optimise the available information in some way. Blackman proposed an approach to task scheduling for Electronically Steered Array (ESA) Radar based on utility theory [8]. Utility was expressed in terms of MoPs concerned with estimation error standard deviation and probability of detection. More recently, Van Keuk also wrote about ESA Radar management [80]. Measures based in information theory have been used by Manyika and Durrant-Whyte to indicate the utility of action for M-level sensor management processes [51].

Information management is a more recent development. In a large system with multiple tasks, information management may be applied to optimise the information access for each task. Deaves used an information theoretic approach to manage the exchange of information between units connected by communications of limited capacity

[18] [19]. In a system of distributed operator stations and sensors, Deaves communicated that information predicted to be of greatest utility to its recipient. Using this approach he demonstrated a quantifiable improvement in the global situation awareness of the group.

In M-level Sensor Management and Information Management applications, the system has control over the sensor or the means of communication and can initiate new observations or transmissions when required. The management process calculates the likely utility of each possible next action by the available sensors or communications devices. The process then chooses the combination of actions with the highest utility. Thus, the system attempts to maximise its own performance or effectiveness. Systems which maximise performance may be referred to as "greedy" in the sense that better information is assumed good at any cost. The results of this thesis might be used to produce a system able to maximise effectiveness and conserve resources when an effective state is reached.

It has been shown that systems which operate adaptively under management achieve better performance than systems which operate to fixed task schedules [8] [18] [19]. However, there remain issues still to be addressed. The information-theoretic measures used to date have assessed the utility of an action based on the predicted information quality of the sum of inputs to the fusion process. The models which will be described in Chapter 6 will allow the information quality of the output to be predicted with the potential to improve performance still further.

Finally, effectiveness models to date have been based on the implementer's judgement. This thesis will propose an analytical basis for the development of models and a means of predicting whether effective operation will be achievable in a realistic scenario.

2.9.2 Self-monitoring Systems

Additional applications of this work are envisaged. It will be possible to create a system capable of reporting on its own information gathering performance. Given an estimate of the current situation, a model of system behaviour would predict a performance level. Failure to achieve this level would indicate malfunction or, for military C^2, possible information denial. Thus, future C^2 systems may be supervised by dedicated Condition Monitoring systems to beneficial effect.

2.9.3 Potential Future Applications

There is current interest in self-learning systems. A self-learning system is one which attempts to "train" a process (usually based on a neural network) to perform some function without the advantage of human intervention in the training activity [13] [34]. The system would have some model of desired and undesired behaviour and would modify itself iteratively in an attempt to optimise its own behaviour. The great difficulty with these systems is the creation of concise, verifiable behavioural models which are based in theory. The techniques described here might be applied for this purpose.

Looking further ahead, the concept of a conscious system has been proposed [57]. Conscious systems would be implemented in applications which required some robot agent to carry out actions autonomously with little or no immediate supervision. The agent would supervise itself by comparing the outcomes of its actions with a simulation or model of itself. Any significant divergence between the model and the real world would trigger remedial actions or alarms. Again there is a requirement for reliable, concise modelling founded in theory and this presents a possible application for this work.

These applications described in this section will not be investigated as part of this work and are left as topics for future research. The analytic approach to modelling information theoretic MoEs is described in Chapter 6.

3 Data Fusion: A Survey

The requirement for information to perform a task or make a decision might be defined in terms of its content and the amount of uncertainty or information error which is tolerable. Each sensor observation, has a particular content and a particular information error and it is possible to map many observations together to determine their overall information and information error. If the required information is available and the error within tolerated limits, the task can be performed or the decision made with the expectation of a successful outcome.

The uncertainties in a system can be static if the system is non-random. However, in a random system, they are dynamic and uncertainty and expected information error increase during any period when no new observation is made. There may also be an intrinsic ambiguity in the situation of interest so that the appropriate mapping is unclear and several different mappings of the information appear to have similar likelihoods.

The purpose of data fusion algorithms is to map and extract information and minimise the resulting levels of uncertainty whilst making due allowance for any randomness or ambiguity which is present. Sometimes, that ambiguity may be represented adequately by a single hypothesis and process model. On other occaisions, multiple hypotheses and process models are required [83]. This chapter considers the data fusion problem and numerous documented approaches to its solution.

3.1 Background

Firstly, the background to data fusion will be considered. To understand a data fusion process it is necessary to understand the information requirement and the act of observation as a means of fulfilling that requirement. When observations are individually incapable of satisfying the information requirement or when continuous renewal of the information is made necessary by the stochastic nature of the observed objects' behaviour, the role and purpose of the data fusion process become apparent.

3.1.1 Attributes, Observations and Observation Uncertainty

Shafer, in his discussion of evidential reasoning, talks about a frame of discernment, interpreted as the information that may be discerned from the available body of evidence or data [74]. The degree to which data fusion exploits its frame of discernment and extracts the available information might be regarded as a basic parameter of system performance. But, just because information is available, that does not make it useful information; and information that is outside the frame of discernment may be vital. We require a definition which can address information requirement as well as information availability. When the required information is delivered, the fusion process will be regarded as effective. We define the attribute space as follows:

A system under observation has a number of parameters and variables that, together, comprise the information required to allow some decision to be made or some task to be carried out. The parameters and variables may be of any type and will be denoted as attributes of the system, for example: identity and location of objects. The parameters and variables will occupy an attribute space.

Attributes are associated with objects and the state of each object is given by a vector or m-tuple of attributes in attribute space. The state of attribute space is the assembly of object vectors. Manyika and Durrant-Whyte use a convenient notation for this [51]. The state of attribute space X is given by a vector of dimension $n \times m$ comprising n objects and m attributes, with the jth object possessing a vector of attributes x_j.

$$X = \begin{bmatrix} x_1^T & \ldots & x_j^T & \ldots & x_n^T \end{bmatrix}^T \text{ and } x_j = \begin{bmatrix} x_{j1} & \ldots & x_{ji} & \ldots & x_{jm} \end{bmatrix}^T.$$

It is axiomatic that the true state of attribute space cannot be known with absolute certainty, only an estimate, $\hat{X}$, will be possible. Given the above definition of attribute space, the distribution of errors in $\hat{X}$ can serve to define the quality of the

information it holds, provided our definition of error can be made to encompass all deviations from full and perfect information. This distribution of errors will form the basis of the information-theoretic Measures of Performance (MoPs), Standards of Performance (SoPs) and Measures of Effectiveness (MoEs) proposed by this work.

Next we will consider the observation process. Manyika and Durrant-Whyte describe global and nodal frames of discernment within a multiple sensor system in which the nodes are sensors and the nodal frames of discernment are subsets of the global frame of discernment [51]. As was the case with Shafer, Manyika and Durrant-Whyte are concerned with available information and once again, we require to make a distinction between available and required information. We will define the observation space:

Observations, made using sensors, will estimate some or all of the required parameters and variables, and the estimates will occupy an observation space that will map onto the attribute space resulting in an estimate Y of X.

Generally, observations will be incomplete and/or partially overlap, and will be subject to errors that may be independent or correlated. The observation operator Ω operates on X so that :

$$Y_k = \Omega_k(X)$$

for the kth observation. Associated with an observation operator Ω_k will be an error $\varepsilon_{\omega k}$. An individual object j may or may not have an estimate of its attribute vector $y_j = \omega_k x_j$. If such an estimate is present, associated with it will be a transformation H and an error ε such that $\omega_k x_j \equiv H_k x_j + \varepsilon_{k,j}$. When no such estimate is present, a large standard error will represent our ignorance. To clarify this, consider the example of figure 3.1:

A sensor observes the range and azimuth angle (horizontal polar co-ordinates) of objects. The errors in the observations lead to uncertainty about the observed objects. This uncertainty was present both in the observation space of the sensor and in attribute space. Manyika and Durrant-Whyte use a geometrical representation of uncertainty that will be discussed in Chapter 5 [51]. The representation derives an ellipsoidal region of uncertainty, centred on the observed or estimated position, from a singular value decomposition (SVD) of the error covariance matrix. We will use this representation to visualise the observation process.

Figure 3.1: Observation Errors in Observation Space
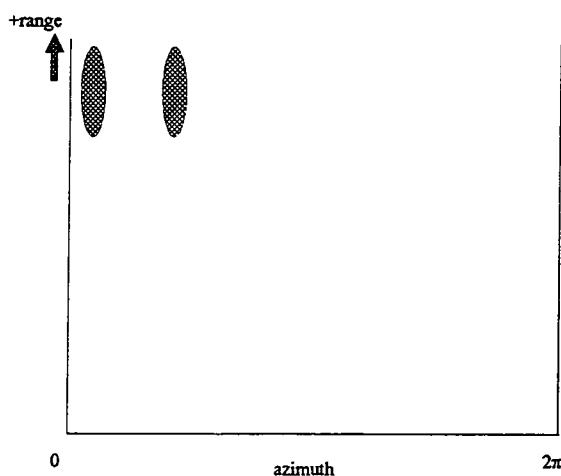
The uncertainty in sensor observations represented as ellipses in observation space.

The ellipses in figure 3.1 are centred on the locations, Hx, of two objects observed by the sensor. The ellipses correspond to a one-sigma region around the reported locations and represent the distribution of errors, ε, about the resulting estimates. The observation space of the sensor may be visualised in this manner.

It is implicit in this work that information is being gathered and fused in order to allow some task to be performed. If the task in hand requires the locations of objects to be known in three dimensions a corresponding attribute space may be defined.

Say the attribute space is the location of objects defined with respect to a

right-handed, 3-dimensional Cartesian axis set with the sensor position as origin. Then

let the x and y axes of the attribute space lay in the horizontal plane with the x axis

aligned with the azimuth datum of the observation space and let the z axis of the

attribute space point upwards. The mapping of the observations onto this attribute space

results in Figure 3.2.

Figure 3.2: Observation Errors in Attribute Space
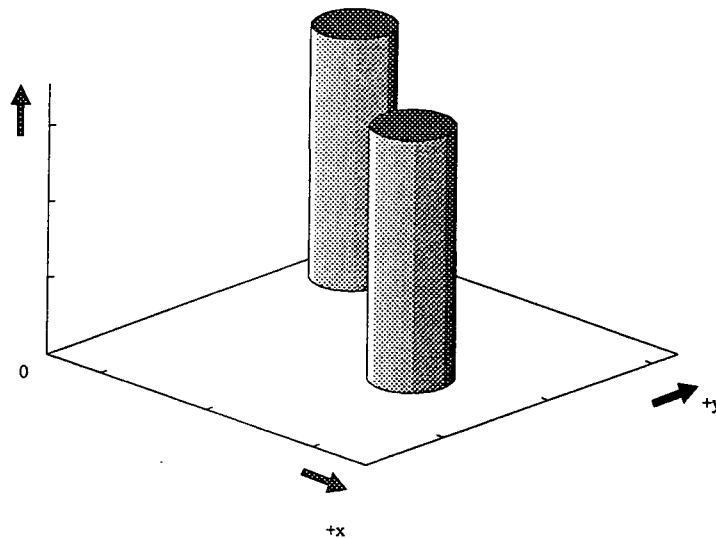
| The same uncertainty (as Figure 3.1) in the same sensor observations represented in attribute space. |

The observation operator Ω may be visualised in this way. The observations

provided information with respect to the horizontal plane only. The approximate

transformation of these error distributions into error distributions in the x, y plane was

simple. However, we knew nothing about the z co-ordinates of the observed objects.

Manyika and Durrant-Whyte would depict this axis of the error ellipsoid as having

magnitude ∞ [51]. Alternatively, the errors in z could be represented by a uniform

distribution from 0 to ∞ but both of these notations would overestimate the uncertainty

in z. In practice, there is a range of interest for every variable and z is shown as

uniformly distributed between its practical limits (0 and z_{max}). This anticipates the

discussion of models of ignorance that will be found in Chapter 6. In that chapter, issues

surrounding the effects on information quality of full or partial ignorance will be discussed in detail.

In the above example, the attribute space was made up of continuous random variables and the observation spaces were related to attribute space by geometrical transformations. The same approach and notation can be applied to discrete random variables but would not result in as clear a geometrical representation.

Discrepancies between attribute space and observation space are recognised by convention in linear filtering and non-linear filtering [2] [8] [38] [42]. The standard Kalman filter will be discussed in section 3.2. It is an algorithm to estimate the parameters of a Dynamic Linear Model (DLM) time-series and incorporates a mapping function between observations and estimates. For a standard Kalman filter the observation is a linear function of the estimate, often just a sub-vector. For an extended Kalman filter the mapping function is non-linear and usually more complex. The details of these and other approaches will be dealt with in Section 3.2.

Manyika and Durrant-Whyte allow for discrepancies between attribute space and observation space [51]. Writing about object classification applications, Manyika and Durrant-Whyte suggest two approaches to dealing with these mismatches when they occur:

●to "pad out" the observations with non-discriminating dummy values or probabilities so that the observations appear to be drawn from the global frame. This is recommended where the discrepancies are small and infrequent.

●to allow each node to use its own frame, constructing the global frame only at the point of sensor fusion. This approach is recommended in situations where the discrepancies are large and widespread.

The latter approach is the more general and an extended form is used here. The approach taken by this work does not restrict the observation space to be a simple subset

of the attribute space. More complex mappings are allowed. Object classification is described in more detail in Section 3.2.

3.1.2 Fusion of Non-random Attributes

Sometimes the behaviour of X over time is fully determined. In these circumstances, if the state of X is known with certainty at time t_i, its state at any later time can be predicted with certainty and we will represent X using a non-random model. Satellite tracking might be regarded in this way.

Let Φ denote the state transition that predicts $X(t_i)$ given $X(t_{i-1})$:

$$X(t_i) = \Phi X(t_{i-1})$$

Sensor Fusion estimates the state of the attribute space, $\hat{X}$, based on all the available measurements and knowledge of the observation processes. The fusion operator F operates on Y so that for k observations made at times $t_1 \ldots t_k$ is:

$$\hat{X}(t_k) = F(Y_1(t_1) \ldots Y_k(t_k) | \Omega_1 \ldots \Omega_k, \Phi, t_1 .. t_k) \tag{3.1.2.1}$$

Assuming that $\hat{X}$ is a consistent estimator of X, each new observation tends to improve the quality of information and, by repeated observation, the expected error in $\hat{X}$ may be made arbitrarily small. This is represented in figure 3.3 where the discontinuities coincide with new observations $\left(\Omega_1, \Omega_2, \Omega_3 \right)$ and between observations information is neither gained nor lost. We have assumed in Figure 3.3 that a single-valued, scalar measure of information error exists. This assumption will be examined in more detail in Chapter 4.

Figure 3.3: Information Error in a Non-random Static System

info error (vertical axis)

Ω1

Ω2

Ω3

0

+time
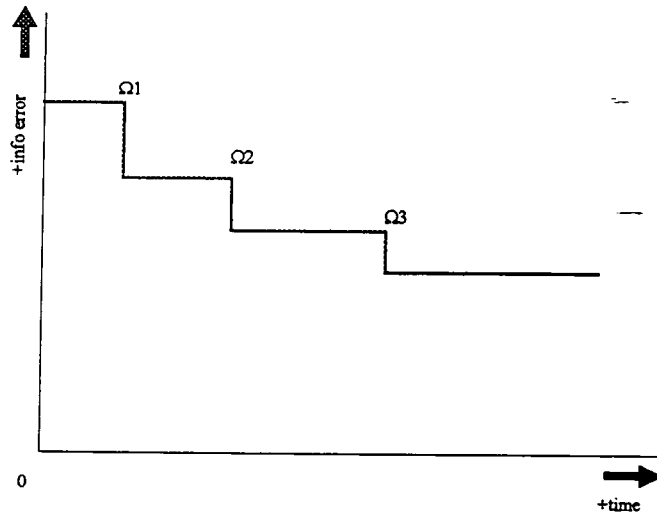
| When a non-random system is observed, each new observation improves the information quality. |

Returning to the example of the previous section, it is possible to visualise the affect of the fusion process. Suppose a second sensor, also located at the origin, is observing the same objects and measures the azimuth and elevation (spherical polar direction) of the objects, its observation space can be represented by Figure 3.4.

Figure 3.4: Observation Errors in Observation Space

$\pi/2$

elevation (vertical axis)

0

azimuth

2π
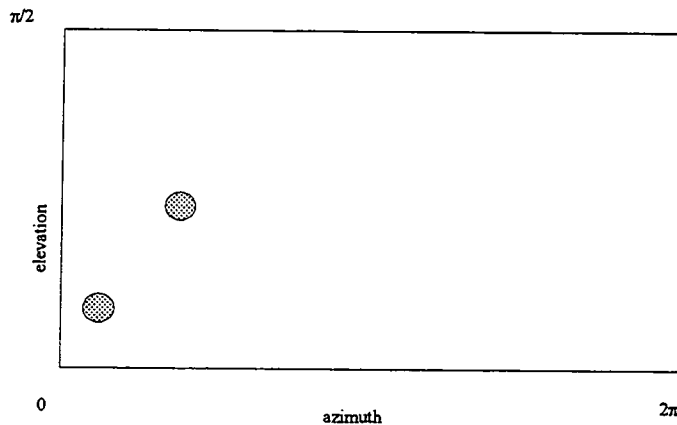
| The same objects (as Figure 3.1) are observed by a second sensor in a different observation space. |

The second sensor has a different observation space and a different mapping will be observed when the new observations are drawn in attribute space.

Page 54

We now have two views of the same situation. If we map the observation errors from both sensors onto attribute space we get the picture of figure 3.5.

Figure 3.5: Two-Sensor Observation Errors in Attribute Space
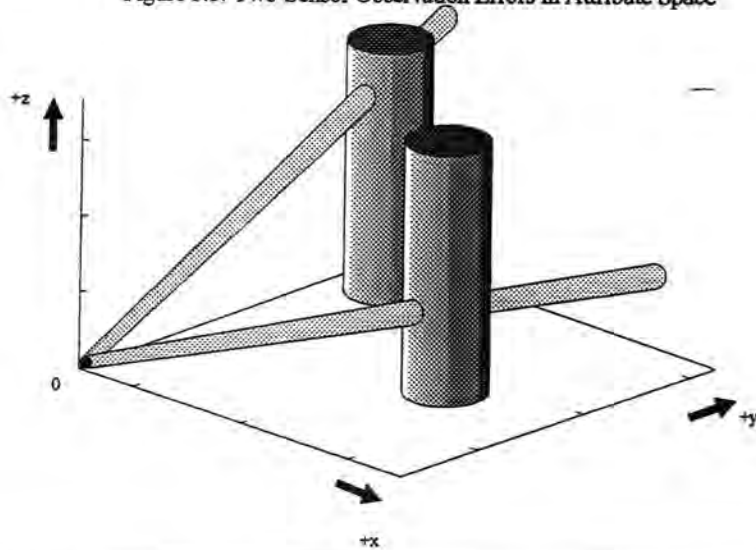
When the observations uncertainties from both sensors are mapped onto the attribute space together.

In figure 3.5 the errors in the observations from the second sensor are represented by narrow beams radiating from the origin. The error in the unobserved object range is represented by a uniform distribution from 0 to r_{max}. Common sense says that the objects are located where the error distributions intersect; data fusion calculates the expected locations and the fused error distributions.

After the fusion operator, F, is applied, the uncertainties corresponding to the estimation errors in the fused information are represented by figure 3.6.

Because the state is non-random, the fusion process is not sensitive to time. The observations could have been simultaneous or separated by a significant interval and similarly the fusion process could have been carried out immediately on receipt of the last observation or after a delay. The information value of the observations and the output of the fusion process would be the same in all cases.

Figure 3.6: Two-Sensor Fused Errors in Attribute Space
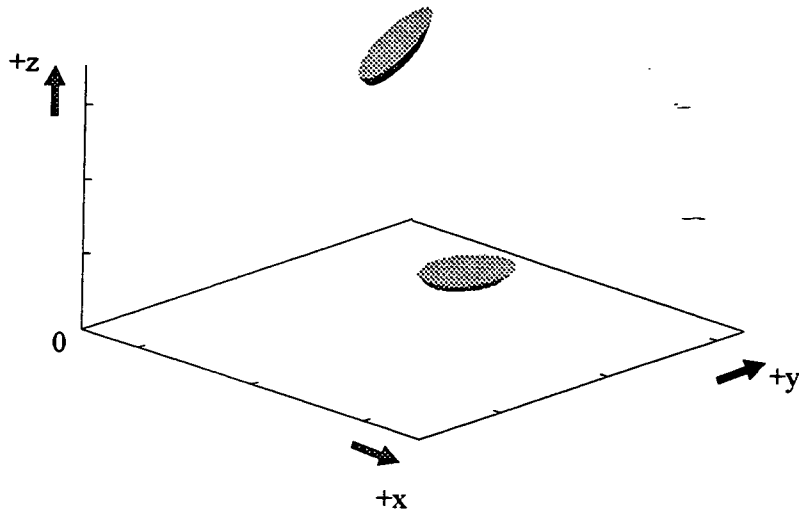
When the observations from two sensors are fused, uncertainty about the locations of objects is reduced greatly.

3.1.3 Fusion of Random Attributes and the Growth of Uncertainty

If X varies stochastically over time, any predicted state is subject to extra uncertainty. For example, the state of a vehicle under human control may change at any time when the controls are operated. To an observer, these changes appear to occur at random and we will represent X using a random model. X may undergo change, unobserved, at any time over the interval t_{i-1} to t_i:

$$X(t_i) = \Phi X(t_{i-1}) + V(t_i)$$

V denotes the stochastic variations which X undergoes and other notations are as in section 3.1.2. If the observations were made at times $t_1...t_k$:

$$\hat{X}(t_k) = F\big(Y_1(t_1)...Y_k(t_p)|\Omega_1...\Omega_k, \Phi, t_1..t_k, V(t_1)...V(t_k)\big) \qquad (3.1.3.1)$$

The fusion process now depends on V and the observations acquire different weightings depending on their age. The more recent an observation, the more likely that observation is to contain information regarding the current state.

In this situation, the quality of the information in our fused estimate degrades with time during any interval without an observation. This is represented by figure 3.7

where the discontinuities coincide with observations and the quality of the information is seen to degrade between observations.

Figure 3.7: Information Error in a Stochastic System
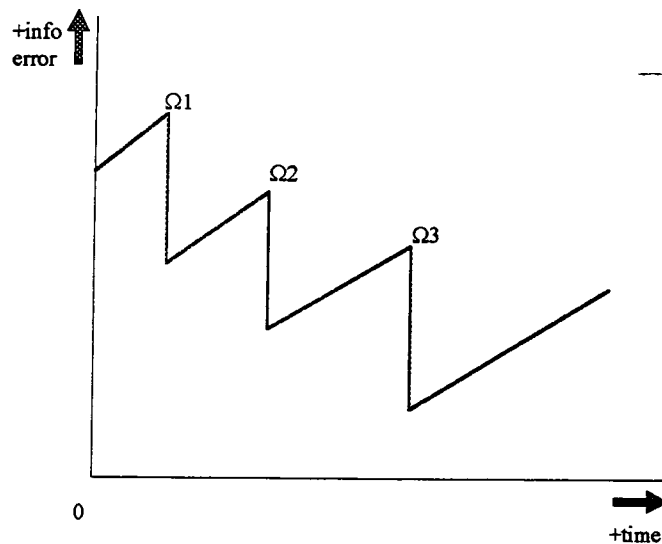
When new observations are fused, uncertainty is reduced. Between observations uncertainty grows. This is characteristic of a random system.

Again we have assumed that a single-valued, scalar measure of information error exists. When the fusion process is delayed similar degradation of the information quality of the observations occurs. Thus, timely processing of new observations will have a positive effect on information quality and long intervals with no observations will have a negative effect. In fact, if an interval without observations is sufficiently long, the effect can be a return to near-ignorance.

Models of stochastic systems are considered in detail in Chapter 6. The modelled information state is such a system is that in which an equilibrium is reached between the rate of information arrival in new observations and the growth of uncertainty due to random behaviour.

In many real world applications, the attributes are a mixture of non-random and random. For example, in the air traffic control example of Chapter 2, the identity of each aircraft is strictly non-random and its flight plan is usually so. However, the current

position and motion of the aircraft are attributes subject to stochastic variation. Thus, they require frequent updates and timely processing. Whilst our knowledge of all the attributes is subject to random error and uncertainty, when the attribute value itself undergoes random variation, we will develop models, based on equation (3.1.3.1), representing the stochastic variations explicitly. That is:

$$\hat{X}(t_k) = F\big(Y_1(t_1)...Y_k(t_p)|\Omega_1...\Omega_k, \Phi, t_1..t_k, V(t_1)...V(t_k)\big)$$

Where V will be defined to have a null effect on non-random attributes.

3.1.4 Fusion in the Presence of Association Ambiguities

The situation depicted in figures 3.1 to 3.6 is characterised by unambiguous data associations. The objects are most likely to lie where sensor uncertainty regions overlap and only one permutation of overlaps exists.

This unambiguous arrangement of objects is not guaranteed. If the same objects were observed with similar azimuth attribute estimates, the clearly interpretable situation of Figure 3.5 would change to the ambiguous one of Figure 3.8.

Figure 3.8: Correlation Ambiguity
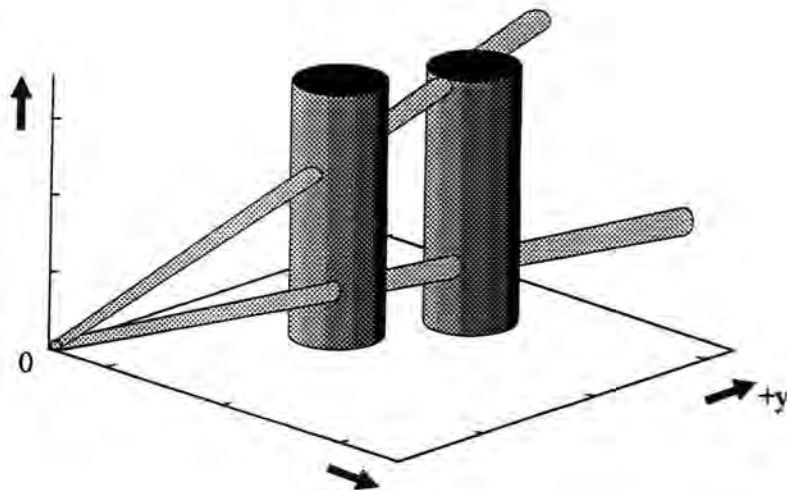
Unambiguous fusion is not guaranteed. Here two objects give rise to four intersections of the uncertainty regions.

The correct data fusion operation is no longer obvious and two association hypotheses exist which lead to different interpretations of the data. This is represented in Figure 3.9 by the light and dark object estimates which were derived from alternative interpretations of the same observation data.

The subject of Data Association is very important in any discussion of data fusion. It will be seen as an important topic in section 3.2 where data fusion algorithms will be discussed, in Chapter 6 where the Simulation and Modelling for this work are described and in Chapter 7 where the case study results are given.

Figure 3.9: Fusion in the presence of Correlation Ambiguity
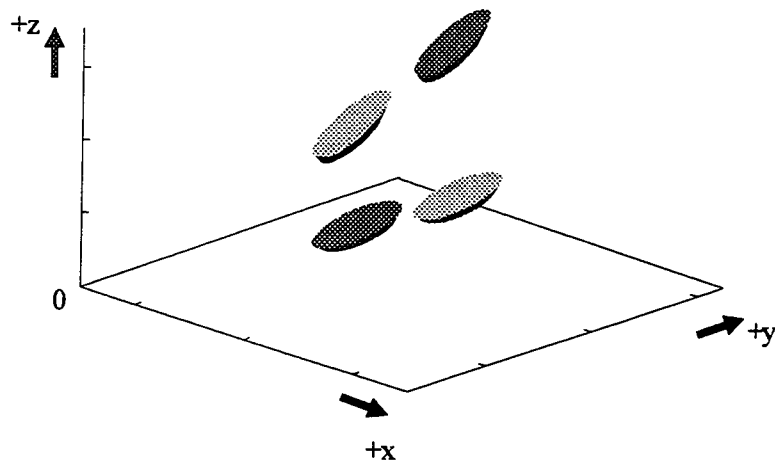
Two objects give rise to four possible object estimates. It is not clear whether the light or dark pair of objects gives the true interpretation of the observations.

3.2 Data Fusion Algorithms

So far, data fusion has been regarded as a "black box". Sensor observations are put into the box and consolidated information comes out. Much has been assumed about the properties of the fusion process and as yet no justification has been offered. This section will consider in more detail the data fusion process and how it is carried out.

First, we will discuss briefly the process of selecting an effective data fusion process from the many that will be described in the following paragraphs. A Bayesian model will be adopted as a basis for this discussion [2] [8] [51]. Given a set of

observations Y we will estimate the state of X using a fusion operator F. The task is to choose F so that $\hat{X}$ is optimal in some sense. The Bayesian model requires that the set of fusion operators from which we select should be mutually exclusive and exhaustive. Mutual exclusivity requires only that the operators be distinct alternatives and exhaustivity requires that we consider all feasible operators, where feasibility would be defined in terms of the practical constraints of the application.

We have proposed in Chapter 2 that performance and effectiveness of the fusion process be expressed in terms of the uncertainty surrounding $\hat{X}$. Hence, if we choose F to deliver a maximum a posteriori probability (MAP) estimate of $\hat{X}$, we minimise the uncertainty in question. Whilst other uncertainty-based expressions of optimality exist, they do not relate so readily to our chosen definition of effectiveness. If F were chosen to maximise the likelihood of the observations (i.e. $P(Y|\hat{X})$), this would deliver a maximum likelihood estimate (ML). Sometimes these definitions of optimality produce identical results, but in general they do not.

By Bayes' rule, the posterior probability of $\hat{X}$ given the set of observations Y is:

$$P(\hat{X}|Y) = \frac{P(Y|\hat{X})P(\hat{X})}{P(Y)}$$ (3.2.0.1)

Recalling equation (3.1.3.1), we choose the MAP operator F so that:

$$F_{MAP} = \arg\max_{\forall i} \left\{ \frac{P(Y|F_i(Y|\Omega,\Phi,t,V)) \times P(F_i(Y|\Omega,\Phi,t,V))}{P(Y)} \right\}$$ (3.2.0.2)

The F_i's are the set of feasible fusion operations that could be performed.

To calculate the MAP operator, the prior probabilities of $\hat{X}$, $P(F_i(Y|\Omega,\Phi,t,V)$, must be known. If they are not known, a ML operator may be obtained:

$$F_{ML} = \arg\max_{\forall i} \{P(Y|F_i(Y|\Omega,\Phi,t,V))\}$$

Each F_i might be based on a different hypothesis regarding the array of objects that inhabit the attribute space and/or regarding the time series of observations each object gave rise to. Each F_i would result in a different $\hat{X}_i$ with the potential for different arrays of objects, with individual objects in different positions and appearing to hold different identities. Further variations in F_i could arise from variations in hypotheses regarding V, the random behaviour of the objects or, in some cases, Ω, the observation process of the sensor. When evaluating system concepts, variations in F_i could arise from different arrangements of sensors, computers and communications links and from different choices of algorithm.

The operator F_{MAP} yields the most likely $\hat{X}_i$ based on the evidence of the observations, Y, and on any prior knowledge we possessed. Recalling the discussion of the previous Chapter we proposed to measure the performance of the data fusion process through the information error present in the final $\hat{X}$. Since we have defined X to contain all the information that is required to be observed, under relatively few assumptions (to be discussed in the following sections), we would have maximised the performance of our system in terms of the likelihood of the solution through this approach. Most texts on data fusion concern themselves with optimising $\hat{X}$ in this way.

This work is concerned with the effectiveness of data fusion and we must recognise that it cannot be guaranteed that an optimal system will be effective. Effectiveness depends on the achievement of Standards of Performance. These relate to the decision or task in hand and are independent of the observation and fusion process. An optimal fusion of the available observations might still fall short of these standards.

Conversely, there may be many fusion operations that satisfy our standards and optimality may be replaced with one of many effective approximations. In such circumstances we might write:

$$F \in \{F_E\} = \arg_{\forall i} \left\{ \frac{P(Y|F_i(Y|\Omega,\Phi,t,V)) \times P(F_i(Y|\Omega,\Phi,t,V))}{P(Y)} > P(\text{SoP}) \right\}$$

where the SoP is expressed in terms of the probability of the fused data and $\{F_E\}$ is the set of effective fusion processes.

In the remaining sections of this Chapter, we will look in more detail at the components of F and at algorithms and techniques that have been applied to calculating or approximating the optimal solution of equation (3.2.0.2). We will consider four classes of algorithm:

• Tracking - where time series of observations of numeric attributes are combined to form tracks.

• Track Fusion - where tracks are combined to form higher tracks.

• Object Classification - where observations of non-numeric attributes are combined to form higher estimates of those attributes.

• Data Association - where the mapping of observations to objects is created.

3.2.1 Tracking Algorithms

There are many applications in which the sensors provide a time series of observations of numeric attributes. In a tracking algorithm, the observations from the time series are combined, recursively, to form tracks.

We will use the term track to denote any estimate of a dynamic system derived from a time series of observations. Whilst much of the published material referred to here is concerned with applications of aircraft tracking, tracking has other applications. For example, in the financial world, investment decisions in tracker funds are based on algorithms of this type. In general, a track is a collection of numeric attributes and their time derivatives describing the world-line or path of some object through space and time.

When the observation process or the dynamics of the system being observed are subject to uncertainty, tracking algorithms often exploit prior knowledge, expressed as a

model or models of the uncertain observation or dynamics, in order to maximise the information gain from each new observation. Models may be single or multiple, depending on the complexity and degree of prior knowledge of the uncertain behaviour, or on the degree of modelling fidelity required of the tracking algorithm. A single model represents the extra errors introduced by the uncertainty as if they were always drawn from the same (usually Gaussian) distribution [2] [8] [76].

A multiple model tracker may be used, either when the uncertain behaviour is known to change during the life of the track, or when the uncertain behaviour is unknown and must be learned as part of the estimation process [10] [13] [16] [50] [83]. Examples of all of these cases will be discussed in the following paragraphs.

The purpose of tracking is to estimate the state of parameter x, given Y, where $Y = \{y_1, y_2, \ldots, y_k\}$. The theoretical basis for modern tracking algorithms is found in time series analysis. Kendall, Stuart and Ord describe an auto-regressive moving average (ARMA) process in which the value of a scalar parameter of a stationary time series x at time k depends on its previous p values and a random process ε_k, with zero mean and variance σ_ε^2, and on a moving average of the previous q random processes [42]. This is called an ARMA(p, q) process and may be written as:

$$x_k - \psi_1 x_{k-1} - \ldots - \psi_p x_{k-p} = \varepsilon_k + \theta_1 \varepsilon_{k-1} + \ldots + \theta_q \varepsilon_{k-q} \qquad (3.2.1.1)$$

The stationarity of the time series means the parameter value in (3.2.1.1) undergoes a characteristic oscillation. The current value of x depends on its previous p values and the current estimation error depends on the error in the current observation and in the previous q observations. Parameter values before p and observations before q are redundant because they are repeated.

The auto-regressive integrated moving average (ARIMA) model extends (3.2.1.1) to non-stationary time series where the mean parameter value "wanders" along

a path which is no longer oscillatory. In the ARIMA model a transformation is applied to the parameter to achieve stationarity. The deterministic wandering motion of the average parameter is removed by the transformation, leaving only an underlying oscillatory motion.

If (3.2.1.1) were applied directly to a non-stationary time series where the parameter underwent little or no stochastic variation, truncation at p states and q observations would exclude previous information not diminished in relevance by the passage of time. Serial correlation in the information error would be slow to die out. The integrating transform of the ARIMA model diminishes the correlation introduced in this way.

When the parameter undergoes significant stochastic variation, previous states and observations diminish in significance as they get older and eventually become insignificant. Truncation of the series at p previous states and q measurements in a direct application the ARMA(p, q) process can offer a good approximation to the process of object tracking in a stochastic system that would not depend on recursive computation.

Kendall , Stuart & Ord cite Kalman and Harrison & Stevens as the source of an alternative, recursive formulation to the complete ARMA model of equation (3.2.1.1) known as the Dynamic Linear Model (DLM) [32] [38]. The DLM is described in detail by West & Harrison [83]. It is defined in two stages and comprises an observation and a system equation respectively for the kth observation of a single object :

$$y_k = Hx_k + \varepsilon_k \text{ and } x_k = \Phi x_{k-1} + v_k.$$

Where H is related to the observation operator of section 3.1.1 such that $Hx + \varepsilon \equiv \omega x$, Φ is the state transition and v the random variation described in section 3.1.3. The first equation may be interpreted as an observation y that is some (possibly partial) mapping, H, of x with error ε. The second equation relates to the time behaviour of x that is modelled as stochastic.

It is assumed that H and Φ are known and that ε and v are white, zero mean error vectors with covariance matrices R and Q respectively. In the notation of West & Harrison, this is DLM(H, Φ, R, Q). The DLM has the big advantage over the ARMA and ARIMA models that the state of interest, x, no longer needs to be directly observable as long as the mapping H on to observation space is known. It has the extra constraint that ε and v are assumed white.

In control theory, it is usual to solve a system based on a DLM (H, Φ, R, Q) using the Kalman filter (Kf) algorithm. When $Q = 0$ the Kalman filter also serves to describe object tracking in a non-random system. When $\Phi = I$ it represents a static system. When $Q = 0$ and $\Phi = I$ it is equivalent to a Wiener filter.

The Kf equations are executed recursively:

$$x(k, k-1) = \Phi x(k-1, k-1)$$

(3.2.1.2 a)

$$P(k, k-1) = \Phi P(k-1, k-1)\Phi^T + Q$$

(b)

$$K = P(k, k-1)H^T[HP(k, k-1)H^T + R]^{-1}$$

(c)

$$x(k, k) = x(k, k-1) + K[y_k - Hx(k, k-1)]$$

(d)

$$P(k, k) = [I - KH]P(k, k-1)$$

(e)

where I is the identity matrix of dimension m, $x(k, k) = \hat{x}_k$, our estimate after k observations, and $P(k, k) = P_k$, its error covariance. Equations (3.2.1.2 a) and (b) extrapolate the existing estimate and its covariance to the time of the new observation, equations (c) and (d) calculate an MMSE combination of estimate and observation and equation (e) calculates the covariance of the new estimate.

Often, H, Φ and R are known, either from the physical properties of the system or from empirical sources. When this is the case, a tracker based on a single observation

model may describe adequately that aspects of system behaviour. However, Q depends on the object under observation, and is external to the system. Q is referred to in texts on aircraft tracking as the manoeuvre model following the analogy of a vehicle under human control. In control theory, it is more often referred to as the process noise model. In financial applications, Q represents changes in market trends and the ability to detect them at the earliest possible time is a valuable one.

In aircraft tracking, it may be that the objects of interest are homogeneous in their randomness and uniform in their random behaviour. If this is the case, nothing is lost if a single manoeuvre model is used for all updates to all object tracks. In the terminology of West & Harrison, a single DLM is sufficient to describe the system in question [83].

It is more likely that different objects have different random behaviours and that the randomness of individual objects depends on the context within which they are observed. These properties may be seen in the air traffic control example. Respectively: a light aircraft is more manoeuvrable than a wide-bodied passenger jet and an aircraft preparing to land is more likely to manoeuvre than it is when it is in transit. If a single manoeuvre model were used, the simplest approach that would guarantee a consistent estimator under all conditions would be to choose Q based on a white noise, random walk (Markov) model and assume a manoeuvre variance sufficiently large to accommodate the most violent manoeuvres which could occur. As a consequence of this model, under all other manoeuvre conditions, the tracker would not converge to its full potential accuracy.

Singer proposed a (single) manoeuvre model that was based on assumed object manoeuvre behaviour and recognised the auto correlated nature of the acceleration of real manoeuvres of a vehicle under human control [76]. Figure 3.10 gives the resulting model of manoeuvre acceleration distribution. The Singer model has the parameters:

Figure 3.10: The distribution of Aircraft Manoeuvre Variance [76]
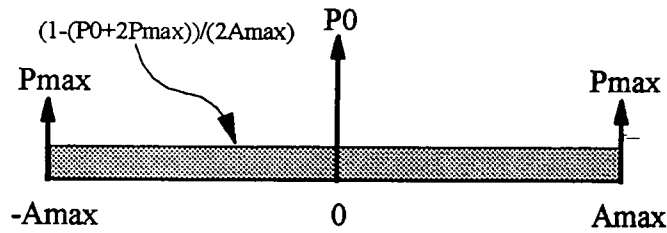
Singer's model for the distribution of aircraft manoeuvre acceleration variance.

• A_{max}, the maximum object acceleration which may be observed during a manoeuvre;

• P_{max}, the probability that A_{max} or $-A_{max}$ is observed;

• P_0, the probability that no manoeuvre is observed; and

• T, the duration of a typical manoeuvre.

The auto correlation function is then modelled as:

$$r(\tau) = E[v(t)v(t+\tau)] = \sigma_m^2 \exp\left(-\tfrac{1}{T}|\tau|\right)$$

where σ_m^2 is the manoeuvre acceleration variance modelled as:

$$\sigma_m^2 = \frac{A_{max}^2}{3}[1 + 4P_{max} - P_0]$$

Singer used these expressions to calculate a general expression for Q and gave limiting values as $\frac{T}{\tau} \to 0$ and as $\frac{\tau}{T} \to 0$.

Singer's approach recognises that different manoeuvre behaviours exist, but averages over them. Furthermore, the approach offers a single DLM to represent all objects and is unrealistic for many applications. The alternative is to use a multiple model approach.

If the components of the distribution of A from the Singer model could be separated and implemented as alternatives, the flexibility of the tracker would be improved and the potential accuracy of the system might be more fully exploited during periods of low manoeuvre. The model switching approach selects the manoeuvre model adaptively by monitoring the error state of the system. If the average error is small, Q is

chosen to reflect minimal manoeuvring. If the average error is large, Q is selected to allow for a violent manoeuvre. A common approach to this, in use in many real data fusion systems, performs a hypothesis test using a moving average normalised residual error:

$$\tilde{d}_k^2 = (1-a)\tilde{d}_{k-1}^2 + ad_k^2 \text{ and } \tilde{d}_k^2 < G_i \in \chi^2$$

where $d_k^2 = (y(k) - H\hat{x}(k,k-1))^T[HP(k,k-1)H^T + R]^{-1}(y(k) - H\hat{x}(k,k-1))$ is known as the Mahalanobis distance and $0 < a \leq 1$ [3]. The smallest threshold, G_i, for which the hypothesis test is satisfied determines the selection of Q and may cause a change to a smaller or larger manoeuvre model, or retention of the current one. More rigorous, theoretical approaches, employing models of manoeuvre noise distributions have been reported. For instance, Wu and Fitzgerald reported work on the detection of discontinuities in time series with Laplacian noise characteristics [86].

Another multiple model approach is a mixed model estimator. Magill did pioneering work in this field [50]. More recent developments have been reported by Blom & Bar-Shalom [10]. The approach of Blom & Bar-Shalom is known as the interacting multiple models (IMM) tracker. The IMM algorithm propagates N trackers based on N different models (hypotheses) and calculates the probability p_i of each hypothesis with $\Sigma p_i = 1$. The estimated state and covariance are then given by a linear combination of the N individual estimates with coefficients p_i such that $\hat{x} = \Sigma p_i \hat{x}_i$ and $P = \Sigma p_i P_i$. West & Harrison describe a system model, based on the supposition the at each time t the process may be represented by one of a finite number of discrete possibilities, as a multi-process class II model [83].

Caputi considered the self-learning properties of the IMM but noted that situations existed in which the IMM failed to converge to any particular hypothesis [13]. Caputi described necessary conditions to avoid such failures. Namely, the models must be sufficiently discriminating and different, and one model must approximate the

observed behaviour better than the others. Caputi provided an analytical formulation of these conditions.

It was noted in section 3.2 that Q is not the only variable tracking parameter. Conditions can arise in which the parameters of the observation process (H and $\overline{R}$) are variable. When the variation is between discrete observation behaviours, a model switching approach, applied to H and R, may be used to accommodate this. Alternatively, the IMM might be applied. Daeipour and Bar-Shalom applied the IMM approach to the problem of target glint [16]. Glint occurs when the object of interest is large in relation to the resolution cell of the sensor and successive observations may arise from different parts of the object. Thus, the observation process is subject to disruption. The IMM solution operated with two versions of R: one was based on the assumption of multivariate Gaussian distributed (MVGD) errors and represented returns from the dominant focus of observations; the other was based on assumed (longer tailed) Laplacian distributed errors and represented the distribution of errors in observations from the secondary foci.

Whilst the Kf is the de-facto standard approach to tracking, an alternative is gaining favour. Maybeck proposed a revised formulation to the solution of the DLM known as the information filter in which the attributes were normalised with respect to the covariance P [54]. B. S. Y. Rao, Durrant-Whyte and Sheen developed this idea to produce an algorithm known as the distributed Kalman filter (DKf) [66]. Define $z(k, k) = P^{-1}(k, k)x(k, k)$, then the information filter algorithm may be written:

$$P(k.k - 1) = \Phi P(k - 1, k - 1)\Phi^T + Q \qquad (3.2.1.3\ a)$$

$$z(k, k - 1) = P(k, k - 1)^{-1}\Phi P(k - 1, k - 1)z(k - 1, k - 1) \qquad (b)$$

$$z(k, k) = z(k, k - 1) + H^T R^{-1}y(k) \qquad (c)$$

$$P^{-1}(k,k) = P^{-1}(k,k-1) + H^T R^{-1} H \qquad \text{(d)}$$

The above algorithm is algebraically equivalent to the Kf algorithm of equations (3.2.1.2 a ... e) and so has the same properties and constraints. The great advantage of the information filter is that it may be conveniently divided up into segments implemented in different computers in different locations, a very useful property if the elements of a system (sensors and processors) are distributed over a wide geographical region. This property of convenient division arises because, when there are multiple sensors in the system, (3.2.1.3 c) and (d) become respectively:

$$z(k,k) = z(k,k-1) + \sum_i H_i^T R_i^{-1} y_i(k)$$

$$P^{-1}(k,k) = P^{-1}(k,k-1) + \sum_i H_i^T R_i^{-1} H_i$$

where the index i denotes sensor. In this latter form the algorithm is known as the DKf.

All of the above concerning the Kf and the DKf depends on the linearity of the system of interest. When the observation process or the object dynamic behaviour are non-linear, the DLM and Kf algorithm no longer apply. The non-linear observation and system models are of the form respectively:

$$y_k = h(t_k, x_k, u_k, \varepsilon_k) \text{ and } x_k = \phi(t_k, x_{k-1}, v_k).$$

where u denotes the control input (for example, when a sensor is mounted on a steered moving vehicle the steering is the control input). A common approach to tracker design is to take a linear approximation to the true system model at each new observation. The Extended Kalman Filter (EKf) takes this approach [2].

The EKf employs four of the Kf equations as defined in (3.2.1.2). They are equations (a), (b), (c) and (e) with $H = \frac{\partial}{\partial x} h(t_k, x_k, u_k, \varepsilon_k)$ and $\Phi = \frac{\partial}{\partial x} \phi(t_k, x_{k-1}, v_k)$. Equation (d) is replaced by:

$$x(k,k) = x(k,k-1) + K[y_k - h(t_k, x(k,k-1), u_k, \varepsilon_k)]$$

The EKf provides an adequate approximation to the track when the non-linearities are weak.

For highly non-linear systems, a different approach is required. Bar-Shalom and Fortmann examined the possibility for constructing an optimal, recursive tracking process to estimate an object state governed by a non-linear stochastic system [2]. Given white process and observation noise sequences that are independent, Bar-Shalom and Fortmann offer the recursive, discrete-time relationship linking the probability density function of the estimate at time k and $k+1$:

$$p(x_{k+1}|Y^{k+1}, U^k) = \tfrac{1}{c}p(y_{k+1}|x_{k+1}) \int p(x_{k+1}|x_k, u_k)p(x_k|Y^k, U^{k-1})dx_k \qquad (3.2.1.4)$$

where Y^k denotes the observations up to and including step k, U^k denotes the controls over the same interval, $\tfrac{1}{c}$ denotes a normalising constant and u_k denotes the control at step k. For linear systems, with Gaussian noise and initial state, the Kf is the optimal solution to (3.2.1.4). For linear systems, with non-Gaussian noise and/or initial state, in general, numerical solution is the only option.

Bar-Shalom and Fortmann describe (3.2.1.4) as "...in most cases analytically intractable and numerically expensive." Citing these as reasons for the degree of interest in approximate solutions. The requirement for white noise sources is cited as "...the reason why the formulation of stochastic estimation and control problems is done with white noise sequences."

Bar-Shalom and Fortmann wrote these words a decade ago. Since then, the speed and storage capacity of computers have increased dramatically and numerical solutions to (3.2.1.4) are becoming popular both for non-linear and non-Gaussian systems. Strictly speaking, when the noise is not white, a recursive tracking algorithm should not be used and a batch solution, based on equation (3.2.1.1) should be used.

3.2.2 Track Fusion Algorithms

Track Fusion is applied to a group of object track estimates, $\{\hat{x}\} = \{\hat{x}^1, \hat{x}^2, \ldots\}$ to create higher (multiple sensor) tracks. Its purpose is to estimate the state of the variable x, given $\{\hat{x}\}$. Track fusion is used when two or more object tracking processes, serving different sensors or sensor groups, operate simultaneously observing the same objects. When it is required to combine their outputs, a weighted combination of the object tracks would be produced. Consider the following simple example :

Two sensor groups produce estimates of the form $\hat{x}^i = x + \varepsilon_{x^i}$ and $\hat{x}^j = x + \varepsilon_{x^j}$ respectively. Assuming the errors have covariances P^i and P^j respectively, and are independent so that $E\left[\varepsilon_{x^i}\varepsilon_{x^j}^T\right] = P^{ij} = 0$, if a Minimum Mean Square estimator (MMSE) for x might be used [2] [41]:

$$\hat{x} = P^j[P^i + P^j]^{-1}\hat{x}^i + P^i[P^i + P^j]^{-1}\hat{x}^j \qquad (3.2.2.1)$$

$$= \hat{x}^i - P^i[P^i + P^j]^{-1}[\hat{x}^j - \hat{x}^i]$$

and this fused estimate has covariance P :

$$P = P^i[P^i + P^j]^{-1}P^j \qquad (3.2.2.2)$$

$$= \left(I - P^i[P^i + P^j]^{-1}\right)P^i$$

In each case the latter expression is the familiar form used in recursive computation. In the recursive form i represents the fused state derived from the sensors considered so far and j represents the next sensor under consideration. Note that this is not time recursion.

The problem with an algorithm based on equations (3.2.2.1) and (3.2.2.2) is the assumption of independence. In a dynamic stochastic system, both sensors are subject to the same process noise (manoeuvre) errors because both would be affected by the same

random variations in object state. So the errors in two tracks of a single object created from two, truly independent, observation sequences must nevertheless be correlated.

Bar-Shalom and Fortmann consider the case where $E\left[\varepsilon_{x^i}\varepsilon_{x^j}^T\right] = P^{ij} \neq 0$. They give the result for a minimum mean-square estimator (MMSE) $\hat{x}$ [2]:

$$\hat{x} = \hat{x}^i - [P^i - P^{ij}][P^i + P^j - P^{ij} - P^{ijT}]^{-1}[\hat{x}^j - \hat{x}^i]$$

(3.2.2.3)

and this fused estimate has covariance P :

$$P = P^i - [P^i - P^{ij}][P^i + P^j - P^{ij} - P^{ijT}]^{-1}[P^i - P^{ij}]^T$$

(3.2.2.4)

where P^{ij} obeys the **time** recursive relationship for a sensor pair $\{ij\}$ (producing synchronised sensor observations) at time n:

$$
\begin{aligned}
P_n^{ij} = \quad & \left(I - \left(\Phi P_{n-1}^i \Phi^T + Q\right)H^{iT}\left(H^i\left(\Phi P_{n-1}^i \Phi^T + Q\right)H^{iT} + R^i\right)^{-1}H^i\right) \\
\times \quad & \left(\Phi P_{n-1}^{ij} \Phi^T + Q\right) \\
\times \quad & \left(I - \left(\Phi P_{n-1}^j \Phi^T + Q\right)H^{jT}\left(H^j\left(\Phi P_{n-1}^j \Phi^T + Q\right)H^{jT} + R^j\right)^{-1}H^j\right)
\end{aligned}
$$

Any real implementation, with sufficient capacity for data exchange between the sensors to provide the data for this recursive calculation, would employ a global (possibly distributed) multiple sensor tracker rather than multiple, single sensor trackers and track fusion. The global tracker implementation would be simpler and more accurate. It follows that track fusion would only be used in situations where data exchange is limited. Blackman offers an example, based on two identical synchronised sensors, which approximates P^{ij} without the requirement for full data exchange but this is very much a special case [9]. Blackman himself writes that this approach cannot be readily extended to many, dissimilar, asynchronous, sensors. In general and in practice, P^i is unknown.

In many cases, the numerical effect of the error correlation will be very small, and the assumption of independence will produce acceptable results. Saha compared the track fusion performance of algorithms that assumed independent errors, based on equations (3.2.2.1) and (3.2.2.2), with the performance of algorithms which made due allowance for the correlation, based on equations (3.2.2.3) and (3.2.2.4) [72]. Saha used a steady-state approximation to P_n^{ij} based on the solution of a discrete asymmetric Lyapunov equation. It was reported that, as co-located sensors were made more dissimilar (disparate), the track-fused performance became increasingly similar to the single sensor performance of the better sensor. He concluded that track fusion based on an MMSE was not worthwhile under such circumstances.

Haimovich et. al. showed that, as co-located sensors were made more dissimilar, the track-fused performance became increasingly similar to an optimal multiple sensor tracker performance [30]. This result was obtained under the assumption of time-synchronised sensors. For sensors operating asynchronously, an optimal multiple sensor tracker would still be expected to show an advantage.

The sensors in the example of section 3.1 may be described as disparate; the first sensor measured horizontal polar co-ordinates (range and azimuth) and the second sensor measured spherical direction co-ordinates (azimuth and elevation). Figures 3.5 and 3.6 showed the observations and the fused information in attribute space (3-Dimensional position). Clearly, the only source of range information was the first sensor's estimate and elevation rotation information came from the second sensor's estimate. Rotation in the horizontal plane was provided by both sensors' azimuth observations. The more accurate estimate was provided by the second sensor and this estimate tended to dominate the fused estimate. This dominance occurred as follows. Let

$$\hat{\theta} \approx \frac{\hat{\theta}_j \times \sigma_{\theta_i}^2 + \hat{\theta}_i \times \sigma_{\theta_j}^2}{\sigma_{\theta_i}^2 + \sigma_{\theta_j}^2}$$ be the azimuth of the fused estimate and $\sigma_\theta^2 \approx \frac{\sigma_{\theta_i}^2 \times \sigma_{\theta_j}^2}{\sigma_{\theta_i}^2 + \sigma_{\theta_j}^2}$ be its

variance (the sub-scripted values represent the single sensor equivalents). When $\sigma_{\theta_i}^2 \ll \sigma_{\theta_j}^2$, $\hat{\theta} \approx \hat{\theta}_i$ and $\sigma_\theta^2 \approx \sigma_{\theta_i}^2$.

With these disparate, co-located sensors the transformed axis set where the errors are independent is approximately aligned with the spherical polar co-ordinates of the object. (The approximation is usually very good.) The overall effect is that each attribute in the transformed attribute space was approximated by the observed attribute from one of the sensors.

For disparate sensors the effect of error correlation is numerically small. Noonan showed that for disparate sensors, an heuristic approach that created a composite state estimate selecting the best sensor observation for each attribute, produced approximately the same results as MMSE fusion when applied to disparate sensors [60].

If the sensors are not disparate, $\varepsilon < \frac{\sigma_{x_i}^2}{\sigma_{x_j}^2} < \frac{1}{\varepsilon}$ for at least one attribute x of x where $0 < \varepsilon < 1$ is a given small value. Then more than one sensor would contribute to the fused estimates of the attributes satisfying this condition. The effects of error correlation would be numerically significant in such cases and the assumption of independence would not produce acceptable results.

At least two algorithms are available that produce results guaranteed free of error correlation problems. Namely, the heuristic composite state estimate evaluated by Noonan and the Covariance Intersection (CI) approach proposed by the Covariance Intersection Working Group (CIWG) [14] [60]. Both approaches produce a fused estimate with an error distribution that encloses the true distribution with covariance given by equation (3.2.2.4). Neither approach requires knowledge of P^{ij}. The latter

approach claims the advantage of producing the distribution which encloses the true error distribution with some norm minimised under the assumption of ignorance of P^{ij}. If the chosen norm were the covariance determinant of the fused estimate, entropy would also be minimised.

Stuart and Ord defined consistency of an estimator t_n of $\overline{\theta}$ based on n samples as $\Pr\{|t_n - \theta| < \varepsilon\} > 1 - \eta$, $n > N$ for any ε and η, however small [41]. This might be stated: by increasing the sample size the probability of improving on an arbitrarily small error in a consistent estimator can be made to approach arbitrarily close to unity. Stuart and Ord cite Fisher as providing an earlier definition which defined a consistent estimator as one which would provide the true value given the entire population [26].

The CIWG refers to the property of predicting an error distribution enclosing the true error distribution as consistency, such that $P - \bar{P} \geq 0$ where $\bar{P}$ denotes the true error covariance of the estimated state and P is the approximation delivered by the data fusion algorithm [14]. The latter property requires the expectation that the current estimate derived from the current sample achieves or betters a specified error squared, P. This property is not the same as the one defined by Stuart and Ord or Fisher.

Alternatively, the CIWG refers to an estimate with this property as conservative. We will use the latter term to avoid confusion with the traditional definition. The CIWG prove that the CI algorithm:

$$P^{-1} = \omega(P^i)^{-1} + (1 - \omega)(P^j)^{-1} \text{ and } P^{-1}\hat{x} = \omega(P^i)^{-1}\hat{x}_i + (1 - \omega)(P^j)^{-1}\hat{x}_j ,$$

with $0 \leq \omega \leq 1$, produces an estimate which is conservative provided the single sensor estimates were conservative.

The heuristic composite estimate algorithm evaluated by Noonan also produces an estimate which is conservative provided the single sensor estimates were conservative [60]. In the heuristic composite estimate algorithm, no fusion of data is performed and each attribute is represented by the best single sensor estimate.

3.2.3 Object Classification Algorithms

In addition to estimating parameter values and time derivatives, it is sometimes required to fuse observations which provide information about object class. Object class observations differ from track data in two important respects:

•the data are non-parametric, non ordered and discrete,

•data association is often subject to greater uncertainty due to difficulty mapping observations from object class sensors on to estimates of an object's track.

The latter feature of this type of application means that a single object class statement might be associated with many objects over a relatively wide region.

Given multiple statements concerning the class of a particular object, data fusion is required to produce a single, consolidated class statement. Documented approaches to object classification (sometimes referred to as identity fusion) are numerous. Moruzzis, Colin and Milhem produced a survey paper that compares and evaluates the approaches [56]:

•Fuzzy logic;

•Bayesian techniques; and

•Evidence theory.

For completeness, heuristic approaches are added at the head of the list and Bayesian techniques are extended to include all probabilistic approaches.

The use of heuristic approaches was reported by Llinas and Hall [48]. Heuristic approaches include voting and rule based reasoning. The voting approach is the simplest whereby all observations are mapped on to object classes and the class that occurs most often in the mapped observations is adopted by data fusion as the object class.

The rule based reasoning approach incorporates rules for interpreting observations in the logic of a computer program. A rule based approach can operate robustly by elimination. For instance a system for identifying aircraft might have a rule:

IF airspeed > 200 Kt THEN class ≠ helicopter.

Such a rule would cause a helicopter classes to be discarded when an unambiguously fast airspeed is observed for any object.

When sensors report probabilities or likelihoods associated with object class observations, heuristic approaches can under-exploit the available data. However, when sensors report only a single class with a non-probabilistic confidence statement or no confidence statement at all, and if no models of sensor observation error distributions are available to allow a weighting of object class observations to be applied, then heuristic approaches can perform as well as any other. Proponents of heuristic approaches claim that they allow the action of human operators to be synthesised in an automatic data fusion system.

Fuzzy logic was regarded as a promising approach by Moruzzis, Colin and Milhem [56]. Fuzzy logic allows membership functions $M(x)$ and possibility distributions $D(x)$ to be described. A membership function relates observed parameter value to object class in a manner which is less formal and more subjective than a probability distribution. Within certain predefined bounds an observed parameter value is said to indicate a particular object class. Either side of these bounds, the indication is said to be less certain and ramp functions are applied. The top-left graph of figure 3.11 shows this.

The possibility distribution sets bounds on the variation of the parameter value about the observed value. In the top-right graph of figure 3.11, the value x0 has been observed. Again this is interpreted as a spread of values with ramp functions on either side. Moruzzis et. al. define possibility P_x and necessity N_x respectively as:

$$P_x = \max_{\forall x} \left[\min(M(x), D(x)) \right] \text{, and}$$

$$N_x = \min_{\forall x} \left[\max(M(x), 1 - D(x)) \right]$$

These relationships are represented in the bottom-left and bottom-right graphs of figure 3.11 respectively.

Figure 3.11: Basic Relationships in Fuzzy Logic [56]
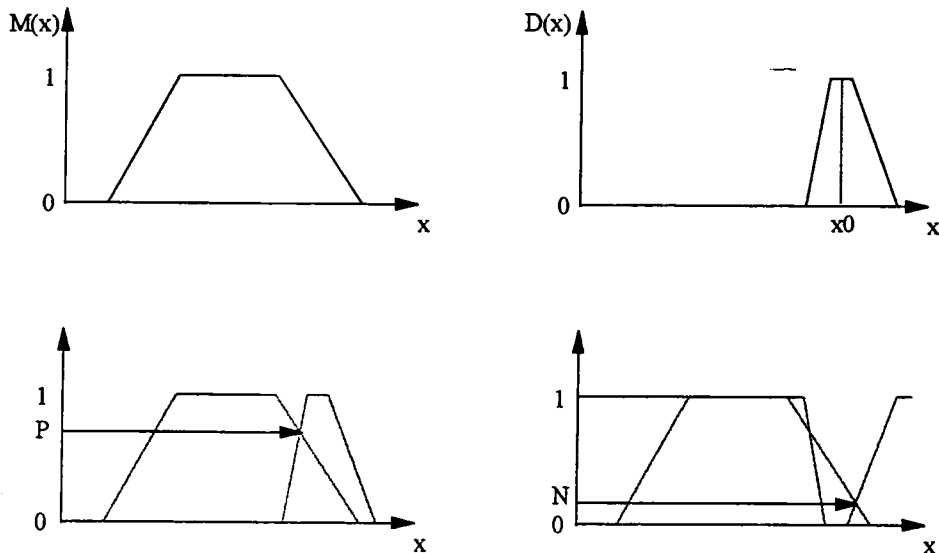
Fuzzy logic has been proposed as an approach to classifying objects.

For multiple observations the logical inference functions Weighted Conjunction and Weighted Disjunction are used to "fuse" data. Weighted Conjunction leads to:

$$P_x = \min_{\forall i} \left[\max\left(1 - W_i, P_{x,i}\right) \right] \text{ and } N_x = \min_{\forall i} \left[\max\left(1 - W_i, N_{x,i}\right) \right].$$

Weighted Disjunction leads to:

$$P_x = \max_{\forall i} \left[\min\left(W_i, P_{x,i}\right) \right] \text{ and } N_x = \max_{\forall i} \left[\min\left(W_i, N_{x,i}\right) \right].$$

Like probability, fuzzy logic leads to indications of object class which are numbers between 0 and 1.

The values W_i in Weighted Conjunction and Disjunction are the weights from which the functions derive their names. The weights define the relationships between observations and resulting class estimates and may be subjective interpretations of prior knowledge or be "learned" from empirical data.

Probabilistic approaches include conditional probability and Bayesian inference. Blackman provides a simple example of object classification using Bayesian inference

[8]. These approaches require probability (or likelihood) statements to be provided by the sensors or by the data fusion pre-processing.

Models of sensor observation error distributions could take the form of a confusion matrix $\{c_{ij}\}$ where each element characterises the likelihood of reporting identity i when the true identity is j. The columns of this matrix are similar to fuzzy set membership functions but are based on objective measurements or analysis. These approaches are useful when the sensors discriminate well between different identities and when targets belong to one of the known identities, j.

Conditional probability calculates the probability of object class conditional on the observed parameter values. No prior knowledge is assumed or allowed for in the calculations.

The recursive form of the Bayesian classifier given by both Blackman and Manyika is [8] [51]:

$$p(x|Y^k) = \frac{p(y_k|x) \times p(x|Y^{k-1})}{p(y_k|Y^{k-1})}$$

Where Y^k denotes the first k observations and y_k denotes the kth observation. This equivalent to calculating:

$$p(x|Y^k) = \frac{p(Y^k|x) \times p(x)}{p(Y^k)} \quad \text{since}$$

$$\frac{p(y_k|x) \times p(x|Y^{k-1})}{p(y_k|Y^{k-1})} = \frac{p(x) \prod_{n=1}^{k} p(y_n|x)}{\sum_{\forall x} p(x) \prod_{n=1}^{k} p(y_n|x)} = \frac{p(Y^k|x) \times p(x)}{p(Y^k)} .$$

In each case $p(y_k|x)$ could be provided from the matrix $\{c_{ij}\}$. In the above calculations $p(x)$ is the prior probability of class x before any observations are made. In the recursive form $p(x|Y^{k-1})$ takes the place of the prior probability. When the prior probability distribution is non-discriminating so that all classes are equally likely in the absence of observations, the conditional probability and Bayesian approaches are equivalent.

In situations where the data association is subject to significant ambiguity, Noonan and Pywell showed that the robustness of this type of algorithm was improved by taking due account of the data association ambiguity, so that [62]:

$$p(x|Y^k) = \frac{p(x|Y^{k-1}) \times \sum_{\forall y_k} \left(p(y_k|x) \times \theta_{y_k} \right)}{p(y_k|Y^{k-1})},$$

where θ_{y_k} is the probability that the observation y_k is the true update for the object of interest and $\sum_{\forall y_k} \theta_{y_k} = 1$. The Bayesian approach requires that the object classes form an orthogonal and exhaustive set. Thus, there must be no overlap between classes and all classes must be allowed for in the calculations.

Evidential approaches, such as Dempster-Shafer reasoning also depend on the availability of probability or likelihood vectors but do not need the classes to be orthogonal or the full vector to be available for every identity statement [20] [74]. These approaches may apply transformations that translate likelihood to "belief" or "probability mass" as opposed to probability [74]. Alternatively, they may calculate maximum and minimum probabilities for the object classes [20].

Evidential approaches cope better than probabilistic approaches when faced with identities that do not belong to j and are robust when there is no unambiguous best identity. Llinas and Hall give a simple example of an application of Dempster - Shafer evidential reasoning [48]. The evidential interval of x is defined as $[Sp(x), Pl(x)]$, Sp is support and Pl is plausibility.

The former is defined

$$Sp \triangleq \sum_{A \in x} m(A, 2^A)$$

where A is the set of all single propositions (Shafer's frame of discernment) and 2^A is the set of all subsets of A.

Plausibility is defined as $Pl(x) \triangleq 1 - Sp(\bar{x})$, where $\bar{x}$ denotes "not x". In plain language: all the evidence that does not refute class x. The uncertainty is defined as $U(x) = Pl(x) - Sp(x)$ and $Sp(x) + Sp(\bar{x}) \leq 1$. The observations of two sensors a, b are combined using Dempster's orthogonal sum:

$$m(A, 2^A) = m_a(A, 2^A) \oplus m_b(A, 2^A)$$, and for the class $\bar{x}$ this gives:

$$m(x) = \tfrac{1}{c}[m_a(x)m_b(x) + m_a(x)m_b(A) + m_a(A)m_b(x)] \text{ and}$$

$$m(A) = \tfrac{1}{c}m_a(A)m_b(A),$$

with $c = m_a(A)m_b(A) + \sum_{\forall x}[m_a(x)m_b(x) + m_a(x)m_b(A) + m_a(A)m_b(x)]$ and

$$m(A) = 1 - \sum_{\forall x} m(x).$$

Both the probabilistic and evidential approaches depend on the availability of models of sensor observation error distributions, either in the fusion pre-processing or in the sensor. If either of these approaches are adopted it is essential that the models are accurate. If they are inaccurate, the meaning of the results produced is not clear and that can be highly misleading.

If the sensor classification error distributions are known but are homogeneous, in the sense that the classification performance is roughly the same for every object class belonging to j, then all approaches are approximated well by simple voting.

3.2.4 Data Association Algorithms

Tracking, Track Fusion and Object Classification share two common assumptions: that it is known what objects exist and also which object gave rise to each observation. In general, neither is true.

Correlation and Allocation are terms that describe processes by which hypotheses, regarding the existence and number of objects in the attribute space, are created. The processes operate by identifying coherent time series of observations that

could, feasibly, be denoted as objects. These processes are sometimes referred to collectively as Data Association.

Two classes of Data Association are recognised in the literature. Those that make "hard" decisions and those that make "soft" decisions. A hard decision is one which irrevocable and, once made, is bound permanently into the estimates produced by the data fusion process. Given an object estimate and two candidate observations, of which one is a true update and the other is not, an algorithm making hard decisions would chose the most likely update. If the wrong one gets chosen, the estimation process may be adversely affected. Nearest neighbour approaches commonly work in this way.

Soft decisions are commonly implemented in one of two ways. One approach forms a weighted average of the candidate observations and updates the estimate using this average observation. So-called Probabilistic approaches work in this way.

The other approach to soft decisions stores multiple estimates for each object, each estimate based on a different sequence of decisions. At any time the most likely estimate is reported but as more information is received and the true decision sequence becomes clearer, the data association algorithm my switch to an estimate based on a more likely sequence. Track Splitting and Multiple-Hypothesis approaches work in this way.

In general, many hypotheses, Θ_i, exist which could explain the observations, $\mathbf{Y}$. We require some means of selecting the best hypothesis. In section 3.2 we identified maximum a posteriori probability (MAP) as a suitable criterion for such an optimisation. Let $\hat{X}_i$ be the estimate which results from the hypothesis Θ_i. Then from (3.2.0.1):

$$P(\hat{X}_i | Y, \Theta_i) = \frac{P(Y|\hat{X},\Theta_i)P(\hat{X})}{P(Y)} \qquad (3.2.4.1)$$

and Θ is chosen so that:

$$\Theta = \arg\max_{\forall i} P(\hat{X}_i|Y, \Theta_i)$$

Many approximate solutions to (3.2.4.1) are documented [2] [8] [51].

The well known approaches to data association were developed for tracking single objects and these will be considered. The ways in which the approaches are extended to multiple object tracking and Track to Track Fusion will be discussed.

The concentration on tracking applications has the consequence that most approaches assume the object state vector comprises only continuous random variables. We will consider variations which extend to non-parametric data and Object Classification applications will be discussed.

Whilst, all these approaches have all been designed for computer implementation as algorithms, some have been implemented non-algorithmically as neural networks and these will be discussed. We will start by considering a simple algorithmic approach to data association in single object tracking applications.

3.2.4.1 The Nearest Neighbour Standard Filter

The Nearest Neighbour Standard Filter (NNSF) updates a track with the nearest new observation received during the current update interval of a tracking process, provided at least one observation is feasibly near to the track [2] [8] [51]. If no nearby observation exists, the track is extrapolated over the interval and no other update is made.

The NNSF calculates the normalised distance from a single, estimated object, $\hat{x}$, to each new object observation, $y_i(k)$ at time k:

$$d_i^2 = (y_i(k) - H\hat{x}(k, k-1))^T U^{-1} (y_i(k) - H\hat{x}(k, k-1)) \qquad (3.2.4.2)$$

where $\hat{x}(k, k-1)$ is the estimate at k based on the observations $y_0 \ldots y_{k-1}$, H is the mapping of x into observation space and:

$$U = E[(y_i(k) - H\hat{x}(k, k-1))(y_i(k) - H\hat{x}(k, k-1))^T]$$

is the covariance of the innovation, $(y_i(k) - H\hat{x}(k, k-1))$. A validation gate is applied so that observations for which $d_{jk}^2 > a$ are rejected, where a is a point on the χ_m^2 distribution giving the desired (low) probability of rejecting the true observation and where m is the dimension of the observation vector. The observation is then chosen from those remaining (if any) so that $y(k) = \arg\min_i (\boldsymbol{d_i^2})$.

As noted above, the NNSF has the weakness that once an allocation decision is made, no allowance is made for the possibility that the allocation was wrong.

3.2.4.2 The Track Splitting Filter

Approaches that make allowance for the possibility of wrong allocation of observations to estimates have been reported. The Track Splitting Filter (TSF) is one such approach that calculates multiple versions of the track. Each version would be based on a different, feasible, time series of observations known as its track history. The Track Splitting Filter calculates the track, j, for each feasible track history up to time k [2] [8] [51]. The value d_{jk}^2 is calculated as in equation (3.2.4.2) for each update, k, in the sequence and the track history yielding the best track is reported, chosen so that

$\hat{x} = \arg\min_j \left(\sum_k d_{jk}^2 \right)$. Infeasible tracks are detected using the test $\sum_k d_{jk}^2 > a$ and deleted, where a is a point on the $\chi_{m \times k}^2$ distribution giving the desired (low) probability of deleting the true track, where m is the dimension of the observation vector. The process of deleting infeasible track histories is sometimes referred to as pruning. In addition to pruning infeasible histories, when two or more histories merge so that their last q updates are identical, the one with the lowest $\sum_k d_{jk}^2$ score is retained and the rest are pruned.

The NNSF and TSF select the best track on the basis of normalised distance square. Bar-Shalom and Fortmann point out that, under the assumption of Gaussian

distributed errors, the normalised distance square is the modified log-likelihood that the potential allocation $\{jk\}$ is the true one [2].

3.2.4.3 The Probabilistic Data Association Filter

The Probabilistic Data Association Filter uses a different approach that calculates Bayesian likelihoods of association hypotheses and uses them to update tracks probabilistically. The Probabilistic Data Association Filter (PDAF) in its sub-optimal form calculates the set of normalised distance squares, d_i^2 in the same way as the NNSF at time k [2] [8] [51]. It then calculates the probability, β_i, that each observation, y_i, is the true update:

$$\beta_i = cP_g^{-1}|2\pi U|^{-\frac{1}{2}} \exp\left(-\frac{d_i^2}{2}\right)$$

and the probability, β_0, that there is no update:

$$\beta_0 = c\left(1 - P_d P_g\right)P_{nfa}$$

where c is a normalising constant, P_d is the probability that the true object was detected, P_g is the probability that the true observation lay within the validation gate and P_{nfa} is the probability that no false alarm lay within the validation gate. The new estimate is obtained from a linear combination of all the possible updates:

$$\hat{x} = \sum_{i=0}^{n} \hat{x}_i \beta_i$$

where $\hat{x}_i$ is obtained using y_i to update the estimate at time k-1 and $\hat{x}_0$ is obtained by extrapolating the estimate without update.

In its optimal form, the PDAF process is performed for all feasible track histories. The probabilities, β_{ij}, are calculated that each observation, y_i, is the true update to the true track history, represented by the estimate $\hat{x}_j$:

$$\beta_{ij} = cP_g^{-1}|2\pi U|^{-\frac{1}{2}} \exp\left(-\frac{d_{ij}^2}{2}\right)P\left(\hat{x}_j|Y(k-1)\right)$$

where $Y(k-1)$ is the set of all observations up to and including time $k\text{-}1$. The probability of no update to track history j, β_{0j}, is calculated:

$$\beta_{0j} = c(1 - P_d P_g)P_{nfa}P(\hat{x}_j | Y(k-1))$$

and the new estimate is obtained:

$$\hat{x} = \sum_i \sum_j \hat{x}_{ij}\beta_{ij}$$

Infeasible and merged track histories are pruned as for the TSF.

All the above approaches assume a single object of interest and all require external initialisation of estimates. The simplest extension to multiple object scenarios would apply the above approaches to each object in turn. A common initialisation logic in a multiple-object scenario would attempt to form a new object estimate when observations lay outside the validation gates for all known objects on successive updates. Provided these observations pass validation with each other, a new object estimate would be formed. These approaches to multiple object scenarios share the weakness that no allowance is made for observations that pass validation for more than one object estimate. However, there are documented approximate solutions to (3.2.4.1) which take multiple object approaches into account.

3.2.4.4 Multiple-object NNSF (Optimal Allocation)

A multiple-object extension to the NNSF is offered by Blackman [8]. Blackman refers to the NNSF as the sequential NN approach and offers an extension to multiple-object scenarios that minimises the sum of distance measures for the selected set of allocations under the constraint that each observation may update only one estimate. We will refer to this as the simultaneous approach to data association. The normalised distance square measure is defined as:

$$d^2_{Gij} = d^2_{ij} + \log_e |U_i|$$

where the term in U_i allows for each existing object track to have a different covariance reflecting different degrees of convergence and track accuracy. Under the assumption of Gaussian distributed errors, $d^2_{Gij} = -2 \times \log$ likelihood $\{ij\} - m \log 2\pi$.

An assignment matrix of d^2_{Gij} is created and solved by the Munkres algorithm [11]. Thus, the hypothesis Θ is chosen so that:

$$\Theta = \arg \min_{\forall l} \sum_{ij \in \Theta_l} d^2_{Gij}$$

The Munkres algorithm serves only to reduce computation. When n objects are present Munkres offers the expectation of a solution after $O(n^3)$ computations. Full enumeration would require $O(n!)$ computations. Like the NNSF on which this extension is based, this approach offers no opportunity to repair false allocations. Once an allocation is made, it is irrevocable. Using this approach, it is possible that allocations that did not pass validation are included in the solution. Under such circumstances, the object estimates are not updated and the observations in question, along with others that were not allocated by the Munkres algorithm, become candidates for the formation of new objects.

3.2.4.5 The Joint Probabilistic Data Association Filter

The Joint Probabilistic Data Association Filter (JPDAF) is offered by Bar-Shalom as an extension to the PDAF [2]. The difference between the PDAF and the JPDAF lies in their treatment of multiple, candidate observations for an object update. The former assumes that the extra observations arose from noise whilst the latter calculates an assignment probability, β_{ij}, that each validated observation i arose from object j, conditional on the probability that the observation arose from all other objects for which it was validated.

The probabilities, β_{ij}, that observation, y_i, is the true update for object, $\hat{x}_j$, are calculated:

$$\beta_{ij} = c_j \frac{b_{ij}}{\sum_i b_{ij} + \sum_j b_{ij} - b_{ij}}$$

where c_j is a normalisation constant calculated so that $\sum_i \beta_{ij} = 1$. b_{ij} is defined:

$$b_{ij} \triangleq \begin{cases} (1 - P_d P_g) P_{nfa} & i = 0 \\ P_d |2\pi U|^{-\frac{1}{2}} \exp\left(-\frac{d_{ij}^2}{2}\right) & i \neq 0 \text{ and } \Omega_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}$$

and:

$$\Omega_{ij} \triangleq \begin{cases} 1 & \text{observation } i \text{ in validation gate of object } j. \\ 0 & \text{otherwise.} \end{cases}$$

The object state estimate is updated:

$$\hat{x}_j = \sum_i \hat{x}_{ij} \beta_{ij}$$

Any observation, i, which fails validation on all known objects, so that $\Omega_{ij} = 0$, $\forall j$, becomes a candidate for the formation of a new object.

Noonan and Pywell applied this approach to the inherently highly ambiguous data association that is common with some object classification sensors [62]. Whilst the joint probabilistic approach did not allow genuinely ambiguous situations to be resolved, it did prevent the gross errors that occur when data is fused on the basis of uncertain and unstable data associations.

3.2.4.6 Multiple Hypothesis Tracking (Reid's algorithm)

Multiple Hypothesis Tracking (MHT) or Reid's algorithm is an extension to the TSF [2] [8] [69]. The TSF estimated the object j for each feasible track history which could have given rise to the set of observations, $Y(k)$, and chose the estimate with the best score so that $\hat{x} = \arg \min_j \left(\sum_k d_{jk}^2\right)$. MHT creates and maintains hypotheses comprising track histories for each feasible set of objects which could have given rise to $Y(k)$ and

chooses the hypothesis that is most likely. The probability of each hypothesis is calculated recursively.

Let $P(\Theta_{l,k-1})$ be the probability of hypothesis l after the update at time k-1. If there are n observations at time k, let $\Theta_{r,k}^{l}$ denote an hypothesis at time k that describes one of the ways in which $\Theta_{l,k-1}$ could have evolved over the interval k-1 to k. If $\Theta_{r,k}^{l}$ comprises a set of $(n-p-q)$observation to object allocations, Λ, a further p observations that are false alarms and q observations that form new objects. Then:

$$P(\Theta_{r,k}^{l}) = cP(\Theta_{l,k-1})(P_{fa})^{p}(P_{new})^{q} \prod_{j \notin \Lambda}\left(1 - P_{d_j}\right) \prod_{i,j \in \Lambda}\left(P_{d_j}|2\pi U|^{-\frac{1}{2}} \exp\left(-\frac{d_{ij}^{2}}{2}\right)\right)$$

where P_{fa} is the probability of receiving a false alarm, P_{new} is the probability of observing a previously undetected object and P_{d_j} is the probability of detecting object j in the interval k-1 to k. c is a normalising constant, calculated such that:

$$\sum_{\forall l}\sum_{\forall r} P(\Theta_{r,k}^{l}) = 1$$

An hypothesis is chosen such that:

$$\Theta = \arg\max_{\forall l \forall r} P(\Theta_{r,k}^{l}).$$

Clearly, the potential exists for the number of hypotheses to grow very large, very quickly. For this reason, the hypotheses are pruned and only unique, feasible hypotheses are carried forward to the interval k to k+1. Non-unique hypotheses are hypotheses with the same number of objects and within which the objects are similar. Reid offers two alternative approaches to establishing similarity [69]:

•the objects share the last N (≥ 2) update observations in common.

•the objects have similar state estimates, so that $|x_{As} - x_{Bs}| \leq \delta\sqrt{\sigma_{As}^{2} + \sigma_{Bs}^{2}}$ and $\sigma_{As}^{2} < \gamma\sigma_{Bs}^{2}$, $\sigma_{Bs}^{2} < \gamma\sigma_{As}^{2}$ for each element s of the object state vectors. ($\delta = 0.1$ and $\gamma = 2$ are recommended constants for these tests).

Reid recommends that a linear combination of similar objects is used to create a joint hypothesis to replace both similar hypotheses that are then rejected. Blackman offers the simpler approach of rejecting only the less likely hypothesis [8].

Infeasible hypotheses are rejected if $P\left(\Theta_{r,k}^{l}\right) < P_0$, where P_0 is some low, threshold probability. Blackman points out than in many practical applications, this approach can still lead to excessive numbers of hypotheses and suggests that only the M, most likely hypotheses should be retained when more than M feasible hypotheses exist. In highly ambiguous situations, with many more than M likely hypotheses, this can degrade the performance of the system as the probability of failing to consider the true hypothesis increases.

On completion of pruning, the remaining hypotheses are re-labelled $\Theta_{l,k}$ ready for the next cycle of observations to be considered by the MHT process.

3.2.4.7 Non-Algorithmic Approaches to Data Association

So far, we have considered only the common algorithmic approaches to Observation-to-Track association for tracking purposes. Non-algorithmic approaches, namely Neural Networks, have been proposed and demonstrated as possible alternatives. An analogue computational network was used by Sengupta and Iltis to simulate the performance of a JPDAF algorithm [73]. Sengupta and Iltis created an energy function to which they applied a Hopfield-Tank optimisation network and reported simulation results that suggest that such a network has the potential to approximate the performance of the JPDAF.

Barker, Brown and Martin proposed a sub-optimal correlation algorithm designed specifically for neural network implementation [3]. The correlation algorithm was an approximation to the NNSF and the Neural Network performed operations equivalent to a Hopfield-Tank analogue network. In their algorithm, the best remaining

association $\{ij\}$ is removed from the system recursively until there are no feasible ones remaining. Barker et. al. used a distance measure called divergence, that lends itself readily to computation in a neural network, defined as:

$$\tfrac{1}{2}d_i^2 + \tfrac{1}{2}\text{tr}\big((HPH)^{-1}R + R^{-1}HPH - 2I\big)$$

where d_i^2 is as defined in equation (3.2.4.2) and H, P, R and I are as in equations (3.2.1.2). Promising simulation results were reported.

Early promise and progress with these technologies have not been exploited. Various explanations for this may be given:

- the progress came at a time when digital computer technologies were advancing in capability and reducing in price at an unprecedented rate;

- advances in theory were not matched by corresponding advances in implementation technology; and

- in general, the processes associated with higher-level Data and Information Fusion may be characterised mathematically and, with modern computing equipment, optimised on-line.

The great potential of Neural Network processing appears to lie in lower-level processes such as pattern matching and signal processing.

3.2.4.8 Track-to-Track Data Association

Next we will look at associating information from independent trackers. The trackers are independent in the sense that each tracker consolidates observations from separate independent sensors. The data associated in this way will be the subject of a subsequent Track Fusion process and the association process is referred to as Track-to-Track association. Track-to-Track association differs from Observation-to-Track association in several ways:

- The data tend to be of higher dimensionality because of the extra attributes (usually time derivatives) estimated by the Tracking processes. This can reduce

association ambiguity in many situations but requires more processing to calculate the distance and likelihood metrics used to discriminate between association hypotheses.

•An effective tracker process acts as a filter on false alarms such that the probability of receiving a false track is small enough to ignore in most circumstances.

•Whilst the trackers may be truly independent, when the attributes of a system are random, the errors in the estimates they produce are correlated as discussed earlier in this chapter.

There are fewer documented approaches to Track-to-Track association. As in the discussion of Observation-to-Track association, the following applies to continuous attributes only. If we denote the set of sensor tracks as X, the track-to-track association hypothesis as Ψ and the Track Fused estimate as $\hat{Z}$, we can rewrite (3.2.1.1):

$$P(\hat{Z}_i|X, \Psi_i) = \frac{P(X|\hat{Z}_i, \Psi_i)P(\hat{Z}_i)}{P(X)} \qquad (3.2.4.3)$$

and we choose Ψ so that:

$$\Psi = \arg\max_{\forall i} P(\hat{Z}_i|X, \Psi_i)$$

Blackman recommends a simple "gating" process that looks for a known fused track that passes a validation test ($d_{ij}^2 < g \in \chi_m^2$ where g is the validation gate threshold) with each sensor track [8]. This is often adequate for associating data from co-located, similar sensors and for maintaining established association hypotheses in other circumstances. In both these cases it is unlikely that more than one feasible (validated) association possibility exists. When multiple feasible association possibilities exist simultaneously, this approach offers no way to discriminate between them.

Various authors have developed extensions to NNSF that allow multiple tracks from multiple sensors to be associated [36] [58]. For each candidate pair of tracks, ij, a distance-square measure, D_{ij}^2, is calculated based on the negative log-likelihood of the association, $\{ij\}$.

$$D_{ij}^2 \approx -2 \times \ln[P\{ij|X\}]$$

The D_{ij}^2's form an allocation matrix that is solved using an optimal search algorithm (Munkres) to give the MAP set of associations [11]. The optimal search yields a set of associations $\hat{\Psi}$ that approximates Ψ and recalling the criterion for optimising the solution to (3.2.4.3):

$$\sum_{ij \in \hat{\Psi}} D_{ij}^2 \propto -2 \times \ln[P(\hat{Z}|X, \hat{\Psi})]$$

noting that:

$$\arg \min_{ij \in \hat{\Psi}} \sum D_{ij}^2 \cong \arg \max_{\forall i} P(\hat{Z}_i|X, \Psi_i)$$

If D_{ij}^2 is calculated using the true log likelihood of association, true equivalence is achieved. These approaches are capable of establishing and maintaining association hypotheses in more complex and challenging situations.

Noonan used the recursive Bayesian probability to derive the expression for inter track distance [58]:

$$d_{Gij}^2(k) = d_{ij}^2(k) + \log_e |U_i| + m \ln 2\pi - 2 \ln \left\{ \text{Pr}_{Gij}(k-1) \right\}$$

where $\text{Pr}_{Gij}(k-1) = \exp\left(-\frac{1}{2} d_{Gij}^2(k-1)\right)$. Houston and Kraus have used the Bhattycharyya distance [36]:

$$D_{ij}^2 = \int \sqrt{(\hat{x}^i - \hat{x}^j)^T (\hat{x}^i - \hat{x}^j)}$$

A non-algorithmic approach to the optimisation of Track-to-Track association has been demonstrated by Jackson [37]. A neural network of the type proposed by

Barker, Brown and Martin was shown to extend to the track-to-track association problem and to give a good approximation to the optimal set of associations, Ψ [3].

3.2.4.9 Object Classification Data Association

Finally, the ways in which non-parametric, discrete attributes (object classifiers) may be used to improve the Track-to-Track association process, will be considered. Here, the Bhattycharyya distance proposed by Houston and Kraus extends readily to measure object class similarity [36]. As an alternative, Moruzzis, Colin and Milhem proposed that this should be measured in terms of the probability of correct association [56]:

$$\mathrm{Pr}_c = \tfrac{1}{c} \sum_x \mathrm{Pr}(x|\hat{x}^i)\,\mathrm{Pr}(\hat{x}^j|x)$$

where c is a normalising constant. Conditional probability, under the assumption of independence, yields:

$$D_{I,ij}^2 = -2\ln\{\breve{x}^i \cdot \breve{x}^j\} ,$$

where $\breve{x}^i$ and $\breve{x}^j$ are the object class probability vectors for the tracks.

4   Information and Entropy

A new analytic approach to the calculation of combined entropy of continuous numerical information and discrete object-classification information is defined in this chapter (section 4.5).

Entropy was first proposed as a means to quantify the relationships between energy, heat and disorder. However, a thought experiment known as Maxwell's demon highlighted the potential role of perception and intelligence in the reduction of entropy. If the energy of individual particles could somehow be perceived and the particles sorted into energetic and non-energetic groups by means of a controllable shutter or gate between two chambers, the entropy of the system would be reduced without work being applied directly to the system. Were this possible, it would contradict the second law of thermodynamics. It is not possible, but it is analogous to the ways in which perception and intelligence are used in data fusion to sort information into related groups and so reduce the entropy of our state of situation awareness.

More recently, entropy has been used as a measure of the information "profit" or "loss" associated with a process or an action. In the Theory of Communication and subsequently in Radar Signal Processing, entropy has been used to quantify the effect on information of the Communications transmission and Radar update processes respectively. Information entropy has been used quantify the expected information gain in automatic Sensor Management. This chapter examines these issues in greater detail and considers the effect, on information entropy, of a data fusion process.

Finally in this chapter, the error and uncertainty in the information delivered by data fusion is discussed. The information may have both continuous and discrete components and this mixing of types raises issues concerning the entropy datum and scale. A reconciliation of scales between discrete and continuous information entropy is

necessary if entropy is to provide a single valued measure of data fusion information error. A new approach is proposed.

4.1   A historical perspective

Entropy was proposed in thermodynamics as a means of quantifying the amount of disorder in an ensemble of particles. When an ensemble of particles is heated its molecules move more energetically and entropy is a means of describing the properties of the order or disorder in these motions.

The second law of thermodynamics implies that: "... in a spontaneous irreversible process, the total entropy of a system and its surroundings always increases; for any process the total entropy of a system and its surroundings never decreases. ..." [21].

James Clerk Maxwell's book *Theory of Heat* of 1871 referred to a thought experiment, now known as Maxwell's Demon, that involved entropy and the second law of thermodynamics [53]. Maxwell wrote: "... One of the best established facts in thermodynamics is that it is impossible in a system enclosed in an envelope which permits neither change of volume nor passage of heat, and in which both the temperature and pressure are everywhere the same, to produce any inequality of temperature or pressure without the expenditure of work. This is the second law of thermodynamics, and it is undoubtedly true as long as we can deal with bodies only in mass, and have no power of perceiving or handling the separate molecules of which they are made up. But if we conceive a being whose faculties are so sharpened that he can follow every molecule in its course, such a being, whose attributes are still as essentially finite as our own, would be able to do what is at present impossible to us. For we have seen that the molecules of a vessel full of air at uniform temperature are moving with velocities by no means uniform, though the mean velocity of any great number of them, arbitrarily selected, is almost exactly uniform. Now let us suppose that such a vessel is divided into two portions, A and B, by a division in which there is a small hole, and that a being, who can

see the individual molecules, opens and closes this hole, so as to let only the swifter molecules to pass from A to B, and only the slower ones to pass from B to A. He will thus, without expenditure of work, raise the temperature of B and lower that of A, in contradiction to the second law of thermodynamics. ..."

A modern demon is depicted in Figure 4.1. We might imagine the demon as an ordinary human with sharpened faculties derived from access to microscopic sensors and the power to intervene in the molecular arrangement of the air in the partitioned vessel provided by remote control of an equally microscopic gate mechanism. The point of the thought experiment is the same. Firstly, the statistical nature of the second law of thermodynamics is demonstrated. There is a finite (if tiny) probability that the exchanges of molecules brought about by the demon would occur by chance if the gate were permanently open.

Figure 4.1: Maxwell's Demon
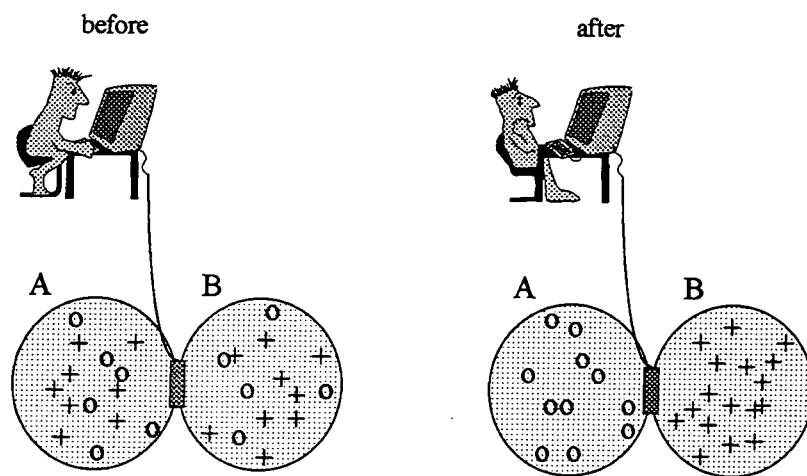
Through selective operation of the gate between two chambers, Maxwell hypothesised that the demon could sort the particles into high and low energy.

Secondly, and of more interest to this work, the role of perception and intelligence in the outcome of the experiment. Ludwig Boltzmann is quoted as having described entropy as a measure of "missing information". By informed, selective sorting

and re-mixing of the molecules, Maxwell's demon brought about a reduction of entropy. An analogy exists with the processes which take place in a data fusion system. By selective combination of data we will reduce the entropy in the information we hold.

Maxwell's demon made only a limited contribution to the field of Thermodynamics. However, it is included here because it highlights issues which are important in the understanding of the ways information quality is improved in the data fusion process. In section 4.3 these issues will be considered in more detail.

The ideas introduced into Thermodynamics developed into Statistical Mechanics. Rushbrooke cites J. Willard Gibbs as a pioneer of the application of Statistical Mechanics in Physical Chemistry (also known as Chemical Thermodynamics) [71]. The Physical Chemist wishes to: " ... interpret and, as far as possible, predict the properties of macroscopic physical and chemical systems in terms of the microscopic systems (atoms, molecules, ions, electrons, etc.) of which these aggregates are made up... " [71]. In particular Physical Chemists are interested the equilibrium properties of these systems.

Rushbrooke offers qualitative examples the ways in which entropy relates randomness and disorder with ignorance and uncertainty. When a liquid evaporates to a gas, its entropy per mole increases and what limited knowledge was held about the positions of the systems (the molecules) of the assembly (the liquid) is destroyed. Alternatively, when a partition separating two gases is removed and the are molecules allowed to diffuse there is again an increase in entropy. Ignorance and uncertainty have increased and not only regarding the locations of the molecules. Previously, knowledge of the location of a molecule would tell us about its class or species, now it does not.

The basic equation of Statistical Mechanics is:

$$S = k \log \Omega$$

S is the entropy, k is a constant and Ω is the number of equally probable complexions (micro molecular states) of the assembly. More precisely, Ω is written $\Omega(E, V, N)$ where E is the energy, V is the volume and N is the number of systems contained.

The application of a measure similar to entropy in information science can be traced back to Hartley [33]. Hartley proposed that the information capacity of an electrical storage system should be defined:

$$C \triangleq \log\{n\}$$

where n is the number of distinguishable states in the system. This definition conveniently makes the capacity of a compound system equal to the capacity of a basic storage unit times the number of units. When the logarithm is to the base 2, C is equivalent to the number of binary storage units and is measured in bits of information.

Shannon extended and developed the application of entropy to the mathematical theory of communications [75]. Shannon defined the average information content of a message as:

$$H = \Sigma_k P(x_k) \log\left[\frac{1}{P(x_k)}\right],$$

for a discrete system, where $P(x_k)$ is the probability of the kth value of the discrete random variable x_k. The states are no longer equally probable except when H is a maximum. For a continuous system Shannon proposed:

$$H(x) = \int_{-\infty}^{\infty} p(x) \log\left[\frac{1}{p(x)}\right] dx,$$

where $p(x)$ is the probability density of the continuous random variable x. He used measures based on these equations to discuss communications channel capacities and rates. These equations give the well-known formulae for the discrete and continuous forms of information entropy. In section 4.2 we will discuss standard results and relationships associated with information entropy, many of which originate from Shannon's early work.

Woodward applied information entropy to Radar system design [85]. Woodward tried to answer the question "what to transmit?" in order to maximise the information provided by his Radar. Whilst he failed to give a definitive answer, his mathematical analysis of Radar information considered, among other things, information gain. Expressed simply in terms of the probability densities, $p_0(x)$ and $\overline{p_1(x)}$ respectively, of some state of interest, x, before and after an action. The example given by Woodward was the estimate of range (distance) to an object before and after a Radar observation. The information gain as a result of the Radar observation was defined as:

$$J = \log\left\{\frac{p_1(x)}{p_0(x)}\right\}.$$

Manyika and Durrant-Whyte proposed "... a framework for addressing multi-sensor data fusion and sensor management in general, and in decentralized systems in particular. ..." based on an information theoretic approach [51]. Manyika and Durrant-Whyte too considered information gain, but in their case it was the expectation of information gain if a particular sensing action were carried out. Based on a recursive formulation of Bayes Theorem:

$$H(p(x|y)) = H(x) - I\left(\frac{p(y|x)}{p(y)}\right)$$

where $I = -H$ and $I\left(\frac{p(y|x)}{p(y)}\right)$ was the information value of the update which was referred to as the Mutual Information. Mutual Information was described as the information about one variable, contained in another.

4.2 Information Entropy : Standard Results

We have seen that Shannon (most notably) applied the idea of entropy to the mathematical theory of communications [75]. He represented a discrete information source as an ergodic Markov process and sought to measure how much information was produced by the process and the rate at which that information was produced . Shannon started by examining stochastic processes as representations of the patterns that occur in

language. He gave examples of simulations of English text, first with a random mixture of alphabetic symbols, then demonstrating increasing similarity to English as models of single, double and triple letter pattern probabilities were introduced and finally even greater similarity as single and double word pattern probabilities were used. He pointed out that multiple alphabetic symbol and multiple word approximations could be represented as Markov processes.

Starting with a set of discrete events, x_k, and associated probabilities, $P(x_k)$, he looked for: "... a measure of how much 'choice' is involved in the selection of an event or of how uncertain we are of the outcome. ..." [75]. He called the measure H and required it to possess the properties:

•H should be continuous in the $P(x_k)$;

•If all the $P(x_k)$ were equal, $P(x_k) = \frac{1}{n}$, then H should be a monotonic increasing function of n;

•If a choice be broken down into two successive possibilities, the original H should be a weighted sum of the value of H for the individual steps. For the example in Figure 4.2 we require that:

$$H(p_1,p_2,p_3) = H(p_1, 1 - p_1) + (1 - p_1)H\left(\frac{p_2}{1-p_1}, \frac{p_3}{1-p_1}\right).$$

Figure 4.2: Decomposition of choice for three possibilities [75]
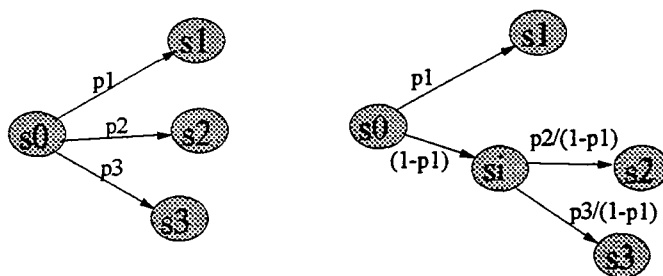
Breaking a choice down into its constituent steps does not change its entropy.

Starting in state s_0 the probabilities of arriving in states s_1, s_2 or s_3 are the same, irrespective of the presence or otherwise of the intermediate state s_i so the entropy of the total state transition is the same in both cases.

Shannon showed that the only H satisfying all three requirements is of the form:

$$H = K \times \Sigma_k P(x_k) \log\left[\frac{1}{P(x_k)}\right]$$

where K is a positive constant which serves to define the unit of measure. He recognised this as the form of entropy used in some formulations of statistical mechanics and defined information entropy as:

$$H = \Sigma_k P(x_k) \log\left[\frac{1}{P(x_k)}\right] \tag{4.2.0.1}$$

This definition, gives entropy as $-E[\log$ likelihood$]$, where E denotes expectation. It comes from being able to enumerate all possible permutations of values of the discrete variable x. This is also true for qualitative attributes such as identity or subjective colour but not for continuous variables.

Shannon listed the following properties of the entropy of a discrete variable:

•$H = 0$ if and only if all but one $P(x_k)$ are zero, this one having the value unity;

•For a given n, H is a maximum and equal to $\log\{n\}$ when the $P(x_k)$ are equal, i.e. $\frac{1}{n}$;

•For joint events x_k and y_l with probability $P(x_k, y_l)$, the following are defined:

$$H(x,y) = \Sigma_{k,l} P(x_k, y_l) \log\left[\frac{1}{P(x_k, y_l)}\right]$$

$$H(x) = \Sigma_{k,l} P(x_k, y_l) \log\left[\frac{1}{\Sigma_l P(x_k, y_l)}\right]$$

$$H(y) = \Sigma_{k,l} P(x_k, y_l) \log\left[\frac{1}{\Sigma_k P(x_k, y_l)}\right]$$

then $H(x,y) \leq H(x) + H(y)$, with equality for independent events;

•Any change towards equalisation of the probabilities, $P(x_k)$, increases H;

•If the conditional entropy $H(y|x)$ is defined as:

$$H(y|x) = \Sigma_{k,l} P(x_k, y_l) \log\left[\frac{1}{P(y_l|x_k)}\right]$$

then

$$H(x,y) = H(x) + H(y|x),$$

$$H(x,y) = H(x) + H(y|x) \leq H(x) + H(y) \text{ and}$$

$$H(y) \geq H(y|x).$$

Thus, uncertainty regarding y is never increased by knowledge of x and will be decreased except when x and y are independent when it is unchanged. Shannon used these results to develop his measures of communication channel capacity and rate.

The entropy of discrete attributes is related to the probability of a particular permutation and to the number of possible permutations. For the continuous case, it is related instead to the probability density of a given observation and to the likely volume occupied in attribute space. Shannon defined the entropy of a continuous variable as $-E[\log$ likelihood] which gives:

$$H(x) = \int_{-\infty}^{\infty} p(x) \log\left[\frac{1}{p(x)}\right] dx \qquad\qquad (4.2.0.2)$$

Shannon listed the following additional properties of the entropy of a continuous variable:

•If x is limited to a certain volume, V, entropy is a maximum and equal to $\log\{V\}$ when x is uniformly distributed over the volume with probability $\frac{1}{V}$. This fact is used in Chapter 6 as the basis for a model which quantifies the Entropies of attributes and objects of which we are ignorant. Provided the possible range of values (V) of the missing information is known, we can calculate the maximum entropy of not knowing.

•If the variance, σ^2, of x is fixed, entropy is maximum for a Gaussian distribution and is given by:

$$H = \log(\sqrt{2\pi e}\, \sigma) \text{, for scalar } x \text{ and}$$

$$H = \log\left(\sqrt{(2\pi e)^m |P|}\right) \qquad\qquad (4.2.0.3)$$

for vector x of covariance P and dimension m. Equation (4.2.0.3) is the entropy of the multivariate Gaussian distribution.

•If x is limited to a half line such that $p(x) = 0$ for $x \le 0$, entropy is a maximum and equal to $\log\{e\mu\}$ when $p(x) = \frac{1}{\mu}e^{-\frac{x}{\mu}}$.

•If a transformation changes the variable from x to y :

$$H(y) = H(x) - E\left[\log J\left(\tfrac{x}{y}\right)\right] = H(x) + E\left[\log J\left(\tfrac{y}{x}\right)\right] \qquad (4.2.0.4)$$

in particular for a general linear transformation

$$y_j = \sum_i a_{ij}x_i \text{ , } H(y) = H(x) + \log|a_{ij}|.$$

•For rotation (or any measure preserving transformation), $J = 1$ and $H(y) = H(x)$. Shannon pointed out the dependence of the entropy of a continuous distribution on the co-ordinate system and described it as a measure of randomness relative to an assumed standard, namely the co-ordinate system chosen with each small volume element dx having equal weight. After the change of variable entropy measures randomness with respect to a different small volume element dy. He noted that these differences do not affect the difference of two entropies.

Lahti took a different approach to the definition of the entropy of a continuous variable, regarding it as the limit of a discrete random variable as the step size approached zero [46]. So, the probability $P(x_k)$ became a probability density $p(x_k)$ and the expression for the entropy became:

$$H(x) = \lim_{\Delta x_k \to 0} \Sigma_k P(x_k) \Delta x_k \log\left[\frac{1}{P(x_k)\Delta x_k}\right]$$

This reduced to:

$$H(x) = \int_{-\infty}^{\infty} p(x) \log\left[\frac{1}{p(x)}\right] dx - \lim_{\Delta x \to 0} \log \Delta x \qquad (4.2.0.5)$$

This well-known result for a single continuous variable illustrates the problem when combining entropy from discrete and continuous quantities. The first term is the same as Shannon's definition and is directly analogous to the expression for a discrete variable. The second term is infinite. Various ways may be found of avoiding the infinity, they include:

- adoption of the approach used in thermodynamics and taken by Shannon and ignoring the infinity, concentrating on changes in entropy only [75]; or

- setting an arbitrary size to Δ_x and normalising with respect to this unit [5].

In the first case, the effect of adding the first term for entropy from a continuous measure to that for a discrete is not invalidated by the exclusion of the second term. However, the normalisation of entropy for a continuous measure requires an accounting being made for the scale. Alteration of scale changes the value of the datum for continuous entropy (second term). If the scale of Δ_x is altered by a factor e, then both the value of $\int_{-\infty}^{\infty} p(x) \log\left[\frac{1}{p(x)}\right] dx$ and that of $\lim_{\Delta x \to 0} \log \Delta_x$ are altered by 1. In general, for a transformation with Jacobian J, both values change by $E(\log J)$.

4.3 Information Entropy and Data Fusion

We will consider the application of entropy to measure the quality of information produced by data fusion processes and whether or not such an approach will allow the effectiveness of a data fusion process to be measured.

Data fusion takes observations regarding the state of the objects which inhabit an attribute space and turns them into one consolidated statement of the state of that

attribute space. We want to know how the information quality of the fused information relates to that of the source observations and we propose to quantify this relationship using entropy as a measure.

The Bayesian model of data fusion reflects common sense regarding the relationships that exist between existing information and new observations. If we treat existing fused information as prior information x and the new observation as the sample y, the probabilities obey the relationship:

$$p(x|y)p(y) = p(y|x)p(x) \text{ or more usually } p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

For the purposes of their work on Sensor Management, Manyika and Durrant-Whyte gave the information form of this expression as [51]:

$$H(p(x|y)) = H(x) - I\left(\frac{p(y|x)}{p(y)}\right)$$

where $I = -H$ and $I\left(\frac{p(y|x)}{p(y)}\right)$ was the information value of the update which was named the Mutual Information. Manyika and Durrant-Whyte had the objective of predicting the utility of a potential update to data fusion arising from one of many potential sensor actions. The largest values of I indicated the sensor actions likely to yield the greatest information returns.

We are concerned with the expected outcome of the data fusion process overall. Taking logarithms and expectations and negating we will concentrate on the value of $H(p(x|y))$ given by:

$$H(p(x|y)) = H(p(x)) + H(p(y|x)) - H(p(y)) \tag{4.3.0.1}$$

Note that the $H(p(x|y))$ is the entropy of the distribution of $p(x|y)$ and $H(p(y|x))$ is the entropy of the distribution of $p(y|x)$ which are not the same as Shannon's definitions of conditional entropy $H(x|y)$ and $H(y|x)$ [75].

For synergy we need $H(p(y)) \geq H(p(y|x))$ where equality means that x and y are orthogonal and that the observation y tells us nothing new regarding the state of attribute space x. Common sense tells us that two uncertain statements should never be more uncertain than either alone and when we deal with expectation, this is always the case.

For an individual data fusion update, it is possible that fused information is more uncertain than the better of the sources. This could arise because previous data had been misinterpreted in some way or because contradictory new observations had been introduced. In Chapter 3, Multiple Hypothesis approaches took advantage of this. The MHT sought the track history which maximised the likelihood of correct interpretation of the data in preference to the best single update.

In Chapter 6 we consider the possibility of using equation (4.3.0.1) as the basis for a model of the Object Classification process and a similar expression is seen to represent the entropy gain of an MMSE Track Fusion process.

The role of perception and intelligence in the outcome of the thought experiment known as Maxwell's demon has been noted. By selective sorting and re-mixing of the molecules in a partitioned chamber containing a gas Maxwell's demon was able, hypothetically, to bring about a reduction of entropy. It is clear that an analogy exists here with the processes which take place in a data fusion system. But there are both similarities and dissimilarities when the two situations are compared. The similarities are summarised:

- Maxwell's demon brings about a reduction in entropy by selective sorting of the molecules and data fusion does the same by selective sorting of data into groups of observations belonging to individual objects.
- The selection is possible, hypothetically, for Maxwell's demon through perception of the energy of the molecules and for data fusion is actually achieved through perception of the information content of the source observations.

•An intelligent strategy is applied to the sorting process in both cases.

The dissimilarities are summarised:

•Maxwell's demon is trying to influence a natural process to behave in an unnatural way; something which can never really happen. data fusion can in fact sort observations into clusters or groups with a high probability of having arisen from the same object. In the sorted form, the observations convey more information about the objects in question, even before any fusion of data takes place.

•Maxwell's demon is concerned only with sorting. In data fusion the sorted groups are consolidated and the information is improved still further.

It might be argued that a managed array of sensors which allocates tasks to optimise its data fusion performance is even more like Maxwell's demon in its operation. Within such a system, observations are gathered only when an overseeing manager (human or automated) decides that they are needed to improve the information quality of the fused picture.

4.4 The Units and Scale of Information Entropy

So far the only reference to the units of the entropy measure was in Section 4.1. It was noted that if a logarithm base of 2 was used, the capacity C of the information storage system was equal to the number of binary storage units in the system [33]. The unit was bits of information. Such a unit is in common use when discrete systems are considered.

However, Shannon's results for the Entropies of continuous statistical density functions imply that the logarithm base for that section of the work was e and the unit was nats of information [75]. For the rest of this thesis, the logarithm base e will be used exclusively and the unit of information will be the nat.

For continuous information there still remain unresolved issues regarding the datum for the information scale. Measured on the same scale as discrete information, the value of H for continuous information is infinite. When a relative entropy is given, the value quoted is:

$$H(x) - \lim_{\Delta x \to 0} \overline{\log \Delta x}$$

and the datum, $\lim_{\Delta x \to 0} \log \Delta x$, changes with any change of scale.

4.5 A New Approach to Combining Continuous and Discrete Entropy

When continuous and discrete information are produced simultaneously by the data fusion process, we require a joint measure of information quality. We have seen that continuous entropy is relative, because its value is infinite on the absolute scale, and discrete entropy is absolute. This difficulty was addressed by Deaves, when he considered optimal communications usage [18]. Deaves used a linear combination of the Track and Object Classification entropies to provide a joint measure.

Deaves wished to select the most valuable information for transfer on a data communications network with limited bandwidth. The information comprised a list of objects for which Track (continuous information: location, velocity, etc.) and Classification (discrete information: type, allegiance, etc.) was held. Communication was subject to the constraint that any transmission comprised both the Track and the Classification. When demand exceeded capacity, some objects' information would not be communicated. The utility measures used were of the form:

$$H = \omega H_t + (1 - \omega)H_c$$

where H_t was Track entropy, H_c was Classification entropy and the value of ω that maximised system performance was found by repeated trials. This approach proved highly successful and this very success gives cause for optimism that a combined measure can be defined.

Similar issues arise when an entropy Measure of Effectiveness for a data fusion system relates to both continuous and discrete information. However, in this case we wish to evaluate the system's performance rather than optimise it. Thus, the criterion used by Deaves to determine the combination of continuous and discrete information entropy is not available to us. The following new analytic approach to the calculation of combined continuous and discrete information entropy is proposed.

We start with an assumption: that there is a certain amount of information error in the discrete information which incurs a penalty of equal worth to a given amount of information error in the continuous information. We require entropy measures for continuous and discrete information error to have similar, aligned scales against which information errors of equal worth have equal entropies.

Firstly, we address the alignment. We will use the terms object track to denote the continuous information entity and object class to denote the discrete information entity.

An m-box enclosing an n-sigma confidence region around an object track which is sufficiently large for a second object, outside the confidence region, to be unambiguously recognised as separate with probability $1 - \varepsilon$ has volume:

$$v = (2n)^m \times \sqrt{|P|} \ ,$$

where P is the covariance matrix of a the track , m is the dimension of attribute space and n is chosen to deliver a particular desired ε.

Recalling the previous discussion of effectiveness, a track, $t0$, may be used effectively if it just satisfies all state estimation SoPs. If we denote the covariance of this track P_0 and choose $n = \sqrt{\frac{\pi e}{2}}$, then the $\log_e$ of the unambiguous volume around the track is its entropy, H_{t0}. (This gives a probability of ambiguity $\varepsilon < .001$ for $m \geq 2$ since the region, so defined, gives a ± 4.27 sigma interval in each dimension.)

$$\ln\{v\} = \ln\left\{(2n)^m \times \sqrt{|P_0|}\right\} = \ln\left\{\sqrt{|P_0|(\pi e)^m}\right\} = H_{t0}$$

The modified continuous entropy is then given by:

$$\tilde{H}_t = H_t - H_{0t} \qquad . \qquad (4.5.1)$$

The maximum number of unambiguous object track volumes v which may fit, in a rectangular arrangement, within the tracking region of interest is then $N \simeq \frac{V}{v}$ where V is the volume of the region of interest over which tracks may be encountered.

We note that $\ln\{N\}$ is the maximum entropy of the tracking process on a scale where unit volume is $(2n)^m \times \sqrt{|P_0|}$. This fact is drawn from the assumption of uniformly distributed errors over the region of interest and follows from the result of Shannon that: "...if x is limited to a certain volume, V, entropy is a maximum and equal to $\ln\{V\}$ when x is uniformly distributed over the volume with probability $\frac{1}{V}$. ..." [75]. $\tilde{H}_t$ is then the Track State entropy with respect to the same scale.

We note also that both $\tilde{H}_t$ and $\ln\{N\}$ are independent of the original scale of the continuous information. Continuous entropy has been normalised with respect to a new scale in a similar manner to the normalisation proposed by Bell [5]. However, this approach differs from that of Bell because the scale is not of arbitrary size. Rather, it is chosen specifically to place the origin to coincide with an information state that is just effective and to deliver the required properties of parity of scale with discrete Information entropy.

To align the scale, we require also to modify the datum of the discrete information entropy. If $c0$ is defined as the maximum entropy probability distribution that just satisfies all object class SoPs and H_{c0} is the entropy of that distribution, the aligned Discrete information entropy is then given by:

$$\tilde{H}_c = H_c - H_{c0} \qquad (4.5.2)$$

The combined entropy after alignment is given by:

$$\tilde{H} = \tilde{H}_t + \tilde{H}_c.$$

An approach such as this gives a measure of the combined entropy of the total information quality delivered by the system. Both the continuous and discrete components of entropy were normalised with respect to the entropy of an estimate which just satisfied the SoPs for the attributes concerned. Thus the datum for each was set at the level of the required information quality.

But, what of scale? If we regard the modification to the continuous information space as having discretised that space into cells of volume $(2n)^m \times \sqrt{|P_0|}$, an unambiguously separate track estimate may be used effectively when it is enclosed within a cell with probability $\geq 1 - \varepsilon$. This is analogous to the situation with discrete information where a classification may be used effectively if it placed the object in a class with probability $\geq 1 - \zeta$.

It is proposed that a track which is enclosed within α (>1) cells with probability $\geq 1 - \varepsilon$ is equally ineffective as a classification which placed an object in α classes with probability $\geq 1 - \zeta$. This is represented in figure 4.3.

The distributions of information error in both Track and Classification result in estimates spread over similar numbers of resolution cells/classes in similar proportions. We require the situations to have similar information entropies.

We can quantify the entropy in each case. For the continuous information to be distributed over α cells, its covariance determinant must be α^2 times that of the information just satisfying the SoPs. It follows:

$$\tilde{H}_t = H_t - H_{0t} = \ln\left(\sqrt{(2\pi e)^m a^2 |P_0|}\right) - \ln\left(\sqrt{(2\pi e)^m |P_0|}\right) = \ln a.$$

For the discrete case:

$$\tilde{H}_t = H_t - H_{0t}$$

$$\simeq -(1-\zeta)\ln\left(\frac{1-\zeta}{a}\right) - \zeta\ln\left(\frac{\zeta}{q-a}\right) + (1-\zeta)\ln(1-\zeta) + \zeta\ln\left(\frac{\zeta}{q-1}\right)$$

$$\simeq \ln a \text{, for small values of } \zeta \text{ where } q \text{ is number of classes.}$$

Figure 4.3: Equivalent Information Errors

Resolution cells in continuous space   Object classes
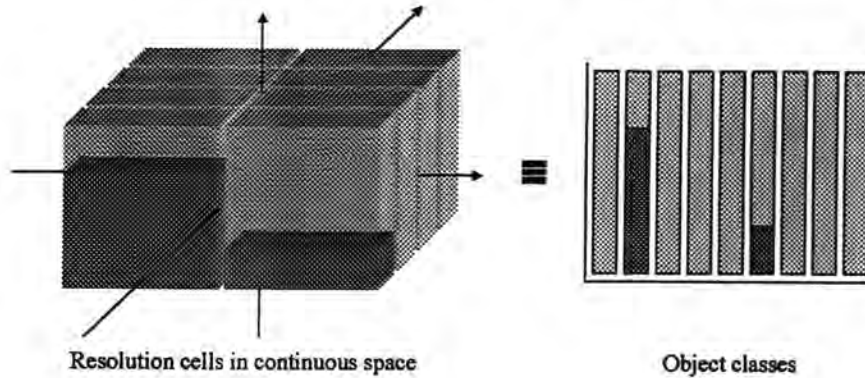
Similar uncertainties should lead to similar entropy measures.

It follows that the modified entropy measures defined in equations (4.5.1) and (4.5.2) approximate to a common scale if we assume that uncertainties in continuous and discrete information, spread over similar numbers of effective resolution cells and classes respectively, have information penalties of similar worth. This result is as expected; similar proportional changes in uncertainty should result in similar changes in the value of entropy.

For continuous information the spread of uncertainty is expressed as the ratio of m-volumes enclosed by contours of equal likelihood for the actual and required (SoP) error distributions. For discrete information, the spread of uncertainty is simply the number of classes for which actual support must be pooled before the required (SoP) support level is reached.

5 The Entropy of Estimation Error

The system of interest is described in terms of the objects that inhabit it and the objects in terms of the states of their attributes. Those attributes can belong to one of three classes: continuous, discrete with parameter (ordered) and discrete without parameter (non-ordered). These classes are represented in figure 5.1 and correspond to classes of random variable.

Figure 5.1: Attributes

The attributes that define the objects fall into three divisions.
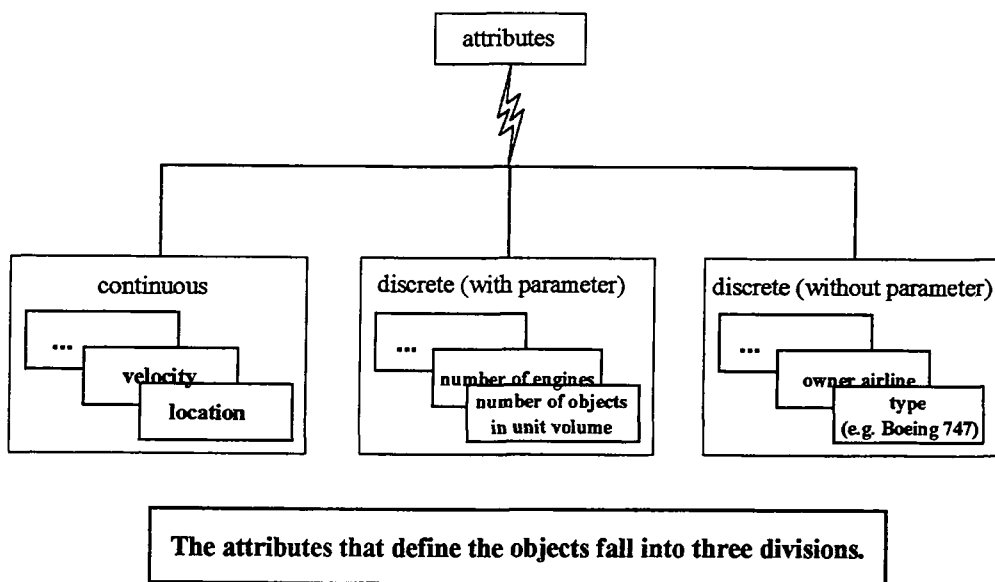
Continuous random variables are the most common form of numerical information error arising from a data fusion system. Attributes such as location and its time derivatives are subject to errors that fall within this class. Errors that are discrete random variables with parameter are encountered in attributes such as number of objects in a region or if the object is an aircraft the number of engines it has. Errors which are discrete random variables without parameter are encountered in attributes such as type or ownership/allegiance. These latter attributes are non-ordered in the sense that, irrespective of any enumeration applied to them, a value might be regarded as no more dissimilar from the furthest outlying value than it was from its neighbours. The nature of entropy of each of these classes of random variable is considered separately.

In this chapter the properties of entropy as a measure will be considered. To provide a useful MoE, entropy must be calibrated. If the entropy MoE at time τ has the value H, what does this mean? If the value of the MoE changes to $H + \delta H$ at time $\tau + \delta\tau$, what is the nature of the change in information quality? We require that entropy should undergo transformation with minimal re-evaluation and re-calibration and it must reflect well the dynamic nature of the uncertainties when a random system is observed.

Such dynamic uncertainties can lead to step changes and switching of the information error delivered by data fusion. This behaviour can also be regarded as possessing an entropy. Two views are possible, one giving the entropy of the equilibrium probabilities associated with the behaviours and another giving the entropy of the possible paths between a given pair of behaviours. The latter approach is a suitable subject for future research.

We will see in section 5.1.4 that entropy jointly measures the spread and the shape of the distribution of information errors in continuous numerical attributes. For distributions with finite covariance, the value of entropy can be expressed as $\ln\{|P|\} + B$ where P is the covariance matrix (the spread) of the errors in the information and B is a constant that indicates the distribution (the shape) of the errors. Whilst it is common to write the covariance of Gaussian distributed errors in this way, we show here that uniform and Laplace distributed errors have entropy which can be written in the same form, with simple exact expressions for B. This result is not found documented elsewhere. We speculate that this fact might offer the basis for a test of the validity of assumptions of Gaussian errors.

We begin this chapter by looking at the information error in attributes that are characterised as continuous random variables.

5.1 Continuous Random Variables

Much of the numerical information produced by data fusion will be concerned with parameter value and rate of change. Good examples of this are location and its time derivatives for an object moving through continuous space. The information error in these parameters may be expressed in terms of a vector of continuous random variables with an associated probability density. This density is the error probability distribution of our estimate and gives rise to a region of confidence around the stated values.

The quality of the information held in the statement is indicated by the extent of this region. A larger region of confidence indicates poorer information quality because larger differences between the estimated parameter values and the true ones are likely. Conversely, a smaller region indicates better information quality because large differences are less likely.

We will assume in this section that our continuous random variables are measured with respect to a Cartesian co-ordinate system with m dimensions. We assume further that errors are measured with respect to a common scale along each axis. The significance of the former assumption will become apparent later in this discussion.

Intuitively, we might seek a measure of the spread or variability of the error probability distribution to indicate error performance. Classical statistics offers various measures of spread. Common among these are [40]:

- Range,

- Mean deviation,

- Variance,

- Standard deviation,

- Covariance.

Range is defined as the difference between the largest and smallest scalar value in a sample. It is applied to empirical measurements where the sample size is small. Whilst,

it would be possible to define a version of range that could be applied to vector values, for instance the Euclidean distance, the measure would not be suited to an analytic approach. An analytical approach is proposed in this thesis and range will not be considered further.

Mean deviation is defined as $E[|\varepsilon|]$ where $E[\ ]$ is expectation and $|\varepsilon|$ is the modulus of the error in some scalar value. Mean deviation would extend readily to vector values but lacks an established basis for analytical treatment. In the past, before the advent of electronic calculators and laptop computers, the mean deviation was held to have advantages (over standard deviation) for manual calculation. It has fallen into disuse and will not be considered further except where it relates to information entropy.

Variance is defined as $E[\varepsilon^2]$ where ε is the error in some scalar value. Its vector form is the covariance matrix that will be discussed below and it has an established theoretical basis for analytical treatment. Standard deviation is defined as $\sqrt{\text{Variance}}$ but does not extend readily to vector values. The advantage of standard deviation over variance is that is a more intuitive measure. However, standard deviation and variance as indicators of information quality are equivalent.

Covariance is $P \triangleq E[\varepsilon\varepsilon^T]$ where ε is the error in some (column) vector value. Covariance is a matrix whose diagonal elements are the variances of ε_i, the elements of ε. The off-diagonal elements of the covariance matrix indicate the cross-variances between pairs of elements of ε, defined as $E[\varepsilon_i\varepsilon_j]$ where $i \neq j$. The errors in element i and j of ε are independent if an off-diagonal element is zero.

It is possible through singular value decomposition (SVD) to define a rotation of a rectilinear axis set, to find a transformation ε' of ε in which the elements are mutually independent [51]. The covariance with respect to that axis set is a diagonal matrix D with elements d_i and the (orthogonal) Jacobian matrix of the transformation is U made up of vectors e_i. Then $P = UDU^T$. For two dimensions, this is represented in Figure 5.2.

Covariance satisfies all but one of the criteria established in Chapter 2 for a measure of effectiveness; what it lacks is conciseness. This will not cause problems if the attribute space of interest is scalar or a vector of small dimension. However, if a large number of attributes are held for a large number of objects, the covariance matrix for the errors in an estimate of the state of attribute space would be very large and, hence, not an informative measure. For a single object, simplification is often possible by rotation of the axes in the manner described above. However, there is no guarantee that this will be similar to the axis set of interest (in which SoPs may have been defined) and, for multiple objects, it is unlikely that the transformation would have the desired effect on more than one of them.

Figure 5.2: Uncertainty Ellipsoid in 2-D Space [51]
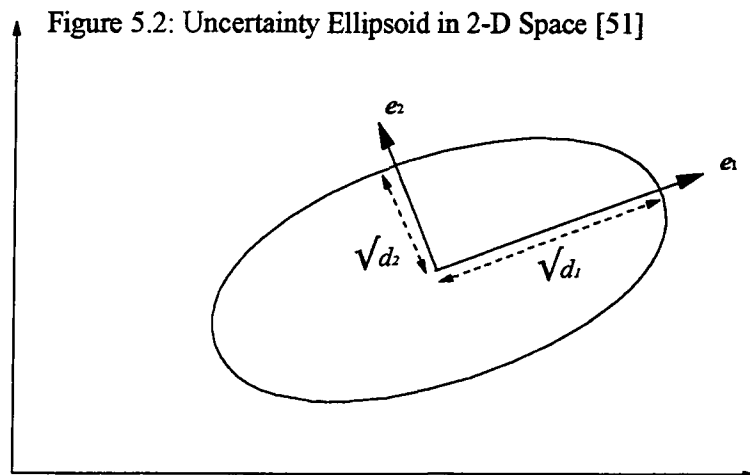
The covariance matrix defines an ellipsoid centred on the location of the estimate.

It would be possible to take an ad-hoc approach to reducing the covariance to a scalar by means of its trace or determinant and Mori, Chang and Chong use a measure of spread called average standard deviation that is defined $\bar{\sigma} \triangleq |P|^{\frac{1}{2m}}$, where m is the dimension of the attribute state vector [18] [43] [55]. In the axis set where covariance is a diagonal matrix, $\bar{\sigma}$ is the geometric mean of the standard deviations of the individual attributes.

The approach preferred here is to take the relative entropy of the error distribution. By adopting relative entropy we change the focus of our search for a measure which tells us more than just the size of the error distribution. Entropy allows us to examine the spread of a distribution in terms of the m-volume occupied by the mean squared error plus an increment related to the shape of the distribution.

Relative entropy (H) was defined in Chapter 4 as:

$$H(\varepsilon) = \int_{-\infty}^{+\infty} p(\varepsilon) \ln\left[\frac{1}{p(\varepsilon)}\right] d\varepsilon. \qquad (5.1.0.1)$$

More simply, this may be written as [51]:

$$H(\varepsilon) = -E[\log \text{likelihood}(\varepsilon)]$$

Deaves found that the determinant of the covariance may be used in place of information theoretic measures for sensor and information management [18]. We will see in the following sections that, for the continuous distributions considered here, the estimation error entropy takes the form:

$$H = \tfrac{1}{2} \ln[|P|] + B$$

where B is a constant. Deaves argued that because ln is a monotonic increasing function $|P|$ may be used in place of an information theoretic measure and produced good results using this simplification. However, we will see that the constant B is different for each distribution studied and, for the purposes of this work, entropy reveals more about the uncertainty, and hence the information quality, than covariance determinant alone.

5.1.1 Multivariate Gaussian Distributed (MVGD) Errors

Many data fusion systems produce numerical information with errors that approximate to a Multivariate Gaussian Distribution. Errors with this distribution are of particular interest for the following reasons:

- they arise in many real-world fusion scenarios,

- they lend themselves to analytical treatment using standard methods.

Consider an m-dimensional Gaussian-distributed error vector ε with zero mean and covariance P. To evaluate relative entropy, H, the expression given in equation (5.1.0.1) will be used:

$$H(\varepsilon) = \int_{-\infty}^{+\infty} g(\varepsilon) \ln\left[\frac{1}{g(\varepsilon)}\right] d\varepsilon$$

where [40]:

$$g(\varepsilon) = \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|P|}} \exp\left(-\frac{1}{2}\varepsilon' P^{-1} \varepsilon\right) \text{ and } P = E[\varepsilon\varepsilon'].$$

Shannon gives the expression for the relative entropy [75]:

$$H(\varepsilon) = \ln\left\{(2\pi e)^{\frac{m}{2}} |\{a_{ij}\}|^{-\frac{1}{2}}\right\}$$

where the matrix $\{a_{ij}\}$ is P^{-1} in the notation used here. This may be rewritten as:

$$H(\varepsilon) = \frac{1}{2}[\ln\{|P|\} + m(\ln\{2\pi e\})]$$

We could relate this to a region bounded by a surface of constant likelihood arising from a multivariate Gaussian error distribution. Using the SVD approach of Manyika and Durrant-Whyte, the surface is an m-dimensional ellipsoid with its axes aligned with the rectilinear axis set in which the elements are mutually independent and in which the covariance is a diagonal matrix [51]. It follows from the results of Shannon discussed in Chapter 4 that the entropy does not change when the axes are rotated [75].

We could consider a rectangular m-box aligned with the transformed axis set with sides of length σ_i' (the standard deviation of ε_i') as a simpler alternative. The m-box has m-volume $V = \sqrt{|P|}$. As a consequence we may write:

$$H(\varepsilon) = \ln\{V\} + G \tag{5.1.1.1}$$

where G is a constant particular to the m-dimensional Gaussian distribution.

5.1.2 Multivariate Uniform Distributed Errors

Next, consider the uniform distribution. In any situation where position is declared as a grid reference and the object in question is small and slow-moving in

relation to the granularity of the grid, the errors in object position will have an approximate uniform distribution. For example, initial sightings of survivors of an accident at sea might be passed to their rescuers as map references.

This is an example of a 2-dimensional uniform error distribution. We will consider a general m-dimensional uniform distribution. If the error is distributed uniformly over the rectangular m-box with length of side $\delta_1 \ldots \delta_m$ with probability $\frac{1}{\delta_1 \times \ldots \times \delta_m}$. We can calculate the entropy of the distribution:

$$H(\varepsilon) = \int_{\varepsilon min_1}^{\varepsilon max_1} \ldots \int_{\varepsilon min_m}^{\varepsilon max_m} \frac{1}{\delta_1 \times \ldots \times \delta_m} \ln\{\delta_1 \times \ldots \times \delta_m\} d\varepsilon_1 \ldots d\varepsilon_m$$

where $\delta_i = \varepsilon max_i - \varepsilon min_i$. Which gives the expression for relative entropy:

$$H(\varepsilon) = \ln\{\delta_1 \times \ldots \times \delta_m\}$$

So, the entropy of an m-variate uniform error distribution is the natural logarithm of the m-volume of the region over which ε is distributed [75]. We note the following similarity with the result given for the MVGD errors in equation (5.1.1.1). The standard deviation of ε_i is $\sigma_i = \frac{\delta_i}{2\sqrt{3}}$, hence:

$$H(\varepsilon) = \tfrac{1}{2}\ln\{|P|\} + m\ln\{2\sqrt{3}\}, \text{ which can be written:}$$

$$H(\varepsilon) = \ln\{V\} + U.$$

Again, V is the volume of a rectangular m-box with sides of length σ_i' and U is a constant particular to the m-dimensional uniform distribution. Shannon identifies the uniform distribution as the maximum entropy distribution for a variable bounded within a region [75].

5.1.3 Multivariate Laplacian Distributed Errors

Thirdly, consider the Laplace distribution. When Daeipour and Bar-Shalom applied the IMM approach to the problem of tracking with observations subject to glint noise, they modelled the observation errors as a mixture of MVGD and Laplacian errors

[16]. Glint is such that not all observations come from the same place on the object of interest. Errors from the primary focus of observations were modelled as MVGD whilst errors from the glint returns from secondary foci were modelled as Laplacian.

Also, when Wu and Fitzgerald reported work on the detection of discontinuities in time series with non-Gaussian noise characteristics, the model they considered was the Laplace distribution [86]. These examples of Laplacian distributed errors were discussed in Chapter 3 in the section concerned with tracking algorithms.

Laplacian distributed errors in one dimension are characterised by the density function:

$$l(\varepsilon, \eta) = \frac{1}{2\eta} \exp\left(-\frac{|\varepsilon|}{\eta}\right).$$

where $\eta = \frac{\sigma}{\sqrt{2}}$ and σ is standard deviation. (η is the mean deviation of ε.) We extend this to m dimensions assuming independent ε_i [16]:

$$l_m(\varepsilon, \eta) = \prod_{i=1}^{m} l(\varepsilon_i, \eta_i) = \frac{1}{2^{\frac{m}{2}} \sqrt{|P|}} \exp\left(-\sqrt{2} \sum_{i=1}^{m} \frac{|\varepsilon_i|}{\sigma_i}\right).$$

The entropy of these Laplacian distributed errors in m dimensions is given by:

$$H(\varepsilon) = -\frac{1}{2^{\frac{m}{2}} \sqrt{|P|}} \int_{-\infty}^{\infty} \ln\left\{\frac{1}{2^{\frac{m}{2}} \sqrt{|P|}} \exp\left(-\sqrt{2} \sum_{i=1}^{m} \frac{|\varepsilon_i|}{\sigma_i}\right)\right\} \exp\left(-\sqrt{2} \sum_{i=1}^{m} \frac{|\varepsilon_i|}{\sigma_i}\right) d\varepsilon,$$

$$H(\varepsilon) = \frac{1}{2} \ln\{|P|\} + m \ln\{\sqrt{2}\,e\}.$$

Which obeys a similar equation to that obeyed by the MVGD errors (5.1.1.1):

$$H(\varepsilon) = \ln\{V\} + L.$$

V is again the m-volume of a rectangular m-box with sides of length σ_i' and L is a constant particular to the m-dimensional Laplace distribution.

5.1.4 To Calibrate Relative Entropy of Continuous Random Errors

When relative entropy changes by a known amount, we wish to know what change in information quality is implied. So, H must be calibrated. To calibrate H we will consider a unit change in relative entropy. We have $H^+ = H + 1$ and we seek a

corresponding change in the error covariance. From equation (5.1.1.1) for MVGD errors:

$$H^+ = \frac{1}{2}[\ln\{|P|\}] + G + 1$$

$$= \frac{1}{2}[\ln\{|P|\} + 2 \times \ln\{e\}] + G$$

$$\Rightarrow |P^+| = |P| \times e^2$$

Replacing G with L or U extends this to the Laplacian and uniform distributions respectively. Thus, a known change in relative entropy can be translated directly into a proportional change in the determinant of the error covariance.

We recall equation (4.2.0.2) from Chapter 4 the full expression for the entropy of a continuous random variable. When this is applied to our error probability distribution:

$$H(\varepsilon) = \int_{-\infty}^{\infty} g(\varepsilon) \ln\left[\frac{1}{g(\varepsilon)}\right] d\varepsilon - \lim_{\Delta\varepsilon \to 0} \ln \Delta\varepsilon$$

A change of scale, by a factor e, caused $\lim_{\Delta\varepsilon \to 0} \ln \Delta\varepsilon$ to be altered by 1. In this case, $\lim_{\Delta\varepsilon \to 0} \ln \Delta\varepsilon$ is unchanged and the relative entropy changes as a result of a change in information quality.

The scale in this case is that of the m-dimensional attribute space. $|P|$ has the same units as m-volume^2. When H changes by one, the proportional change in $|P|$ is e^2 rather than e. However, observing equation (5.1.1.1) we note that the m-volume (V) of the m-box with sides of length σ_i' undergoes a proportional change of e. This indicates that this proportional change holds true in general.

Thus, a known change in entropy can be translated directly into a proportional change in the m-volume of the confidence region surrounding an estimate defined by the mean squared error. This implies that if two (or more) estimates of the same X are available differences in their relative entropy values serve not only to rank them in order of information quality but also determine the ratio of the m-volumes of their regions of confidence.

However, this is not the only calibration of the relative entropy that might be made. Figure 5.3 shows the 2-dimensional MVGD, Laplacian and uniform distributions drawn to have identical covariance matrices, $P = I_2$ (the identity matrix of dimension 2). Whilst measures such as $|P|$, trace(P) or average standard deviation, $\bar{\sigma} = |P|^{\frac{1}{2m}}$, would find samples from these three distributions identical, entropy is capable of discriminating between them.

Figure 5.3: Error Distributions with a Common Covariance

a) Gaussian Distributed Errors

b) Laplacian Distributed Errors

c) Uniformly Distributed Errors

d) The Area $\sqrt{|P|}$
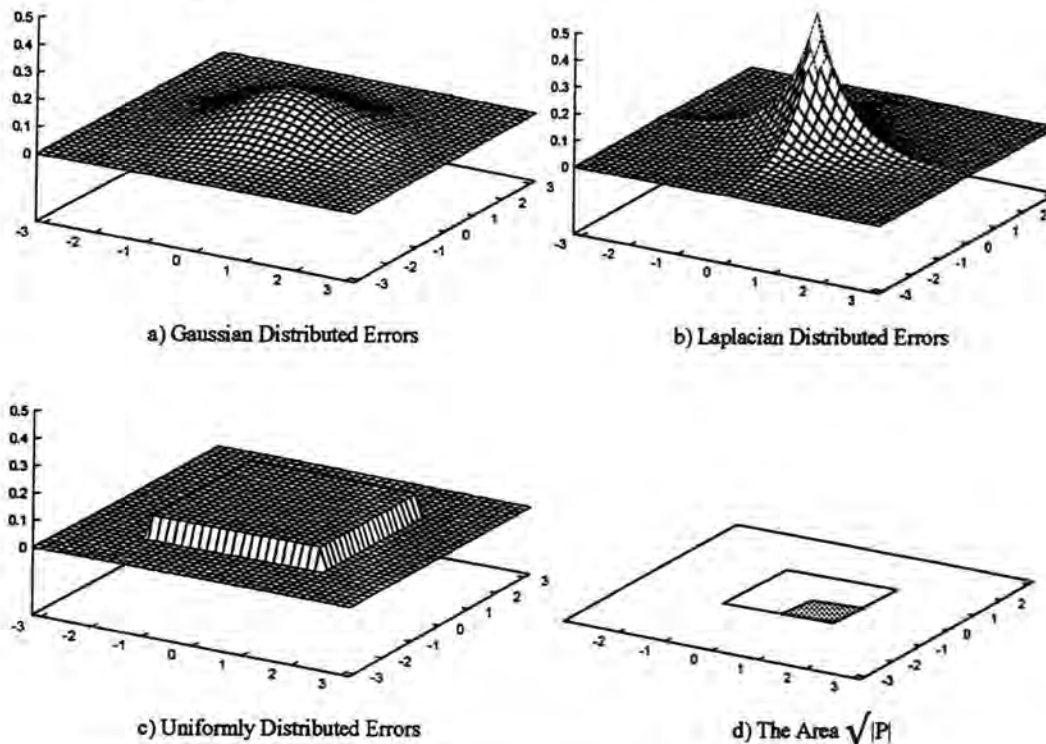
Three distributions with the same covariance have different Information Entropies.

In sections 5.1.1 to 5.1.3 the entropies of the MVGD, Laplacian and uniform distributions were written in the form:

$$H(\varepsilon) = \ln\{V\} + B,$$

In figure 5.3 d), the outline square is drawn at $\pm\sigma$ about the origin. The shaded square has area $\sqrt{|P|}$. In a 2-dimensional system with covariance I_2 the shaded square is an m-box of area (m-volume) V. It is noted with interest that for these three distributions:

$$b = \frac{H(\varepsilon) - \frac{1}{2}\ln\{|P|\}}{m},$$

and b is a different constant, independent of covariance and dimension, for each distribution:

- for MVGD errors $b = g = \ln\{\sqrt{2\pi e}\} = 1.419;$

- for Laplacian errors $b = l = \ln\{\sqrt{2}\,e\} = 1.347;$ and

- for uniform errors $b = u = \ln\{2\sqrt{3}\} = 1.242.$

There is a characteristic constant associated with each of these distributions. This relationship has not been found reported elsewhere in the literature. It may offer the capability to test the truth of assumed distributions particularly MVGD. For MVGD errors the constant must be unique since, for a given covariance the entropy of MVGD errors is a maximum [75].

5.1.5 Transformations of Systems with Continuous Random Errors

In this section we will consider some transformations with particular relevance to data fusion processes. Shannon considers a general transformation of the form $z = \phi(x)$ where $\phi(x)$ is a vector valued function, then the density function of the errors with respect to the transformed axes is [75]:

$$f(z) = \left[\frac{\partial x}{\partial z}\right]g(x) \text{ and } H_z = H_x - E\left[\ln\left\{|\frac{\partial x}{\partial z}|\right\}\right].$$

More conveniently, where Φ is the Jacobian matrix of the inverse transformation, $\left[\frac{\partial z}{\partial x}\right]$, for the linear transformations which are of interest here:

$$H_z = H_x + \ln\{|\Phi|\}$$

If the determinant of the matrix Φ is known, the effect of the transformation on relative entropy can be predicted. Similarly, the effect on covariance determinant:

$$P_z = \Phi P_x \Phi^T \tag{5.1.5.1}$$

We will use these results to examine common transformations, undergone by observed data and partially fused information in data fusion processes. Multiple distributed data fusion centres, responsible for fusing a local subset of the observations need not all operate with respect to a common, global spatial definition. The transformations required to perform the mappings between spatial definitions are:

- translation, where spaces are defined with respect different local origins,

- scaling, where units vary between spatial definitions,

- rotation, where the angle datum varies,

- time transition, where observations are asynchronous.

Transformations between polar and Cartesian axis sets may be regarded as compound transformations where the components are translation, rotation and scaling.

A change in origin leads to a transformation of the form:

$$z = x - w \quad .$$

Effectively, the origin has moved from 0 to w. The Jacobian of this transformation is the identity matrix and $H_z = H_x$. This is true of any measure preserving transformation [75].

A change of scale, by a constant factor a, affecting one of the attributes of x gives:

$$\Phi = \text{diag} [1 \ \dots \ 1 \ a \ 1 \ \dots \ 1 \ ] \ \text{so} \ H_z = H_x + \ln\{a\}.$$

When n attributes undergo the same scaling $H_z = H_x + n \times \ln\{a\}$ or if each undergoes a different scaling $H_z = H_x + \Sigma \ln\{a_i\}$.

The value of the relative entropy MoP has changed by a constant amount and the position of zero on the H axis has changed. It can be seen that the relative term changes by the same amount as the $\lim_{\Delta_\varepsilon \to 0} \ln \Delta_\varepsilon$ term of equation (4.2.0.2), so that the net effect on the total entropy is no change. For practical purposes we will use the relative entropy term and must take due account of changes of scale. We note also that, when the scaling is by a constant factor (for example metres to kilometres) and H is a function of an

independent variable (for example time), its derivatives are not affected by the change of scale. Thus, entropy gradients, as measures of rate of convergence could be compared between estimates of the same dimension without reference to the measurement scale [75]. However, this is not true of polar to Cartesian transformations where scaling is performed with respect to the distance variable.

This is an important feature of polar co-ordinate systems and the reason for the Cartesian assumption adopted throughout this section. The numerical value of relative entropy for a given region in a polar axis set changes depending on its distance from the origin. For a spherical axis set the value changes with elevation rotation also. This occurs because the extent of Δ_ε and hence $\lim_{\Delta_\varepsilon \to 0} \ln \Delta_\varepsilon$ is itself a function of the distance parameter and is not uniform over the entire space. For example, comparing 3-dimensional Cartesian co-ordinates with spherical co-ordinates [22]:

$$\Delta_\varepsilon(\text{cartesian}) = \Delta_{\varepsilon_x}\Delta_{\varepsilon_y}\Delta_{\varepsilon_z} \quad \text{whilst} \quad \Delta_\varepsilon(\text{spherical}) = r^2 \cos \psi \Delta_{\varepsilon_r}\Delta_{\varepsilon_\theta}\Delta_{\varepsilon_\psi} .$$

This property is not unique to entropy. The same problems arise with any measure of spread related to m-volume including $|P|$, trace(P) or average standard deviation.

Another common transformation in data fusion applications is rotation. Axis rotation will not change the entropy of the estimation error because it is another measure preserving transformation. The angular orientation of the axes should have no bearing on the quality of information present. If $\mathbf{x}$ is the Cartesian position co-ordinate pair (x, y), and the axes are rotated through angle θ :

$$\Phi = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix},$$

then $|\Phi| = 1$, and $H_z = H_x$.

Finally, we note that we may take the same approach to transformations which result from a change in our temporal point of view as we did for the spatial point of

view. Transitions through time for non-random X will have no impact on the quality of the information present. As an example: if x comprises x and its first n time derivatives:

$$x = \begin{bmatrix} x & x^{(1)} & x^{(2)} & \cdots & x^{(n)} \end{bmatrix}^T$$

then given $x(t)$, $z = x(t + \tau)$ may be approximated by the Taylor series expansions:

$$x(t+\tau) \approx x(t) + \tau \times x^{(1)}(t) + \ldots + \frac{\tau^n}{n!}x^{(n)}(t),$$

$$x^{(1)}(t+\tau) \approx x^{(1)}(t) + \tau \times x^{(2)}(t) + \ldots + \frac{\tau^{n-1}}{(n-1)!}x^{(n-1)}(t),$$

and so on, so that the Jacobian, $\left[\frac{\partial x(t+\tau)}{\partial x(t)} \right]$:

$$\Phi \approx \begin{bmatrix} 1 & \tau & \cdots & \cdots & \frac{\tau^n}{n!} \\ 0 & 1 & \tau & \cdots & \frac{\tau^{n-1}}{(n-1)!} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & \tau \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \text{ and } |\Phi| \approx 1, \ H_z \approx H_x.$$

Recalling that if X is non-random, the above is sufficient to describe the effect on X of a transition through time. The information we hold about the state of X at t does not degrade with time because the state at t defines completely the state at time $t + \tau$.

If X is random, the above is sufficient to describe the deterministic part of its transition through time. More simply, the above describes the way in which the transition through time would proceed if no random disturbance to X were to take place in the interval t to $t + \tau$. However, X is subject to random disturbances. So the state at t, predicted on the basis of its state at $t + \tau$, is subject to extra uncertainty and the increase in relative entropy, in the intervals between observations, will occur as predicted in Chapter 3. The equation for this transition through time is no longer the same as equation (5.1.5.1). Rather, it takes the form:

$$P_z = \Phi P_x \Phi^T + Q(\tau)$$

where $Q(\tau)$ is the covariance of the unobserved random disturbances to X in the interval.

5.2 Discrete Random Variables with Parameter

Some of the numerical information produced by data fusion will be concerned with number. A good example of this is the number of objects in a given region. The information error in any such statement is likely to be expressed in terms of a scalar integer random variable with an associated probability density. Like the error in continuous random variables, this density defines the error probability distribution of our estimate and gives rise to a region of confidence around the estimated value. Again, the quality of the information held in the statement is indicated by the size of this region with a large region of confidence indicating poorer information quality and a smaller region indicating better information quality.

Again, classical statistics offers some measures of spread [40]:

- Range,

- Mean deviation,

- Variance,

- Standard deviation.

Range and mean deviation are rejected for the reasons cited previously and covariance is not relevant to a single scalar value.

This leaves variance and standard deviation that, we have already noted, are equivalent as indicators of information quality. For a scalar value, variance satisfies all of the criteria established in Chapter 2 for a measure of effectiveness but, to maintain consistency with our approach to continuous random variables, we will continue to use the entropy of the error probability distribution. Also, we will see later that entropy extends naturally to include discrete variables without parameter for which variance has no meaning.

In this section, we are interested in a discrete random variable with one or more

parameters. This type of variable will have a distribution function of the form

$P(x) = F(x, \lambda, \mu, ...)$ where $\lambda, \mu, ...$ are the parameters of the distribution function.

5.2.1 Estimating the Parameter of the Poisson Distribution

Many data fusion systems produce discrete numerical information. We will

consider again the number of objects in a given region. If objects are approximately

uniformly distributed over the wider region of interest and the expected number of

objects in a unit volume is λ the number of objects x in a unit volume selected at random

would be assumed to have the Poisson distribution:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!} \ .$$

If we calculate the entropy of this frequency function, using equation (4.2.0.1):

$$H(x) = - \sum_{n=0}^{\infty} \ln\left\{ \frac{\lambda^x e^{-\lambda}}{x!} \right\} \frac{\lambda^x e^{-\lambda}}{x!} \ ,$$

$$H(x) = E[-x\ln\{\lambda\} + \lambda + \ln\{x!\}] \qquad\qquad (5.2.1.1).$$

Using Stirling's equation this becomes [71]:

$$H(x) = E\left[x\ln\left\{ \frac{x}{\lambda} \right\} + \frac{1}{2}\ln\{2\pi x\} + O\left(\frac{1}{x}\right) \right]$$

We require the entropy of the error, ε, in an estimate of λ based on a single sample, x.

The frequency function of the distribution of ε is:

$$P(x) = \frac{\lambda^{\hat{x}-\varepsilon} e^{-\lambda}}{(\hat{x}-\varepsilon)!} \ \text{where } \hat{x} - \varepsilon \in \mathfrak{I}^+ \text{ (the set of non-negative integers).}$$

With entropy as per equation (5.2.1.1). More conveniently, we use the Gaussian

approximation to the Poisson distribution. For sufficiently large λ:

$$H(\varepsilon) \approx \frac{1}{2}(\ln 2\pi\lambda e).$$

Assume now that we obtain, simultaneously, samples of the number of objects in

n randomly selected unit volumes. The entropy of the mean of n samples from the

Poisson distribution is approximated by the entropy of the error in the mean of n samples from the appropriate Gaussian approximation:

$$H(\varepsilon) \approx \tfrac{1}{2}\left(\ln 2\pi \tfrac{\lambda}{n} e\right).$$

Stuart and Ord give the distribution of the mean of n samples from a Poisson distribution [40]:

$$P(x) = \frac{(n\lambda)^x e^{-(n\lambda)}}{x!}.$$

Where x is on a scale with an interval $\frac{1}{n}$ times the original interval. With respect to the new interval, this has mean and variance of $n\lambda$. However, with respect to the original interval, it has mean λ and variance $\frac{\lambda}{n}$.

The distribution has become a better fit to its Gaussian approximation than was obtained for single sample because the Poisson parameter is n times larger. Its entropy is also better approximated.

5.3 Discrete Random Variables without Parameters

Many data fusion systems produce non-ordered, non-numerical information as well as numerical information. For example, an air traffic control system may hold the identity of each aircraft for which it is responsible. Such a system is performing object classification and the classification process is subject to uncertainty. Thus, the object class attribute has an error distribution and we require some means of measuring the spread of this distribution. The object class attribute is an example of a Discrete Random Variable without parameters. The information it holds may be described as qualitative.

Classical statistics is less forthcoming with measures of spread for distributions of this type. The possibilities include:

•Entropy,

•Information,

•Minimal error probability.

Shannon first proposed the information theoretic approach based on the entropy measure and Woodward first applied the approach to Radar observation processing [75] [85]. Some basic results concerning entropy were discussed in detail in Chapter 4. Manyika & Durrant-Whyte and Deaves et. al. favour another information measure known as Information [18] [19] [51]. Information is defined as $-1 \times$ entropy. Entropy and Information are interchangeable for the purposes of this work.

Minimal error probability is defined as $1 - c_{max}$, where c_{max} is the probability associated with the most likely object class. A similar measure was used by Pal and Pal as the basis for an alternative definition of entropy and Feder examined the relationship between entropy and minimal error probability [24] [63]. Minimal error probability has the weakness that it does not extend readily to continuous random variables.

The distribution of errors in an object classification derived from one or more observations may be represented as a vector of probabilities:

$$c_i = \begin{bmatrix} c_{i1} & \cdots & c_{iq} \end{bmatrix}^T \tag{5.3.0.1}$$

where c_{i1} to c_{iq} are the probabilities that the observed object, which belongs to class i, is classified as belonging to one of q classes. The entropy of this distribution is the entropy of object classification when the true class is i.

If we wish to generalise further, it is possible to define a matrix of probabilities of observing each class when the true class is any of the q classes:

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1q} \\ \vdots & \ddots & \vdots \\ c_{q1} & \cdots & c_{qq} \end{bmatrix} \tag{5.3.0.2}$$

Note that C is not a single probability distribution. It is an array of q distributions and:

$$\sum_i \sum_j c_{ij} = q$$

Thus, these distributions may be characterised. Shafer described this representation (C) as the Cartesian product of the true set and the set of possible outcomes, where the observation process is regarded as a statistical trial [74].

Shafer argued that aleatoric (chance) and epistemic (belief) probabilities are separate things and that the latter need not obey the same mathematical rules as the former. He cites the Bayesian approach to classification as an example of the rules governing chance being applied to belief. If we start with a matrix C that defines the probabilistic relationship of certain observable characteristics to the object class, when that collection of characteristics is observed, Bayesian theory offers a means by which the probability of each class may be calculated. By probability, we certainly mean our degree of belief rather than chance. The class of the object is not subject to chance. It is simply what it is.

If C is fully defined and the q classes form an exhaustive set of distinct classes, the Bayesian assumption is well founded and belief might be expected to behave exactly like chance. On the other hand if C is partially defined, if the classes are not distinct or classes exist that we know nothing about, the Bayesian approach offers only a coarse approximation. Shafer developed an alternative approach capable of making allowance for these possibilities based on a generalised variation of the Bayesian approach which had been proposed a decade earlier by Dempster [20] [74].

Once a decision has been made we are able to talk about the chance that it was the right one. Given prior information about the likelihoods of encountering certain classes based on empirical data, and given prior information about the sensing behaviour of our classification sensors when faced with these classes, we might predict the chance that we classify them correctly when they occur. We will see in Chapter 6 that concentration on the entropy of belief arising from a data fusion process leads to a different model to one based on the entropy of the decision.

For the purposes of this work we will concentrate on the Bayesian approach, noting that Shafer's approach might offer a fruitful area for future work. We will concentrate on the probability of error in a given decision rather than the mechanism by which support for (belief in) the decision is calculated. We will also adhere to the Bayesian model as a common thread running through the subject. Shafer does not say that the Bayesian model is untrue, rather, that situations will arise in which we are required to classify objects without access to all the components of the Bayesian model.

5.3.1 The Entropy of Object Classification

To evaluate H, the expression for the entropy of Discrete Random Variables of Chapter 4 is used. For a particular true object class, i, the distribution of errors has the form of equation (5.3.0.1), and:

$$H(c_i) = \sum_j c_{ij} \ln\left[\frac{1}{c_{ij}}\right]$$ (5.3.1.1)

Thus, for a specific object class, i and finite q, $H(c_i)$ may be evaluated by enumeration.

If we are concerned with the performance of the system when it observes any unspecified object class of q, the array of distributions of equation (5.3.0.2) is appropriate. A convenient definition for the entropy of this matrix is:

$$H(C) = \sum_i \sum_j c_{ij} \ln\left[\frac{1}{c_{ij}}\right]$$

However, a more useful MoP will be provided by:

$$\bar{H}(c) = E[H(c_i)] = \frac{1}{q}H(C)$$

5.3.2 To Calibrate the Entropy of Discrete Random Variables without Parameters

It is not meaningful to calibrate the entropy of qualitative information in the same way as numerical information. The quality of the information is not measured by the likely spread of errors around the true object attribute. C has no parameter and the spread has no spatial meaning. An estimate is either right or wrong and all wrong

estimates might be regarded as equally wrong. Entropy is a measure of how certain we are regarding the object class. The issue is confidence, not accuracy.

It is important to note that entropy is measuring (un)certainty rather than accuracy or correctness. The wrong answer will deliver a better (lower) entropy than no answer at all. We will see later in this discussion that the maximum entropy corresponds to ignorance, defined as the state in which all classes are equally likely. In Chapters 6 and 7 the apparent advantageous affect on performance of any answer, right or wrong, will play a part in the design of models and the interpretation of their results.

Consider the entropy regarding a particular object class, $H(c_i)$. It can be seen from equation (5.3.1.1) that:

$$0 \leq H(c_i) \leq \ln[q]$$

where 0 corresponds to the situation where the class is known with certainty and $\ln[q]$ corresponds to the situation where nothing is known and all classes are equally likely.

If it is required to minimise the probability of a classification error, the declared object class j is chosen such that:

$$j = \arg \max_{k} (c_{ik}) \text{ and } c_{\max} = c_{ij}$$

Feder denoted the minimal error probability as $\pi = 1 - c_{\max}$ [24]. When $H(c_i) = 0, \pi = 0$ (its minimum value) and when $H(c_i) = \ln[q], \pi = \frac{q-1}{q}$ (its maximum). Feder went on to show that, for any other value of π, the entropy of object classification is bounded above and below such that:

$$H_{\min}(c_i) \leq H(c_i) \leq H_{\max}(c_i),$$

then:

$$H_{\max}(c_i) = -(1-\pi)\ln[1-\pi] - \pi\ln\left[\frac{\pi}{q-1}\right] \tag{5.3.2.1}$$

In Chapter 6, this expression is used to provide maximum entropy estimates of distributions for which only the entropy is known. Given H, equation (5.3.2.1) may be solved numerically to give π. It is then a simple matter to create a maximum entropy distribution of the form of Figure 5.4. This approach will provide part of a new model of object classification that will be proposed in Chapter 6.

Figure 5.4: Maximum Entropy Distribution

$\pi = .7, q = 5$
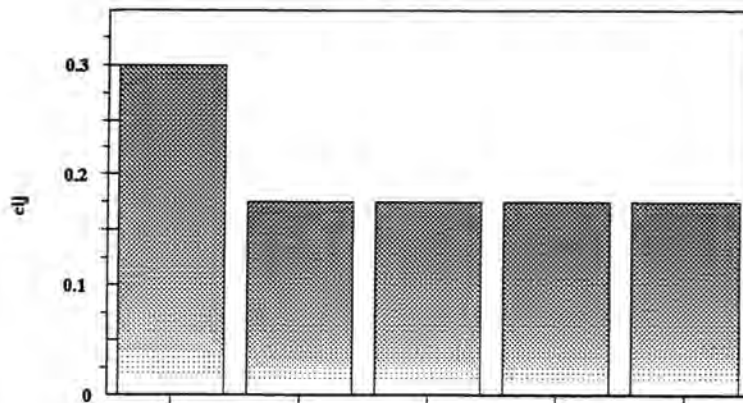
It is possible to visualise the error distributions that give rise to the maximum and minimum entropies. Figure 5.4 shows the maximum entropy distribution when $\pi = 0.7$ and $q = 5$. The classes have been reordered for the sake of clarity so that the most probable class appears on the left of the diagram. We see that there is a single most probable class and that all other classes are equally probable.

> Maximum entropy occurs for a given c_{max} when there is one most likely class and all other classes are equally likely.

Given integer r in the range 1 to $q - 1$ satisfying $\frac{r-1}{r} \leq \pi \leq \frac{r}{r+1}$ the expression for minimum entropy given by Feder was:

$$H_{min}(c_i) = -(r\pi - r + 1)\ln[r\pi - r + 1] - r(1 - \pi)\ln[1 - \pi]$$

The form of the minimum entropy distribution shown in figure 5.5 depends on the parameter r. In general, there are r equal most probable classes and one less probable class that attracts the remaining probability. All other classes have probability zero. The simplest form of the minimum entropy distribution occurs when $\pi < \frac{1}{2}$ and $r = 1$. Under

Figure 5.5: Minimum Entropy Distribution
$\pi = .7, q = 5$
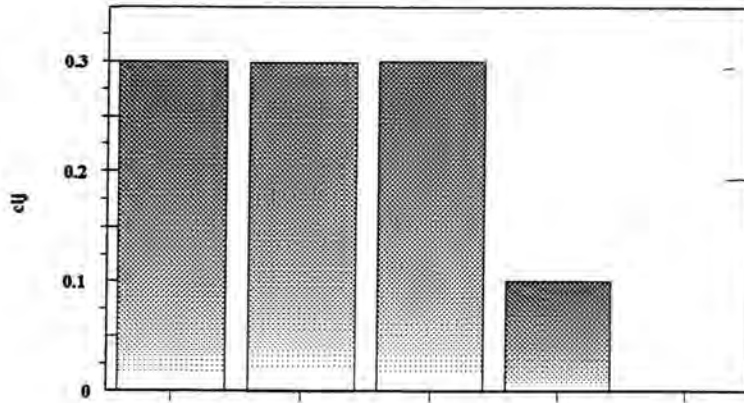
these circumstances, there are two probable classes: one with probability c_{max} and one

with probability π. The distribution becomes more complex as the minimal error

probability grows.

> Minimum entropy occurs for a given c_{max} when there are r most likely classes, one class to
> which the remaining support attaches and all other classes with zero probability.

The minimum entropy distribution is shown in figure 5.5 when $\pi = 0.7$ and $q = 5$.

Here, $\frac{2}{3} < \pi < \frac{3}{4}$ and $r = 3$. Again, the classes have been reordered for clarity so that the

most probable classes occur on the left and the least probable on the right. Finally, for a

given q we can plot maximum and minimum entropy against minimal error probability as

in Figure 5.6.

There is a vertical line drawn at $\pi = 0.7$. The intersections between the vertical

line and the maximum and minimum entropy lines correspond to the situations

represented in figures 5.4 and 5.5 respectively.

Figure 5.6: Maximum and Minimum Entropy of Object Classification
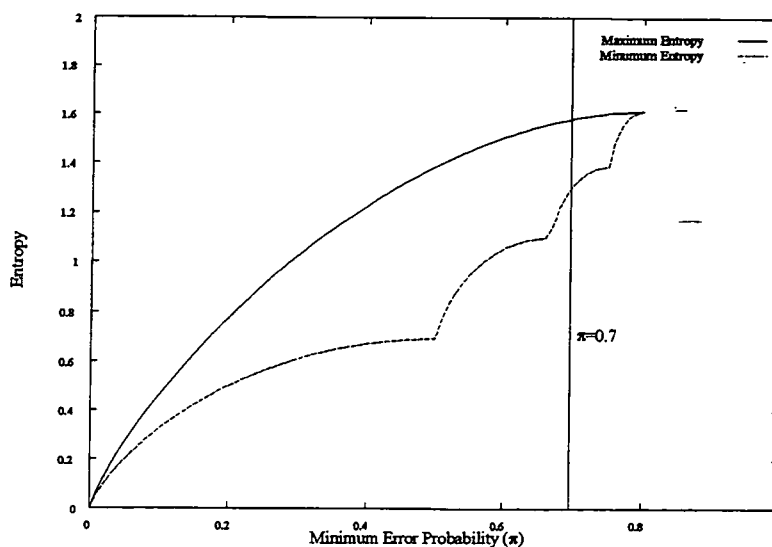
Maximum and minimum entropies are plotted against minimal error probability.

5.4 The Entropy of a Markov Process

When Shannon proposed that entropy, established in the field of statistical mechanics, should be applied to information dynamics, he started by examining stochastic processes as representations of the patterns that occur in language [75]. He gave examples of simulations of English text, first with a random mixture of alphabetic symbols, then demonstrating increasing similarity to English as models of single, double and triple letter pattern probabilities were introduced and finally even greater similarity as single and double word pattern probabilities were used. He pointed out that multiple alphabetic symbol and multiple word approximations could be represented by Markov processes.

A Markov process describes the transitions between the states of a system in terms of the probabilities of the transitions. If the system is currently in state s_i, the probability that the next transition takes the system to state s_j is denoted p_{ji}. There is no residue of influence from preceding states. Shannon proposed that to turn a Markov process into a description of an information source required only that a letter was produced at each transition.

Two representations of Markov processes are common: a diagrammatic representation as shown in figure 5.7 and as a (square) matrix of transition probabilities, $\Phi = \{p_{ji}\}$.

There is a class of Markov processes referred to as ergodic. The Longman dictionary defines ergodic as: "... of a statistical system in which the fraction of time each subsystem spends in a given statistical state equals the fraction of subsystems in that state at any given time..." [49].

Shannon gives two properties of or tests for ergodic Markov processes, paraphrasing:

•It must be possible to get to any state from any other state (i.e. no breaks in the chain); and

•If the lengths of all the circuits from a particular state back to itself were measured the highest common factor of the set of circuit lengths would be unity (i.e. no periodicity).

Figure 5.7: An Ergodic Markov Process

An ergodic Markov process has a path from any state to any other and has no periodicity.

An everyday situation which may be approximated by an ergodic Markov process is a city-centre circular bus route. Assuming all stops are roughly equally busy and

ignoring start-up and shutdown it is possible to create an ergodic Markov process model. The state in question is the number of passengers on the bus which may vary between zero and the capacity of the bus and a state transition occurs every time the bus stops and passengers get on and off.

It should be noted that the bus is an example of the most general case of ergodic Markov process because the network is fully connected and transition from any state to any other is possible in a single step. The networks in the situations of interest to this work tend to be more sparsely connected. In a sparser network all transitions of an ergodic Markov process are still possible but some require several steps.

The important property of an ergodic Markov process from the point of view of this work is that the probability of selecting an object at random and encountering it in a particular state is the same as the probability of encountering a particular object in that state when the object is observed over many state transitions. In other words, sequences drawn from a single ergodic Markov process tend to have the same statistical properties and given a long enough sequence the probability of encountering each state, P_i, approaches equilibrium. Thus:

$$P_i = \sum_j P_j p_{ji}.$$

For an ergodic Markov process the equilibrium is approached irrespective of the starting conditions. We can approximate the equilibrium by iterating $p(n) = \Phi p(n-1)$, where n denotes iteration number, $p = [\cdots P_i \cdots]^T$ and $p(0)$ is any probability vector such that $\sum_i P(0)_i = 1$. Shannon calculated the entropy of the P_i's to represent the amount of information "produced" by the process. It is seen in Chapter 6 that we use an ergodic Markov process to represent the possible behaviours that may be encountered from a data fusion system.

Ekroot and Cover took a different view of the entropy of a Markov process that has applications in backgammon, gambling and population growth [23]. Ekroot and Cover considered the ideas of logical depth and thermodynamic depth. In plain language, they studied entropy measures of the process or path by which the current state of affairs arose. A Markov trajectory t_{ij} from s_i to s_j was defined as having initial state s_i, final state s_j and no intermediate visit to the final state, and Υ_{ij} was defined as the set of all such trajectories. The entropy of a trajectory was then defined as:

$$H_{ij} = \sum_{t_{ij} \in \Upsilon_{ij}} p(t_{ij}) \ln\left\{ \frac{1}{p(t_{ij})} \right\}.$$

This offers a possible avenue of future research into the mixtures of behaviours that arise from data fusion systems and may have applications in systems autonomy and automatic sensor management. Using Ekroot and Cover's Markov trajectory model it may be possible to examine the utility of possible actions or sequences of actions in terms of the probabilities of reaching desired and undesired information states as a result.

6 Simulations and Models

Numerous simulations and models are used in this work. A simulation is a partial implementation of a system, its human operators, the environment and the situation. Missing elements are replaced with abstractions; for instance: the environment may be synthetic. When a simulation is performed, the objective is to give the appearance of the true and complete system performing a real operation to fulfil some task. The task and its outcome are then observed as if they were real.

For this work, when a data fusion process is evaluated by simulation, it is the information "feed" of sensors and other sources that is simulated. The data fusion process may be present in its entirety or it might be represented by a simpler collection of key components without loss of fidelity. The simulations are discussed in section 6.1.

A model is an abstraction of the system, its operators, the environment and the situation. The aim of a model is to predict the outcome of the task without the operation actually taking place. When a model of a data fusion process is evaluated, it is the expected final information quality of the fused information which is calculated along with the probability that this information quality is sufficient for the task.

There are four types of model described in this chapter, all novel in some respect:
1)Data association modelling is concerned with the ability of the data fusion process to discern true clusters of observations in the source data, where a true cluster comprises observations of a single real-world object.
2)Estimation modelling is concerned with the effect of combining clustered observations.
3)Ignorance is modelled to quantify the penalty when information is missing.
4)Fusion state modelling is concerned with the sequence of decisions taking place within data fusion in response to random events and the stability of the likely resulting information.

A new Data Association model is discussed in section 6.2.1. The proposed model is novel because it operates in a space where distances are normalised with respect to the distribution of relative errors between new information and information already accumulated. The main advantages of this compared with known approaches are greater accuracy and applicability to a wider range of situations.

Estimation modelling is discussed in sections 6.2.2 (tracking), 6.2.3 (track fusion) and 6.2.4 (classification). Tracking is represented by the well known steady-state solution of the matrix Riccati equation. New formulations of the solution as a series expansion and a nested computation are offered. The effect of track fusion may be represented by the standard expression for the covariance matrix of the estimator in use. Using the above models for tracking and track fusion, the estimation error covariance matrix is used to calculate the information entropy. Alternatively, a new model is offered by which track fusion entropy may be propagated directly.

Two models are offered for classification. The first model is new and predicts the evolution of the entropy of the class belief distribution when class observations are combined using a recursive Bayesian process. The second model takes the obvious approach of summing the class decisions , weighted by their probabilities, for each possible outcome of the fusion process. The result is the predicted distribution of classification decisions for which an entropy may be calculated. Surprisingly, the two models were found to produce different results and this difference was borne out by the case studies reported in Chapter 7 where it is discussed in more detail.

A model of the information penalty of ignorance is offered in section 6.2.5. Ignorance may result from partial or missing information and an approach is proposed which represents it as an uncertainty distributed uniformly over the entire region of interest. That uncertainty then has a finite entropy.

The new concept of fusion states, and a model to predict their influence on fused information quality, is proposed in section 6.2.6. A significant problem in modelling the effectiveness of a situation awareness system arises because information quality can undergo step changes; for example, when a new information source starts to report or an established source ceases reporting. These must be accounted for and it is proposed in this thesis that the states of an ergodic Markov process be used to represent the resulting switching behaviour of the information error.

6.1 Simulations of the Data Fusion Process

In general, access to operational data fusion systems for research work is rare. Some of this work is evaluated by comparison with an operational system. However, the majority is evaluated by comparison with detailed Monte Carlo simulations. Simulations of stochastic systems are used to provide an information "input" or "feed" to the data fusion processes which are studied; thus a wider range of situations and system are studied.

Time-stepping simulations are used for this work. Simulations of this type are activated repeatedly and each activation is taken to represent a specific time interval. The interval can be of fixed duration or can be defined as the interval between key events. All the system or process activity in the interval is then simulated during the activation. On completion of the sequence of time-steps prescribed by the situation of interest, a single example of a possible behaviour of the system is obtained. In general, large numbers of repetitions are used in these simulations in order to measure the average system behaviour.

Figure 6.1 shows the generic process , comprising four stages, by which the simulations were carried out. Capture State is where the scenario data for the current time step is extracted from the Situation Database and made available to the subsequent processes; for example, in an air traffic control simulation the Situation Database could comprise a list of aircraft and their flight plans. At each time-step, parameters defining the latest position, velocity, orientation and status of each aircraft could be read. The Situation Database defines the "true" situation.
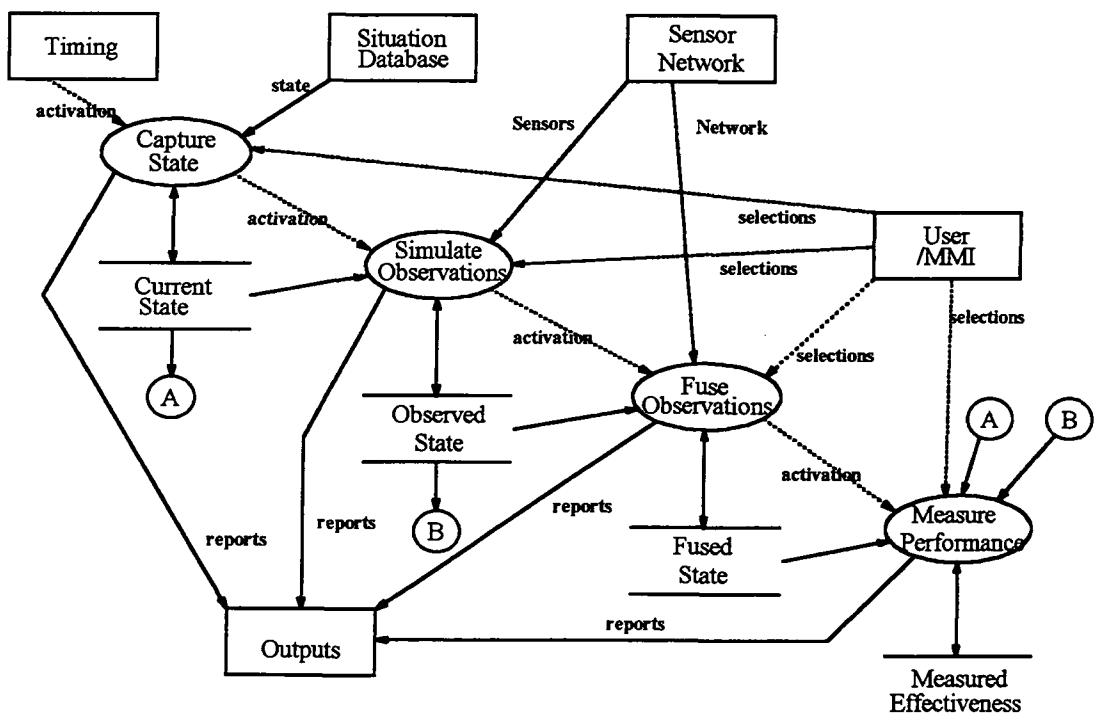
Figure 6.1: The Simulation Process

> The four stage simulation process flows diagonally from top left to bottom right. The process builds up a fused picture whose information quality can be measured.

Simulate Observations executes the sensor simulations. The scenario data for the current time is compared with a definition of the physical properties of the sensor observation process. In the air traffic control simulation, the statistical distributions of detection and parameter estimation for each aircraft are predicted in this way and a set of "observations", with appropriate statistics are created using Monte Carlo methods.

Fuse Observations executes the fusion processes. The simulated observations from the sensor simulations are processed by data fusion as if they are real sensor data. Finally, Measure Performance calculates the MoPs and records the results.

Capture State is highly scenario dependent and there is little which can be said about it in general terms. It is specific to the system of interest and the situation in which that system is being evaluated and these issues are dealt with in Chapter 7 where case studies are described. Measure Performance makes and stores the MoPs and MoEs discussed in Chapters 4 and 5. Results from specific experiments are also given in Chapter 7.

It is in the activities Simulate Observations and Fuse Observations where the behaviour and effect of the systems of interest are synthesised. In Simulate Observations the process of observation is represented and in Fuse Observations the actions of the system.

6.1.1 Simulation of Sensor Observations

We have seen that time-stepping sensor simulations are used to produce time-series of "observations" to drive the subsequent data fusion process. Figure 6.2 represents the stages in the sensor simulation. Figure 6.2 is a decomposition of the activity Simulate Observations of Figure 6.1.

The activities in this decomposition may be described generically as:

• Spatial Coverage, where the regions observed by the sensor in the current time-step are determined and the objects lying within them are extracted from the scenario.

• Detection, where the objects are considered in turn and a detection decision made for each one.

• Observation Error, where observations are created for detected objects by adding appropriate random errors to their true states.
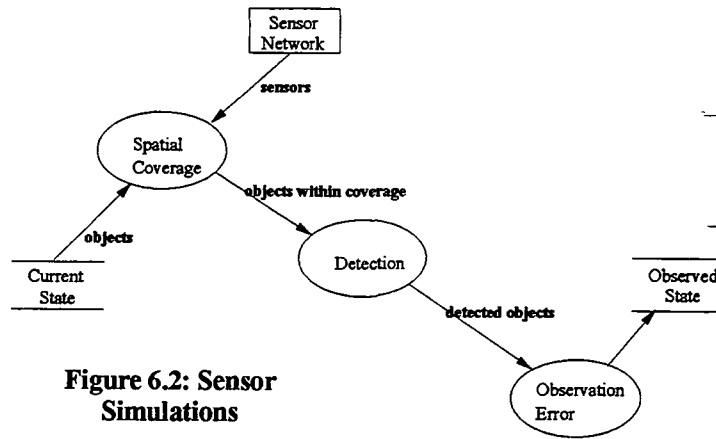
Figure 6.2: Sensor Simulations

The sensor simulation process flows diagonally from top left to bottom right. Starting with definitions of the scenario objects and sensors the process creates simulated sensor observations.

The detail of each stage depends on the sensor being modelled. The observed region is determined from a definition of the seeking behaviour of the sensor. For example, the simplest behaviour might be described as staring. A staring sensor observes a fixed "window" continuously. For such a sensor, the observed region would likewise be fixed and correspond to the window.

More complex behaviours, such as scanning, are common; where a window is swept over a larger region in a given pattern at some given rate. Modern sensors may have an agile scanning capability where the position of the window is not constrained by a set pattern and can be moved around the overall observed region in any selected progression of window positions.

As an example, consider the detection process of a Radar simulation. It calculates a signal to noise ratio for each object in question via the Radar equation [35] [77]:

$$\frac{S}{N} = \frac{\text{Transmitted power density} \times \text{Reflected power density} \times \text{Effective antenna receiving area}}{\text{Noise} \times \text{Loss}}$$

$$= \frac{P_t G}{4\pi r^2} \times \frac{\sigma}{4\pi r^2} \times A_e \times \frac{1}{N} \times \frac{1}{L}$$

$$= \frac{P_t G A_e \sigma}{(4\pi)^2 r^4 NL}$$

Where $P_t G A_e$ are constants describing the Radar's properties (transmitted power, antenna gain and effective antenna receiving area respectively), σ is the mean Radar cross section (RCS) which denotes the reflecting area presented by the object, N is the sum of the environmental and systematic noise, L is a factor to allow for propagation losses and r is the distance from the sensor to the object.

The expected signal to noise ratio is then used to make a random detection decision based on the probability of detection, Pd, for the object given by the

approximate relationship: $P_d \approx P_{fa}^{\frac{1}{\left(1+\frac{S}{N}\right)}}$.

Where P_{fa} is the probability of a false alarm (a spurious observation due to clutter or noise) corresponding to the Radar's detection threshold. Thus, a particular Radar observing a particular object in particular environmental conditions can be characterised by a graph of P_d versus object range. Such a graph is reproduced in figure 6.3.

Detected objects are turned into observations by the addition of random errors with the appropriate distributions. Finally, false alarms are generated at random with uniform distributions that corresponded to P_{fa}. Based on comprehensive sensor parameters, accurate sensor simulations may be created in this way.

Figure 6.3: Radar Model Detection Behaviour (vs. range)
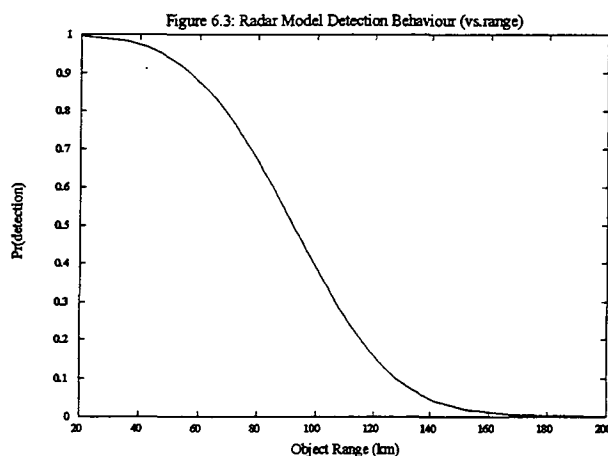
The curve of Radar Pd versus range for a given combination of Radar, observed object and environmental conditions.

Page 149

6.1.2 Data Fusion Architectures and Simulation Networks

Given accurate sensor simulations providing a source of observations, it is required to construct a simulated data fusion process. If an existing system is to be simulated, it is usual for a full and complete definition of the fusion process to be available. If a data fusion concept is to be evaluated through simulation, the concept itself usually provides the definition.

A simplistic, but adequate, approach to simulation would be to build the processing centres and data interconnections belonging to the system in their entirety and connect them to the sensor simulations. The drawback to such an approach is cost and lead-time. Given a system of interest expressed as a network of sensors, algorithms and interconnections various options exist for simplification. .

Given a particular task with a particular attribute space of required information, if there existed a sensor whose observation space matched the attribute space or was a superset of the attribute space, and if the performance of that sensor regularly and reliably exceeded the SoP for the task in question, there would be no need for a data fusion process. The information output of the sensor would simply feed the task and all requirements would be satisfied. Usually, this is not possible.

It might be the case that a sensor could observe only a subset of the attribute space; or that its performance consistently fell short of the SoP for the task; or that an ideal sensor was too expensive; or too big and immobile and not in the right place. In such circumstances, multiple sensors would be used.

Multiple sensors with the combined potential to provide all the information in the attribute space require a data fusion process to do the necessary combination. In its simplest form data fusion might be implemented as a single consolidation process receiving observations from all the sensors. More usually, if the sensors are distributed over a wide geographical region or some of them incorporate data processing provided

by the manufacturer, multiple consolidation processes are implemented with a network

of interconnections. Within this network would exist a hierarchy of processes. The

arrangement of sensors, processes and interconnections and the control hierarchy

imposed on them is commonly referred to as the data fusion architecture. Three basic

architectural forms are well known.

6.1.2.1 Fully Centralised Data Fusion Architectures

Data fusion systems with a fully centralised architecture gather all the sensor

observations in a single fusion process [8] [82]. Figure 6.4 shows a fully centralised

architecture and a corresponding simulation. Such an architecture is likely to provide

optimal fusion of the observations and would be represented simply in a simulation as a

single tracker and data association process receiving observations from all the sensor

simulations.

Figure 6.4: Fully Centralised Architecture
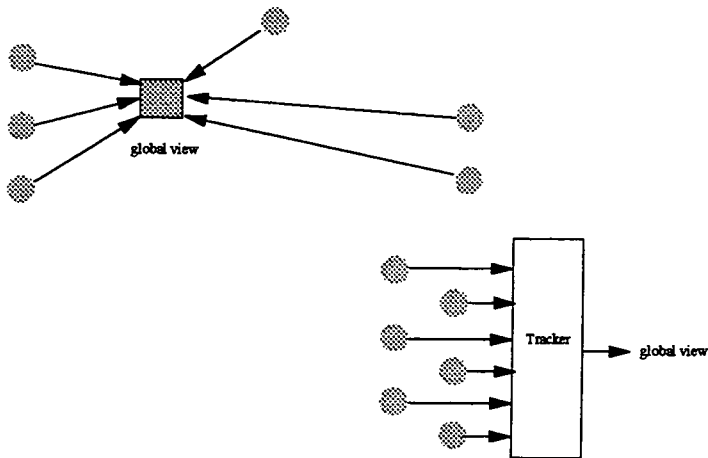
A Fully Centralised Data Fusion Architecture is represented in simulation as a single tracker
fed by multiple sensor simulations.

6.1.2.2 Autonomous, Distributed Data Fusion Architectures

The alternative to a fully centralised architecture is a distributed architecture.

Within a distributed architecture, the fusion process is divided into sub-processes in

multiple computers and part-processed data is exchanged between the sub-processes.

Within autonomous distributed architectures there would be sensor-level fusion

processes (for example, trackers) that would consolidate single sensor observations

independently of all the other processes and, in general, without reference to the data

produced by other sensor-level processes [8] [9] [72] [82]. Intermediate-level fusion

processes would then combine the sensor-level views to form a local view that would, in

turn, be communicated to the higher level and combined with other local views. This is

represented, alongside a corresponding model, in Figure 6.5.

Figure 6.5: Autonomous Distributed Architecture
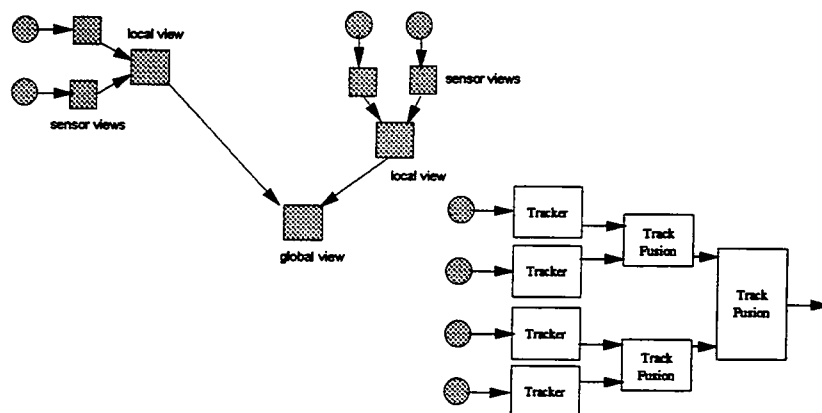
An Autonomous Distributed Data Fusion Architecture is represented in simulation as multiple trackers fed by sensor simulations. Tracks are consolidated by a hierarchy of Track Fusion processes.

The strengths of this architecture are that communications between processes

may be scaled to fit the capacity of the data-interconnections and data from highly

disparate information sources may be combined. The weakness of this approach is that

data is under exploited, since the sensor processes will not interpret their data as

accurately as they would if they had access to the global picture. In addition, the global views held in different parts of the system may diverge.

In a simulation, each sensor would have its own tracker and data association process. Several trackers would provide tracks to the intermediate-level Track Fusion and data association process. The data-interconnections need not be physical, rather they could be represented by restrictions on data exchange between software modules in the simulation.

Within this architectural model, it is possible for multiple Track Fusion Processes to provide consolidated tracks to still higher Track Fusion and data association processes in a scaleable multiple-layered hierarchy.

6.1.2.3 Integrated, Distributed Data Fusion Architectures

Data fusion systems with an integrated distributed architecture differ from those with autonomous architecture in that each sensor process has access to the global picture and produces updates to that picture [18] [19] [51] [66]. The global picture is distributed using algorithms such as the distributed Kalman filter (DKf) described in Chapter 3.

The global picture updates are communicated between the processes. Each sensor process constructs its own updated copy of the global picture and all copies are the same. This approach can operate with or without the presence of a dedicated process to maintain the global view. It can accommodate sensor-less platforms that construct their own copy of the global view from receive-only communications. It is shown in Figure 6.6 alongside a corresponding simulation. In a simulation, this type of architecture would be represented very simply by a single tracker and data association process identical to that of the fully centralised architecture.

The strengths of this approach are that it is inherently expandable, that it exploits fully the available situation awareness data and that the global views held by different platforms are in effect copies of the same thing. Each copy of the global view is the same

as would be produced by an optimal, fully centralised system. The weakness of this approach is that its demands for inter-platform communications are relatively high.

Figure 6.6: Integrated Distributed Fusion Network
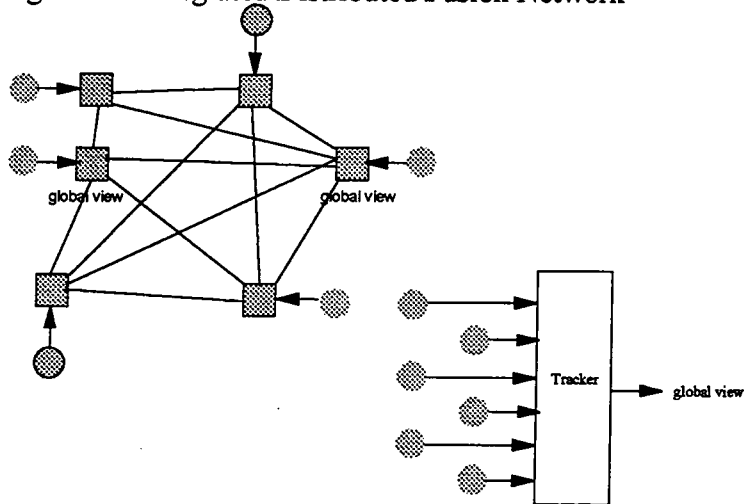
An Integrated Distributed Data Fusion Architecture is algebraically equivalent to a single tracker fed by multiple sensors and is represented in simulation as a single tracker.

6.1.2.4 Hybrid Data Fusion Architecture

The architecture in a system need not be homogeneous. Parts of a system could benefit greatly from an integrated, decentralised approach or a centralised approach; whilst the overheads this approach would impose in other parts of a system might be excessive. The option exists to adopt a pragmatic approach that varies the architectural model applied in different parts of the system. The result would be a hybrid architecture of the type represented in Figure 6.7. In a simulation, this type of system would be represented by a mixture of the forms described in previous sections [8] [82].

Once chosen, the architecture model is populated with sensor simulations as described in the preceding section and with data fusion algorithms from those described in Chapter 3, selected to best represent the system of interest. Details of each simulation

used in this work are given in Chapter 7 where the case studies carried out during this work are described.

Figure 6.7: Hybrid Fusion Architecture
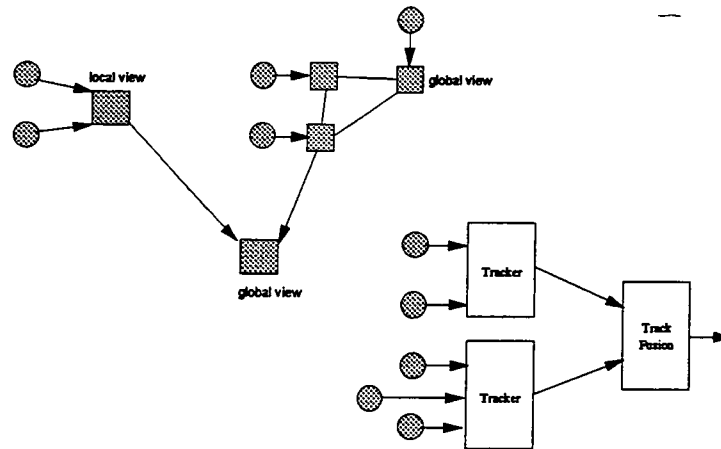
A Hybrid Data Fusion Architecture. The architecture combines centralised and distributed features and is represented in simulation by the most apt architectural forms.

6.2   Modelling the Data Fusion Process

Analytical models are proposed here for the information output of the various parts of the data fusion process as well as models to bind the parts together and predict the information output of complete systems. Detailed analyses were made of the processes that associate and consolidate information from sensor observations and representations of these processes are the bases of the models.

The alternative to analytical models are empirical models. Empirical models might be used if it were required to extrapolate the measured and recorded performance of a system in one situation in order to predict its performance in a different situation. Such models operate by postulating relationships between the parameters of the situations of interest and the system performance. For example, the Lanchester equations are used to model military engagements in terms of differential equations [84]. However, that is not what we are attempting here. We wish to evaluate system concepts and at the

concept definition stage no system exists. The approach adopted is represented in Figure 6.8.
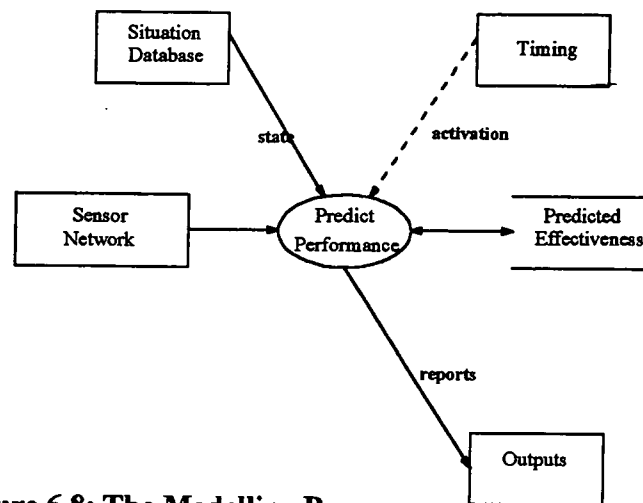
Figure 6.8: The Modelling Process

> The single stage modelling process predicts the information quality of the fused picture based on definitions of the situation and the sensor network.

Sensor models were used to represent the observation processes but unlike the simulations no time-series of observations was generated. Instead the observations were characterised by parameters such as revisit time, probability of detection and observation error distribution. The information processing within the system was represented by entropy models for common fusion processes.

Given reliable, accurate models of these processes, the performance of information systems was predicted in terms of entropy-based Measures of Performance. There were models for the types of data fusion algorithm: Data Association, Tracking, Track Fusion and Object Classification. In addition, models of special information states and system interactions were produced. They were: partial information, ignorance and Fusion States.

6.2.1 Data Association Models

Multiple objects have the potential to create Data Association ambiguity in the data fusion process and the ambiguity is a function of the object separation in relation to

the sensor resolution. It was seen in Chapter 3 that the situation of Figure 3.5 led to unambiguous Data Association and a single likely fusion operation when objects were well separated in relation to the resolution properties of the sensors. The similar situation of Figure 3.8 led to ambiguity and two different, likely fusion operations when objects were closer together in azimuth. The two fusion operations led to noticeably different ways of interpreting the data.

We saw also that there are three distinct ways in which Data Association algorithms deal with these ambiguities:

- by accepting the most likely interpretation of the data at each step as if it were true as in the Nearest Neighbour Standard Filter (NNSF);

- by deferring the decision as to the most likely interpretation until sufficient data becomes available, and meanwhile accepting the current most likely as interim, as in the Multiple Hypothesis Tracker (MHT); or

- by weighting the new observations by the probabilities that they are the true updates for each track as in the Joint Probabilistic Data Association Filter (JPDAF).

The first type of algorithm is considered in detail. The performance of the other algorithms has been written about by van Keuk on MHT performance and Kershaw and Evans on JPDAF performance [43] [80]. The question of whether the approach proposed in the following paragraphs can be extended to conveniently represent these other algorithms is left as a topic for future research.

6.2.1.1 Data Association Models: Background

A qualitative model of the effect of data association ambiguity on the data fusion process was proposed by Blackman [8]. Blackman gave a graph of object separation against sensor resolution cell size of the form of Figure 6.9.

There were two lines on the graph representing threshold levels of object separation at which the data association outcome changed. Above the upper threshold, Data Association was free of ambiguity and the probability of an allocation error was insignificant. Below the lower threshold, the object separation was so small that the extra numerical errors in the estimates due to allocation error were insignificant. When separation is very small it would be common for sensors to fail to separate multiple objects. In such circumstances, the group of objects might be observed as a single object. Between the thresholds, the outcome was uncertain. Our models will attempt to quantify these thresholds and uncertainties.

Figure 6.9: Data Association Outcomes [8]
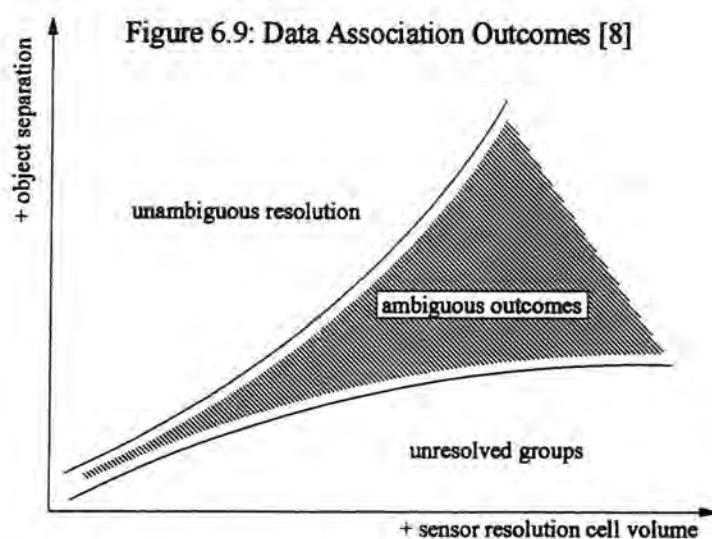
Blackman's conceptual model of Data Association outcomes. Above the shaded region, the resolution is unambiguous. Below the shaded region, ambiguity cannot be resolved. Inside the shaded region, the outcome is less predictable.

Suppose there is an object with m-dimensional state vector of continuous variables at location x in attribute space and another at $x+\delta$, with $\delta = \begin{bmatrix} \cdots & \delta_i & \cdots \end{bmatrix}^T$.

If a nearest neighbour standard filter (NNSF) algorithm is used to perform the allocation, it considers one of the object estimates first, say $\hat{x}$, which has zero-mean MVGD errors of covariance P. Both observations are considered as potential updates

for x and if one observation is allocated, there is one observation left for consideration as an update for the second object. The NNSF will be referred to as a sequential algorithm because the object estimates are considered in sequence.

If the new observations have zero-mean MVGD errors of covariance R, the normalised Mahalanobis distance of the true observation from $\hat{x}$ is:

$$z = (y-x)^T[P+R]^{-1}(y-x),$$

and this is used in the criterion for allocation [3]. Provided z is smaller than the normalised distance from the wrong observation,

$$w = (y-x-\delta)^T[P+R]^{-1}(y-x-\delta),$$

the correct data association will be made.

In Blackman's extended form of the NNSF, all the object - observation pairs are considered simultaneously [8]. In algorithms such as these, all track - observation normalised distance squares are calculated and the feasible set giving the smallest sum is selected. We will refer to this type of algorithm as simultaneous (also known as optimal allocation).

If we consider again the pair of objects above and denote the distances of the true observations from the object estimates as z_1 and z_2, and those of the false (interchanged) observations as w_1 and w_2, we allocate the observations in the correct pairing when $w' = w_1 + w_2 > z_1 + z_2 = z'$. We will assume that the estimate of the object at $x+\delta$ also has zero mean MVGD errors of covariance P.

Mori, Chang and Chong calculated the probability of correct association in a simultaneous data association algorithm for two targets with the sensor probability of detection of individual objects, $pd \approx 1$ [55].

$$\Pr_c = \Pr\{w' > z'\} = E\left[erf\left(\sqrt{\tfrac{\lambda}{2}}\right)\Big|x=0\right]$$

Where $erf\left(\sqrt{\frac{\lambda}{2}}\right) \triangleq \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\sqrt{\frac{\lambda}{2}}} \exp\left(-\frac{t^2}{2}\right) dt$ and $\lambda \triangleq \delta^T[P+R]\delta$. Mori et. al. found that

probability of correct association could be approximated by the expression:

$$\mathrm{Pr}_c \approx 1 - \left(\frac{\bar{\sigma}}{r}\right)^m \times 2^{m-1} \pi^{\frac{1}{2}} \Gamma\left(\frac{m+1}{2}\right) = 1 - C_m\left(\frac{\bar{\sigma}}{r}\right)^m$$

where r is the radius of an m-ball over which the expectation was calculated, centred on

one target and just enclosing the other, $\bar{\sigma} \triangleq |P+R|^{\frac{1}{2m}}$ and $\Gamma(p+1) \triangleq \int_0^{\infty} e^{-q} v^p dq$. When a

given N objects lay in the m-ball this became:

$$\mathrm{Pr}_c \approx \left[1 - C_m\left(\frac{\bar{\sigma}}{r}\right)^m\right]^{N-1},$$

and when N was Poisson distributed with mean u:

$$\mathrm{Pr}_c \approx \exp\left[-u C_m\left(\frac{\bar{\sigma}}{r}\right)^m\right],$$

When extraneous (clutter) objects were present and their number was Poisson

Distribution with mean v:

$$\mathrm{Pr}_c \approx \exp\left[-(u C_m + v D_m)\left(\frac{\bar{\sigma}}{r}\right)^m\right], \text{ where } D_m \triangleq 2^{\frac{m}{2}} \frac{\Gamma(m)}{\Gamma\left(\frac{m}{2}\right)}.$$

These models fell short of our needs for this work for three reasons:

- we were interested in situations where $pd \neq 1$;

- we required to examine specific scenarios and configurations of objects as well

  as generalised object and clutter densities;

- the approximation excluded small values of r; and

- initial experiments indicated only moderate accuracy.

6.2.1.2 A Model for the Probability of a Correct Association

The following new approximations for the probability of correct association for

an object with m-dimensional state vector of continuous variables are proposed. These

approximations are both simple and easy to apply to specific situations.

First, consider a sequential allocation process where the known objects are considered in turn and the "best" matching new observation is allocated to update the current object before the process moves on and considers the next object. Consider the pair of objects defined above about which ambiguity may arise.

Stuart and Ord give the distribution of the normalised distance, z, of the true observation from the track as the familiar χ_m^2 distribution [40]:

$$f_1(z) = \frac{1}{2^{\frac{1}{2}m}\Gamma\left(\frac{1}{2}m\right)} e^{-\frac{1}{2}z} z^{\frac{1}{2}m-1}$$

and that of the normalised distance, w, of the wrong observation from the track as the non-central χ_m^2 distribution:

$$f_2(w) = \frac{1}{2^{\frac{1}{2}m}\Gamma\left(\frac{1}{2}(m-1)\right)\Gamma\left(\frac{1}{2}\right)} e^{-\frac{1}{2}(w+\lambda)} w^{\frac{1}{2}m-1} \sum_{r=0}^{\infty} \frac{\lambda^r w^r}{(2r)!} B\left(\frac{1}{2}(m-1), \frac{1}{2}+r\right)$$

Where $\Gamma(\cdot)$ and λ are as defined above and $B(p,q) = \int_0^1 v^{p-1}(1-v)^{q-1}dv = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$.

Stuart and Ord give the characteristic functions of $f_1(z)$ and $f_2(w)$ respectively as [40]:

$$\phi_z(t) = (1-2it)^{-\frac{1}{2}m}, \text{ and:}$$

$$\phi_w(t) = (1-2it)^{-\frac{1}{2}m} \exp\left\{\frac{\lambda it}{1-2it}\right\}.$$

Thus, assuming w and z to be independent, the distribution of $(w - z)$ has the characteristic function:

$$\phi_{w-z}(t) = \exp\left\{\frac{i\lambda t - \frac{4\lambda t^2}{2}}{1+4t^2}\right\},$$

which indicates:

$$\Pr_c = \Pr(w > z) = \text{erf}\left(\frac{\sqrt{\lambda}}{2}\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{\sqrt{\lambda}}{2}} \exp\left(-\frac{t^2}{2}\right)dt.$$

This can be obtained from a standard table. When we extend this to represent a simultaneous data association algorithm we find:

$$\phi_{w'-z'}(t) = \exp\left\{\frac{2i\lambda t - \frac{8\lambda t^2}{2}}{1+4t^2}\right\},$$

which indicates:

$$\text{Pr}_c = \text{Pr}(w' > z') = \text{erf}\left(\sqrt{\frac{\lambda}{2}}\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\sqrt{\frac{\lambda}{2}}} \exp\left(-\frac{t^2}{2}\right)dt.$$

We note that this is the expression for which Mori et. al. calculated the expectation [55]. If we work only in space normalised with respect to $[P+R]^{-\frac{1}{2}}$ we find that we can use these models directly without calculating expectations and this leads to greater simplicity. In this normalised space, the probability of correct association for a given λ is constant over an m-ball and equal to Pr_c.

For specific scenarios and object constellations, this approach is highly convenient and accurate. Such scenarios can usually be transformed to normalised space with greater accuracy than multivariate error distributions can be transformed to natural space.

Figure 6.10 gives the graph of probability of false allocation, $\text{Pr}_f = 1 - \text{Pr}_c$, versus λ, produced by the above models, for sequential and simultaneous data association algorithms.

An immediate and striking feature of these graphs is the performance advantage of the simultaneous algorithm over the sequential one. If this were borne out by experimental results, there could be little justification for ever using the latter algorithm and little need for such a model in this work. We will see in Chapter 7 that it is borne out. However, we will also see later that when $pd \neq 1$ the simultaneous algorithm can behave like the sequential algorithm and this model will be required to provide a complete representation of its performance.

The models have an unexpected feature in their independence of the dimension of the object state vector of the system of interest. Unlike the models proposed by Mori et. al. these models depend only on the scalar λ and not on m [55].

Figure 6.10: Pr(False Allocation) for Two Objects

Object Separation as a Normalised Distance Square (λ)
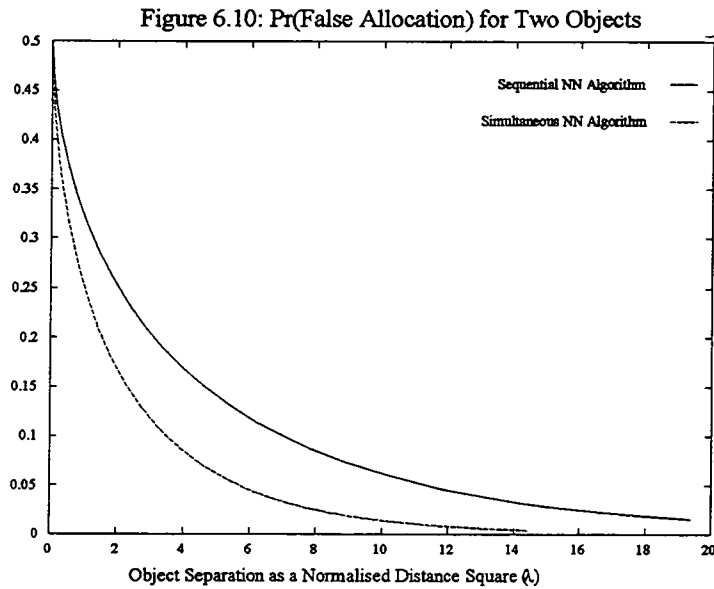
Probabilities of false allocation predicted by the models. The superior performance of the simultaneous algorithm is clearly visible.

When the ambiguous group comprises more than two objects the models may be extended using the argument put forward by Mori et. al. Namely, given N objects in an ambiguous group, object x is correctly associated by the event in which there is no transposition of observations involving that object. Assuming transposition events are independent the probability of correct association is approximated by [55]:

$$\Pr_C = \prod_{\forall w'} \Pr(w' > z').$$

Where w' is the distance separating the estimated object from one of one of the possible false observations and z' is the corresponding separation for the true observation. For our model this gives: $\Pr_C = [\text{erf}(a)]^{N-1}$,

where $a = \begin{cases} \dfrac{\sqrt{\lambda}}{2} & \text{for a sequential algorithm,} \\[2ex] \sqrt{\dfrac{\lambda}{2}} & \text{for a simultaneous algorithm.} \end{cases}$

Page 163

Furthermore, we can adopt the argument by which Mori et. al. arrived at their approximation for the probability of correct allocation when N is Poisson distributed with mean u, namely:

$$\Pr_C = \sum_{N=0}^{\infty} (\Pr_C |N) \Pr(N)$$

$$\approx e^{-u} \left\{ 1 + \sum_{N=1}^{\infty} [1 - \mathrm{erf}(-a)]^{N-1} \frac{u^N}{N!} \right\} \text{ noting that } \mathrm{erf}(a) = 1 - \mathrm{erf}(-a)$$

$$= \frac{e^{-u\,\mathrm{erf}(-a)} - \mathrm{erf}(-a) e^{-u}}{1 - \mathrm{erf}(-a)}$$

$$\approx \exp[-u\,\mathrm{erf}(-a)] \tag{6.1.2.1}$$

When clutter is present and is Poisson distributed with mean v, this becomes:

$$\Pr_C \approx \exp[-(u + v)\,\mathrm{erf}(-a)] \tag{6.1.2.2}$$

Independence of the errors in z and w is assumed in the preceding paragraphs in the calculation of the characteristic function of the distribution of the decision metric $(z - w)$ for a pair of objects. In general, errors in z and w are correlated. However, the numerical consequences of this are small and these models of correct association probability were found to produce acceptable results in general.

6.2.1.3   A Model for the Effects on Estimation of Data Association Error

Based on the above models for the probability of correct association, we shall predict the effects of false association on estimation accuracy. First, consider the case when established tracks exist and the probability of detection for the individual objects, $pd \approx 1$.

Given the pair of objects from the discussion above, we see that the estimate for the object at x is created from a mixture of true observations and false observations (from the interfering object at $x + \delta$). The proportions in the mixture are $\Pr_C$ and $1 - \Pr_C$ respectively. This mixture of data has a mean and variance that differ from those of the

true observations belonging to x and the differences will affect the performance of the data fusion system.

If we consider the way in which association errors arise, we recall that transpositions of object - observation pairs occur when the normalised sum of squares of the relative errors between observation and estimate in the transposed allocation $(\overline{w}')$ is less than the normalised sum of squares of the relative errors in the true one (z').

First, consider the case where observation errors are much larger than estimation errors $(P \ll R)$; a model of the behaviour of the relative error between observation and estimate will provide a good approximation for the apparent observation error in the set of allocated observations. We will model the estimated position of the track of x in normalised space as the mean of the allocated set of observations. Define:

$$\mu_t \triangleq E[t \mid y \in y(x)] \text{ and } \sigma_t^2 \triangleq E[(t - \mu_t)^2 \mid y \in y(x)].$$

where $y(x)$ denotes the set of observations allocated to x. Then:

$$\mu_t \approx \frac{1}{\sqrt{2\pi}} \left[\int_{-\infty}^{a} t \exp\left(-\frac{t^2}{2}\right) dt + \int_{-\infty}^{-a} (t + 2a) \exp\left(-\frac{t^2}{2}\right) dt \right]$$

The first term represents the true observations' contribution to the estimate and the second term represents the falsely associated observations' contribution. For a sequential algorithm the bias $b \approx \mu_t$ and for a simultaneous algorithm $b \approx \frac{\mu_t}{\sqrt{2}}$. In attribute space b is related to the bias vector $\beta_{P \ll R}$ such that:

$$b = -\sqrt{\beta^T [P + R]^{-1} \beta}$$

If we adopt the convention that the object separation $\sqrt{\lambda}$ is in the positive direction in normalised space, we note that b is negative. This is interpreted as β operating in the direction directly away from the interfering object.

The variance σ_t^2 is approximated by:

$$\sigma_t^2 \approx \frac{1}{\sqrt{2\pi}} \left[\int_{-\infty}^{a} (t - b)^2 \exp\left(-\frac{t^2}{2}\right) dt + \int_{-\infty}^{-a} (t + 2a - b)^2 \exp\left(-\frac{t^2}{2}\right) dt \right].$$

For a sequential algorithm the apparent observation standard deviation $\sigma' \approx \sqrt{\sigma_t^2}$ and for

a simultaneous algorithm $\sigma' \approx \sqrt{\frac{\sigma_t^2 + 1}{2}}$.

In normalised space, a (one-sigma) region of constant likelihood around an object

estimate is an m-ball of unit radius. We find that $\sigma' < 1$ and this may be interpreted as a

flattening of this region in the direction of the line joining x to $x + \delta$, to make an

approximate m-ellipsoid. The apparent error standard deviation along that line is less

than that of the true data. In all planes perpendicular to that line, the projection of the

region remains a circle of unit radius.

The apparent observation error covariance $\bar{R}_{P \ll R}$ is obtained by:

$$\bar{R}_{P \ll R} = \psi R \psi^T$$

where ψ is a rotation of the matrix $diag\begin{bmatrix} \sigma' & 1 & \cdots & 1 \end{bmatrix}$ such that the scaling factor, σ',

applies along the line joining the two objects.

Figure 6.11 (a) represents these effects as they would be experienced in a system

with a 2-D object state using the geometric representation of uncertainty [51]. The

representation is in normalised space. The apparent position of $\hat{x}$ is $x + \beta$, and the

apparent distribution of observation errors in the data used to create $\hat{x}$ is approximated

by the shaded region. The distribution of the true observation errors is indicated by the

circle.

Both these effects are the opposite of what we would expect if a random mixture

of data had occurred. For the above data association errors, the tail of the true

distribution in the direction of the interfering object is lost to that object's estimate and

vice-versa. The gained false data are closer to x than the lost true data. That is why the

apparent observation error standard deviation decreases. These effects occur for both the

sequential and simultaneous algorithms. The difference between the algorithms is the

magnitude parameter, α.

Figure 6.11(a): The Effect of Data Assciation Errors (pd~1 and R>>P)
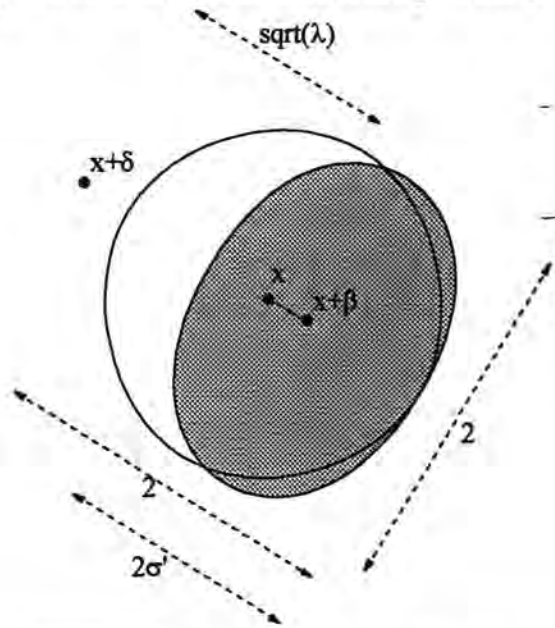
The effects of data association errors when Pd~1 and $P \ll R$. The observations allocated to x are biased away from the interfering object and their standard deviation is less than the true data.

When the opposite is true and estimation errors are much larger than observation errors ($P \gg R$), the allocation of observations to estimates is dominated by the estimation error and from the point of view of the observation, is completely random. Thus, the allocated set of observations take on the statistics of a random mixture. Such a mixture has a variance larger than that of the true observations and a bias towards the interfering object. Stuart and Ord give the expression for the mean of a mixture of scalar random variables with means μ_i as $\mu = \Sigma p_i \mu_i$ [40]. For our example this gives:

$$\bar{y} = x + (1 - \mathrm{Pr}_c)\delta \Rightarrow \beta_{P \gg R} = (1 - \mathrm{Pr}_c)\delta.$$

Stuart and Ord give the expression for the variance of a mixture of scalar random variables as $\sigma^2 = \sum_i p_i \sigma_i^2 + \sum_i p_i (\mu_i - \mu)^2$ [40]. For our example this gives the covariance:

$$\bar{R}_{P \gg R} = R + (\mathrm{Pr}_c - \mathrm{Pr}_c^2)[\delta\delta^T].$$

These effects are represented in figure 6.11(b). The mixture of distributions of the allocated observations can be bi-modal but, using the geometric representation of

uncertainty, it is represented in figure 6.11(b) as a single ellipse corresponding to the uni-modal distribution with the same mean and covariance [51].

Figure 6.11(b): The Effect of Data Assciation Errors (pd~1 and **R<<P**)⁻
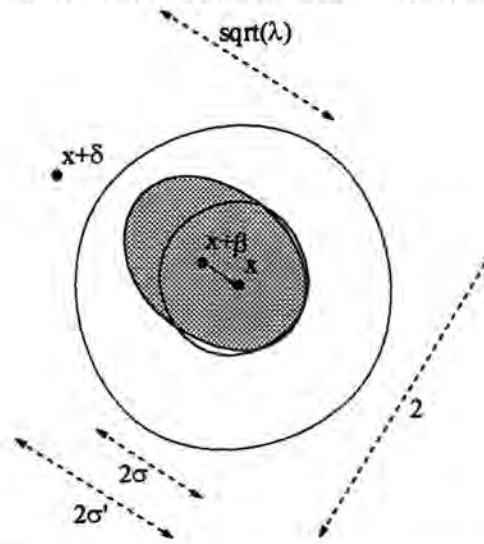
When *Pd*~1 and *R<<P*, the observations allocated to x are now biased towards the interfering object and their standard deviation, σ', is greater than that of the true data, σ.

These models correspond to the extreme cases in the range of possible ratios of observation to estimation errors. A linear combination of these models is used to represent intermediate values with coefficients $\frac{|R|}{|P+R|}$ and $1 - \frac{|R|}{|P+R|}$ respectively.

When the ambiguous group comprises more than two objects, the model may be extended. Let p_{ij} denote the probability of falsely allocating observations between the pair of objects $\{ij\}$, and let β_{ij} denote the bias vector in the estimate of $\hat{x}_i$ resulting from the error induced by the presence of object j. If there are n objects in the ambiguous group the bias, due to all such errors, in the mixture of observations used to create $\hat{x}_i$ will be represented:

$$\beta_i = c \sum_j^n p_{ij} \beta_{ij}$$

with p_{ii} denoting the probability of correct association calculated using equation (6.1.2.1) or (6.1.2.2), c is a normalising factor and $\beta_{ii} = 0$. This model can accept a specific configuration of objects with a different λ_{ij} for each pair, or it can be applied to

a generalised situation where the arrangement of objects is defined in terms of its density. The covariance of the mixture will be represented as:

$$\bar{R}_i = c \sum_{j}^{n} p_{ij} R$$

6.2.1.4 More Complex Data Association Behaviours

In the preceding sections, the assumption of $pd \approx 1$ has a simplifying effect. If established tracks of the objects do not already exist and the probability of detection $pd \neq 1$, more complex behaviours are possible. Consider again the pair of objects from the above example. If the probability of observing both objects in a single sensor visit is small and if the objects are sufficiently close together, a single estimate can form from the pooled observations of both objects. This is represented in Figure 6.11(c).

Figure 6.11(c): The Effect of Data Assciation Errors (pd << 1)
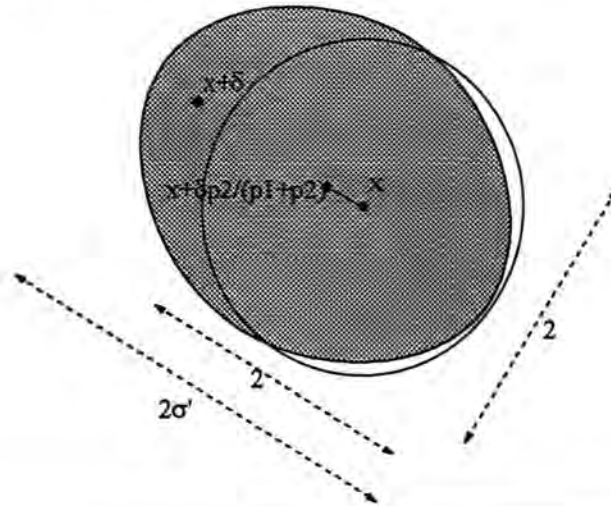
The effects of data association errors when $Pd<<1$. The observations from the objects at x and x+δ are pooled to form a single estimate.

Once again, the statistics of the random mixture take over as the best representation of the system behaviour irrespective of the relationship between observation and estimation errors. If the probabilities that the objects at x and $x + \delta$ are observed during any sensor visit are pd_1 and pd_2, the mean of the mixture of observations that go to form $\hat{x}$, $\bar{y}$ is approximated by the mean of a random mixture.

$$\bar{y} = x + \frac{pd_2}{(pd_1 + pd_2)}\delta,$$

and the covariance of the mixture is approximated by the covariance of a random mixture.

$$\bar{R} = R + \frac{pd_1 pd_2}{(pd_1 + pd_2)^2}[\delta\delta^T].$$

In general, any of the behaviours (the simultaneous model, the sequential model and the pooled observations model) might be observed with switching between them that would obey a Markov process. The component states might be:

- when one track is established and one object observed and allocated to the track, the pooled observations behaviour would be most likely;

- when two tracks are established but usually only one object observed, the sequential algorithm behaviour would be more likely;

- when two tracks are established and two objects usually observed, the simultaneous algorithm behaviour would dominate;

One of the other possible track to observation combinations would increase the probability of a behaviour switch. When none of the above states is dominant, instability can give rise to discontinuous, short-lived estimates.

Issues surrounding mixtures of behaviours have implications over and above their effects on data association errors. These issues are discussed in section 6.2.6 where the idea of Fusion States is introduced. In that section, ways of quantifying these mixtures of behaviours will be proposed.

6.2.2 Models of Tracking

We will model an object tracking process in terms of its information output.

6.2.2.1 Models of Tracking: Background

An object tracking process is applied to a time series of discrete sensor observations, $Y = \{y_1, y_2, ..., y_n\}$. Its purpose is to estimate the state of the variable x, given Y. We will predict the entropy of the distribution:

$$p(x|Y) = \frac{p(Y|x)p(x)}{p(Y)} \qquad (6.2.2.1)$$

To create a model, we could attempt to solve equation (6.2.2.1) and calculate the entropy of the solution, we could take the entropy of equation (6.2.2.1):

$$H[p(x|Y)] = H[p(Y|x)] + H[p(x)] - H[p(Y)] \qquad (6.2.2.2)$$

and attempt to solve for $H[p(x|Y)]$ or for a particular estimator, we could predict the parameters of $p(x|Y)$ and calculate the entropy on that basis.

We recall from Chapter 3 that Kendall, Stuart & Ord cite Kalman and Harrison & Stevens as the source of the computationally convenient, recursive formulation to the complete ARMA model of equation (3.2.1.1) [32] [38] [42]. The model is defined in two stages and comprises an observation equation and a system equation respectively for the kth observation:

$$y_k = h_k x_k + \varepsilon_k \text{ and } x_k = \Phi x_{t-1} + v_k.$$

Where h and Φ are known and ε, v are white zero mean errors with covariance R and Q respectively.

Kendall, Stuart and Ord show that, for scalar x, y and $h = \Phi = 1$, the estimation error variance after the nth update is given by:

$$P_n = \frac{(P_{n-1}+Q) \times R}{P_{n-1}+Q+R} \qquad (6.2.2.3)$$

and the variance, P, of the converged estimate, $\hat{x}$, for large n, is given by the limiting value:

$$P = \tfrac{1}{2}Q\left[\left(1 + \tfrac{4R}{Q}\right)^{\tfrac{1}{2}} - 1\right]$$

The limiting value corresponds to the information output of a converged tracking process [42].

For vector x and y, combining equations (3.2.1.2 b), (c) and (e) and using the notation P_k for $P(k,k)$, equation (6.2.2.3) becomes:

$$P_n = \left(I - (\Phi P_{n-1}\Phi^T + Q)H^T(H(\Phi P_{n-1}\Phi^T + Q)H^T + R)^{-1}H\right)(\Phi P_{n-1}\Phi^T + Q)$$

(6.2.2.4)

which is not readily solvable by analytical means.

6.2.2.2 Analysis of Covariance

Blackman proposed solving equation (6.2.2.4) numerically, by iteration, to model the propagation of estimation error covariance within a Kalman filter (Kf) [8]. This approach is referred to as Analysis of Covariance. Bar-Shalom and Fortmann refer to an equation, algebraically equivalent to (6.2.2.4), as the matrix Riccati equation and state the conditions under which P converges for large n [2]. The following simplification to Blackman's approach is proposed: let

$$A_{n-1} = \left(I - (\Phi P_{n-1}\Phi^T + Q)H^T(H(\Phi P_{n-1}\Phi^T + Q)H^T + R)^{-1}H\right)$$

then

$$P_n = A_{n-1}(\Phi P_{n-1}\Phi^T + Q)$$

and if the initial value of P was P_0, after n iterations we have:

$$P_n = (A_{n-1}\Phi \ldots A_0\Phi)P_0(\Phi^T)^n + A_{n-1}\sum_{i=0}^{n-2}(\Phi A_{n-2}\ldots \Phi A_i)Q(\Phi^T)^{n-2-i} + A_{n-1}Q$$

For large n, in a stochastic system, $(A_{n-1}\Phi \ldots A_0\Phi)P_0(\Phi^T)^n \approx 0$. Referring to the ARMA model of equation (3.2.1.1), this corresponds to $n > p$. We have the approximate solution:

$$P_n \approx A_{n-1} \sum_{i=0}^{n-2} (\Phi A_{n-2}...\Phi A_i)Q(\Phi^T)^{n-2-i} + A_{n-1}Q \qquad (6.2.2.5)$$

Thus, we can predict the estimation error entropy of the converged-Kf linear estimator. Our solution is a function of H, R, Φ and Q and differs from Blackman's approach because it requires no initial covariance matrix P_0. We note also that equation (6.2.2.5) may be written in the form:

$$P_n \approx A_{n-1}(Q + \Phi A_{n-2}(...(Q + \Phi A_1(Q + \Phi A_0 Q \Phi^T)\Phi^T)...)\Phi$$

which has a nested structure, highly convenient for computation.

We have assumed a minimum mean-square estimator (MMSE) solution to the dynamic linear model (DLM) time-series and calculated the covariance matrix of the errors in $p(x|Y)$ under that assumption. If we further assume multivariate Gaussian distributed (MVGD) errors, we recall from Chapter 5 that the entropy is a function of this covariance.

The above argument may be applied to a non-random system as a special case with $Q = 0$. The solution becomes:

$$P_n = (\mathring{A}_{n-1}\Phi...\mathring{A}_0\Phi)P_0(\Phi^T)^n \qquad (6.2.2.6)$$

where:

$$\mathring{A}_{n-1} = (I - (\Phi P_{n-1}\Phi^T)H^T(H(\Phi P_{n-1}\Phi^T)H^T + R)^{-1}H)$$

Equation (6.2.2.6) is also amenable to nested computation.

6.2.2.3 Convergence Issues for Tracker Models

A more elegant approach to predicting the converged covariance of an estimator in a non-random system is given by the Cramer-Rao Lower Bound (CRLB) [2] [51] [67]. In the estimation of a non-random parameter with an unbiased estimator, the covariance is bounded below:

$$P \geq J^-$$

where J is known as the Fisher information matrix and is given by:

$$J = -E[\nabla_x \nabla_x^T \ln\{p(Y|x)\}]$$

Provided that the estimator used is unbiased and efficient, such that it produces errors that attain the CRLB, this approach offers a means of predicting the converged performance. MVGD errors have this property [2] [59].

When the behaviours of the object of interest and the observations coming from the sensor obey linear laws, we are justified in taking the approach of equations (6.2.2.5) and (6.2.2.6). Furthermore, when the process noise (errors generated by the random variations in the object's state) and the observation errors are MVGD, this is a model of the optimal treatment of the incoming information since, for Gaussian distributed errors, the MMSE produced by the Kf coincides with the maximum a posteriori likelihood (MAP) [2]. As such, provided we may assume convergence to a stable error state, this approach is promising.

The error state convergence behaviour of a Kf depends on the ratio of measurement error to process noise errors. If the measurement errors are large and the individual observations contain only coarse information regarding the object of interest but the process noise is small so that random variations in the object behaviour are minor and rare, the Kf will be slow to converge and the approach of equation (6.2.2.5) would be inappropriate. In such circumstances, approximation by a non-random system would suffice and equation (6.2.2.6) would be used as the model.

Applications where the opposite is true are common. In such applications accurate observations are made of highly dynamic situations. Under these circumstances, error state convergence of the tracker is rapid and the model of equation (6.2.2.5) provides a good approximation to the information output after relatively few terms have been calculated. What is more, the number of observations required for convergence of the tracker and the number of terms for convergence of the nested computation of the

model are approximately equal. So the convergence behaviour of the model is an indicator of its appropriateness.

6.2.2.4 Tracker Models in Non-linear and Non-Gaussian Systems

When the observation process or the object dynamic behaviour are non-linear, modelling becomes more difficult. The non-linear observation and system models are of the form:

$$y_k = h(t_k, x_k, u_k, \varepsilon_k) \text{ and } x_k = \Phi(t_k, x_{k-1}, v_k) \text{ respectively,}$$

where u denotes the control input .

The extended Kalman filter (EKf) algorithm takes a linear approximation to a non-linear system at each observation. The model of tracker estimation error given in equation (6.2.2.5) provides an adequate approximation to the estimation error covariance from an EKf when the non-linearities are weak. Such circumstances are exactly those under which the EKf is likely to perform adequately as a tracker.

For highly non-linear systems or non-Gaussian systems, a different approach is required. Bar-Shalom and Fortmann give the recursive, discrete-time relationship linking the probability density function of the estimate at time k and $k+1$ [2]:

$$p(x_{k+1}|Y^{k+1}, U^k) = \tfrac{1}{c} p(y_{k+1}|x_{k+1}) \int p(x_{k+1}|x_k, u_k) p(x_k|Y^k, U^{k-1}) dx_k$$

where Y^k denotes the observations up to and including step k, U^k denotes the controls over the same interval, $\tfrac{1}{c}$ denotes a normalising constant and u_k denotes the control at step k.

When this equation may be solved numerically, the error estimation process might be executed in isolation from state estimation and used as the basis for a calculation of the entropy of an assumed Gaussian or Gaussian mixture error model. Similar to the approach applied to obtain (6.2.2.5) but this time applied to a direct (numerical) solution of equation (6.2.2.1). The detailed implementation of such models will be left as topics for future research.

Advances in algorithm formulation are limited by the analytic tractability of the system of interest. Some advances have been achieved in distributed tracking algorithms by algebraic manipulation of the Kf equations [51] [54] [66]. Whilst these algorithms offer new and powerful properties in applications that require distributed sensors and processing resources, if implemented optimally, the algorithms are numerically identical to the standard linear Kf and the models of (6.2.2.5) continue to apply.

6.2.3 Track Fusion Models

It is required to model the performance of a Track Fusion process in terms of its information output. We recall from Chapter 3 that Track Fusion is applied to a collection of track state estimates, $\{\hat{x}\} = \{\hat{x}^1, \hat{x}^2, ...\}$. Its purpose is to estimate the state of the variable x, given $\{\hat{x}\}$. We wish to predict the entropy of the distribution:

$$p(x|\{\hat{x}\}) = \frac{p(\{\hat{x}\}|x)p(x)}{p(\{\hat{x}\})}$$

(6.2.3.1)

The approaches available to solve equation (6.2.3.1) are the same as in the previous section. Namely: we could solve the equation directly, we could take the entropies and then solve or we could estimate the parameters of the solution. We will examine the process in more detail.

Track fusion is used when two or more object tracking processes, serving different sensors or sensor groups, operate simultaneously observing the same objects. If it is required to combine their outputs, a weighted combination of the object tracks will be produced. Consider again the simple example of Chapter 3:

Two sensor groups, i and j, produce estimates of the form $\hat{x}^i = x + \varepsilon_{x^i}$ and $\hat{x}^j = x + \varepsilon_{x^j}$ respectively. Assuming the errors ε have covariances P^i and P^j respectively, and are independent so that $E\left[\varepsilon_{x^i}\varepsilon_{x^j}^T\right] = P^{ij} = 0$, if a Minimum Mean Square estimator (MMSE) for x is used [42]:

$$P = P^i[P^i + P^j]^{-1}P^j$$

and using the expression for the entropy of MVGD errors from Chapter 5:

$$H = \tfrac{1}{2}[\ln\{|P^i|\} + \ln\{|P^j|\} - \ln\{|P^i + P^j|\} + m\ln\{2\pi\} + m].$$

This may be written as:

$$H = H_{x^i} + H_{x^j} - H_{x^i - x^j} \tag{6.2.3.2}$$

where $H_{x^i - x^j}$ is the entropy of the distribution of relative errors $p(x^i - x^j)$. Equation

(6.2.3.2) is used as a model of the Track Fusion performance of the system when the

errors introduced by assuming $P^{ij} \approx 0$ are insignificant numerically.

We recall from Chapter 3 that the problem with the above approach is the

assumption of independence. In a stochastic system, both sensors would be subject to the

same process noise (manoeuvre) errors. So the errors in two tracks of a single object

created from two, truly independent, observation sequences must nevertheless be

correlated.

We saw that Bar-Shalom and Fortmann considered the case where

$E\left[\varepsilon_{x^i}\varepsilon_{x^j}^T\right] = P^{ij} \neq 0$. We recall also that, in practise, P^{ij} would usually not be known and

if the errors introduced by ignorance of P^{ij} were likely to be significant, approaches

were available capable of producing conservative estimates. Conservative is defined as

$P - \bar{P} \geq 0$, and $\bar{P}$ is the true error covariance of the final estimate; in other words that the

predicted error distribution encloses the true error distribution of the estimate.

Each such approach would require an information error model appropriate to the

consolidation operations it performed. For example, if Covariance Intersection were

employed, P would be given by the relationship [14]:

$$P = \left[\omega(P^i)^{-1} + (1-\omega)(P^j)^{-1}\right]^{-1} \tag{6.2.3.3}$$

where $\omega < 1$ is a weight calculated to optimise a chosen norm. Equations (6.2.3.2) and (6.2.3.3) are the lower and upper bounds respectively on the entropy of optimal fused tracks. The former assumes that the errors are known to be independent and the latter assumes that the errors are known to be fully correlated. Both are optimal under their respective assumptions.

6.2.4 Object Classification Models

In this section we will consider the performance of object class estimation algorithms. We recall from out discussion of Chapters 3 and 5 that class is discrete, non-numeric and non-ordered.

When an object is observed by two or more sensors, or repeatedly by a single sensor, multiple statements regarding the class of the object may be available. If this occurs data fusion will produce a joint class statement. It is assumed that the performance associated with the individual sensors or observations is known, either from specification or empirically, and is available through the matrices C of equation (5.3.0.2). It is required to model the performance of the fused object class.

Given repeated, independent object class statements from a sensor that obey the known error distribution C, we might estimate the entropy of the distribution of object class after each new observation. Recalling equation (6.2.2.2):

$$H[p(x|Y)] = H[p(Y|x)] + H[p(x)] - H[p(Y)]$$

We may rewrite this as the entropy of the recursive form of the Bayesian classifier for the kth observation [51]:

$$H[p(x|Y^k)] = H[p(y_k|x)] + H[p(x|Y^{k-1})] - H[p(y_k|Y^{k-1})] \qquad (6.2.4.1)$$

where Y^k is the set of the first k observations and y_k is the kth observation. We propose to base a model of object classification on this recursive relationship. Two of the terms are easily obtained:

• $H[p(x|Y^{k-1})]$ is simply the result of the previous iteration,

• $H[p(y_k|x)]$ is enumerated from the appropriate row of the C matrix.

The third term is more difficult. We require to estimate the distribution of $p(y_k|Y^{k-1})$ in order to calculate its entropy. We know that:

$$p(y_k|Y^{k-1}) = \sum_{\bar{x}} p(y_k|x)p(x|Y^{k-1})$$

(6.2.4.2)

but the distribution of $p(x|Y^{k-1})$ is not known, only its entropy. Recalling equation (5.3.2.1) and replacing $1 - \pi$ with c_{max}, we propose to solve, numerically:

$$H_{max} = -c_{max}\ln[c_{max}] - (1 - c_{max})\ln\left[\frac{(1-c_{max})}{q-1}\right]$$

to calculate the maximum entropy distribution (as per Figure 5.1) with the entropy $H[p(x|Y^{k-1})]$. This maximum entropy distribution might be used to estimate $p(x|Y^{k-1})$ in equation (6.2.4.2).

It is worth noting that, in this case, the approach is **not** that of iterating equation (6.2.4.1) to estimate some limiting, converged state of information. The object class is a non-random, static attribute and each new observation improves the state of information in the system. The significance of the recursive nature of equation (6.2.4.1) is that the information state after k (= 1, 2, ...) observations may be estimated.

Another point to note is that this approach estimates the entropy of the distribution of belief in the object class estimate. We recall from the discussion of section 5.5 that no answer at all equates to the state where all classes are equally likely and gives maximum entropy. Thus, a wrong answer produces a "better" entropy result than no answer at all. Therefore, this model does not discriminate between right and wrong answers. We require a more complex model.

To discriminate in this way we require the distribution, $\hat{d}_i$, of the decision (or report) made as a result of the joint class distribution. It is assumed that any useful identification system is more likely to report the true identity than any single false

identity. Under this assumption, entropy improves with true reported identity and worsens with false reported identity. It is further assumed that all the observations are made regarding the same object and that that object belongs to one of the known classes. After k observations, the joint class decision distribution for an object of true class i is:

$$\hat{d}_i = \begin{bmatrix} \hat{d}_{i1} \\ \vdots \\ \hat{d}_{iq} \end{bmatrix}$$

where:

$$\begin{aligned}
\hat{d}_{ij} = \quad & r_j(k,k) \binom{k}{0} c_{ij}^k \\
& + r_j(k-1,k) \binom{k}{1} c_{ij}^{k-1}(1-c_{ij}) \\
& + r_j(k-2,k) \binom{k}{2} c_{ij}^{k-2}(1-c_{ij})^2 \\
& + \ldots \\
& + r_j(0,k) \binom{k}{k} (1-c_{ij})^k
\end{aligned}$$

where the $r_j(n,k)$'s are the probabilities of reporting an object as class j given that n sensor reports out of k have made that classification, c_{ij} is drawn from matrix C and $\binom{k}{n}$ is the binomial coefficient:

$$\binom{k}{n} = \frac{k!}{n!(k-n)!}.$$

This is equivalent to calculating the scalar product of the vector of r_j's with a vector of binomial probabilities of observing class j respectively k, $k-1$ ... 0 times in k observations when the true class is i.

$$\hat{d}_{ij} = r_j \cdot b_{ij} \tag{6.2.4.3}$$

If the sensor performance may be modelled as uniform, so that it results in a similar maximum entropy distribution (with $c_{ii} = c_{max}$) for every true object class, the r_j's obey

a simple voting relationship. If the sensor performance is strongly non-uniform, so that its performance varies greatly from class to class, Bayesian techniques can be used to derive the r_j's [8] [51].

6.2.5 Models of Ignorance - the Penalty of Missing Information

All the above models operate under the assumption that the observations are capable of providing information on the full set of attributes and for all objects of interest. The models predict the quality of the fused information for fully observed objects. Clearly, in many situations this will not be the case and information will be partial or missing. It is possible that no observations are available of one or more attributes of a particular object. There might even be no observations of the object at all.

In such circumstances there exist two options. The partial or missing object can be ignored or a numerical value may be assigned to the effect on information quality of a missing object or attribute. The choice depends on the circumstances under which the entropy measure has been obtained and the use to which the measure is to be put.

In Chapter 2, possible uses of data fusion MoPs and MoEs were discussed, including: on-line system "health" monitoring, system control for adaptive behaviour, performance evaluation for concepts and designs and evaluation of prototypes and operational systems.

The requirements for MoPs and MoEs may well vary from one application to another. For instance, any automatic on-line system monitoring might be required to operate unsupervised for long periods. Such a system would be required to deliver a high probability of fault detection and a low probability of false alarm; characteristics most likely to be achieved through monitoring the information state of stable, fully converged data. Missing data would not be apparent to such a system and partial data would not satisfy the criteria of stability and convergence.

An adaptive control system, such as an automatic sensor allocation and tasking unit could operate by comparing observed information quality levels with prescribed required levels. Missing data would not be apparent to this kind of process, but the ability to recognise the presence of partial data could be crucial in allowing information needs to be detected and used as the basis for sensor control.

For system evaluation, particularly theoretical evaluation of concepts and designs, missing data would be an important element in any MoP or MoE and, provided sufficient trial instrumentation was available, so that the true state may be known independently of the system under evaluation, missing data would be equally important in the evaluation of a prototype or operational system. Some way of quantifying the quality of missing or partial information is required.

Ignorance in object classification has been discussed above. No information corresponds to the a priori case where all classes are equally likely. The entropy of object classification in such circumstances is $\ln\{q\}$ where q is the number of possible classes.

The same idea may be applied to numerical discrete random variables. However, if the distribution of the unknown parameter were allowed to drift unboundedly to infinity its entropy will also be infinite and that will not help to quantify data fusion effectiveness. In practice, every parameter can be argued to have finite bounds. If we recall the number of objects in a region example of section 5.4.2 and let the objects be aircraft in a fixed region of interest around a busy airport, if we then make simple assumptions about the minimum horizontal and vertical separation of aircraft we quickly arrive at a maximum number (n_{max}) that can fit into the region. If nothing is known about the number of aircraft, any number between zero and the maximum may be present with equal probability and $\ln\{1 + n_{max}\}$ is the required entropy.

The same principles may be applied to continuous random variables. Consider the above example again and recall the sensor observation depicted in Figures 3.1 and 3.2.

The sensor observed the range and azimuth angle (horizontal polar co-ordinates) of objects. The distribution of errors about the resulting estimates was represented in figure 3.1 as ellipses in the 2-Dimensional space of the sensor observation. In the 3-Dimensional region of Figure 3.2 the observations lack height information. The locations of aircraft observed by this sensor might be expressed as points on the ground, vertically below the aircraft, but nothing could be known about their heights. The errors in height could be represented by Gaussian distributions with infinite standard deviation or by uniform distributions from 0 to ∞, but both approaches would make their entropies infinite [51].

Again there is a finite range of interest as there was in the case of the number of aircraft in a given airspace. Shannon tells us that the distribution giving the maximum entropy for a variable, enclosed within a bounded region, is a uniform distribution [75]. Thus the convention will be adopted that the entropy of a continuous attribute for which no estimate is available will be that of a uniform distribution over the region of interest for that variable. For the above example, height is modelled as uniformly distributed between 0 and ht_{max}. The resulting contribution to information entropy from ignorance of height is $\ln\{ht_{max}\}$.

When objects are known to be present, independently of a system under evaluation, but have not been detected by the system, all attributes may be modelled in one of the ways described above. Thus, the information error of an unseen object may be quantified as the sum of the entropies of all attributes unobserved. In other words the object state will be modelled as uniformly distributed over the entire region of interest.

6.2.6 Fusion States and Switching Data Fusion Outcomes

The preceding sections propose models capable of predicting aspects of the behaviour of a data fusion system. In practice, rarely can any system be characterised by a single model that will represent it fully and with fidelity. In many cases a hierarchy of

processes in the system implies a hierarchy of models, with one or more models representing each process. The autonomous distributed system data fusion architecture certainly has this property. At the lowest level, single sensor trackers operate. Each has its own model predicting its performance. At the higher levels, track fusion models operate. These models take as inputs the predicted performance levels of the individual trackers and predict the resulting track-fused performance, and so on up to the highest level in the hierarchy.

This is not the whole story. We cannot assume that every object is observed by every sensor, tracked by every tracker or that every higher track has full data available as input. At any time the quality of information available regarding a particular object might be that associated with one of the many possible Fusion States.

We define a Fusion State as the outcome of any subset of the data fusion process that constitutes a feasible data fusion process in its own right and is sufficient to exploit the available sensor observations.

To clarify this, consider as an example the very simplest of Track Fusion processes. Two sensors are served by independent trackers and the output of the trackers is Track Fused. In this situation four Fusion States are possible. Respectively they are: neither sensor tracking; the first sensor tracking alone; the second sensor tracking alone; and both sensors tracking with Track Fusion producing higher tracks. Each state can be represented by one or more of the models described above and each model will lead to a different predicted performance.

•"Neither sensor tracking" can be represented by the ignorance model of section 6.2.5;

•"First, sensor tracking alone" and "second sensor tracking alone" can be represented by the combination of a data association model from section 6.2.1 and an appropriate tracker model from section 6.2.2.

•"Both sensors tracking" would be represented by the data association and

tracker models used above combined with an extra data association model from

section 6.2.1 with parameters set to reflect the track to track relationships and a

track fusion model from section 6.2.3.

If we observed many objects in similar situations we would observe a mixture of the

behaviours associated with the four Fusion States. We require a model of this mixture.

6.2.6.1   The Track Life Cycle

Blackman considered the transition from non-tracking to tracking and back for a

single sensor (the track life cycle) [8]. Simple rules for track formation and deletion were

proposed and modelled as a Markov chain. A similar set of such rules is reflected in

Figure 6.12 where a new track is formed when two out of any three successive

attempted observations result in detection and an existing track is deleted if n successive

attempted observations fail to detect.

Figure 6.12: Track Life-Cycle Markov Chain [8]
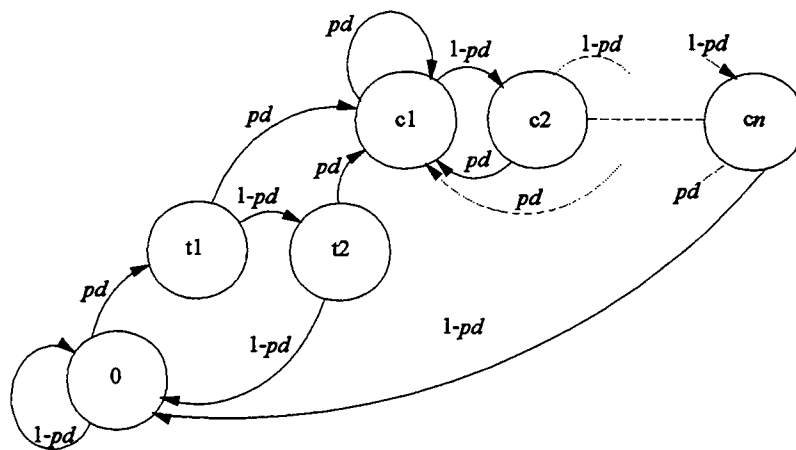
A model of a simplified track life cycle. Tracks begin when the same object is observed twice
in three attempts and end when no observation is successful in n attempts.

The chain of Figure 6.12 has the property of ergodicity. The important property

of an ergodic Markov chain from the point of view of this work is that the probability of

selecting one of many objects at random and encountering it in a particular state is the same as the probability of encountering a particular object in that state when the object is observed over many state transitions. Under these conditions, an ergodic Markov process model will converge to give the required probabilities. If we equate Markov process states with the system behaviours encapsulated in Fusion States, this offers the prospect of predicting the proportions of the mixtures of behaviours from a description of the mechanisms by which changes in behaviour come about.

In the model of Figure 6.12, state 0 corresponds to no track at all, states t_1 and t_2 correspond to an internal (non-reported) tentative track based on one observation, states $c_1...c_i...c_n$ correspond to a confirmed track, last observed i cycles ago, Pd is the probability of sensor detection and it is assumed that the probability of correct Data Association $Pr_c \approx 1$. A state transition takes place every time an attempt is made to observe the object in question. Blackman's model used the Markov chain definition to create a state transition matrix, Φ:

$$\Phi = \begin{bmatrix} 1-pd & 0 & 1-pd & 0 & 0 & \cdots & 0 & 1-pd \\ pd & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1-pd & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & pd & pd & pd & pd & \cdots & pd & pd \\ 0 & 0 & 0 & 1-pd & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-pd & 0 \end{bmatrix}$$

The appropriate mixture of fusion states is calculated by the iteration:

$$s(k) = \Phi s(k-1)$$

initialised with $s(0) = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T$, i.e. no tentative or confirmed track where

$s = \begin{bmatrix} s_0 & s_{t1} & \dots & s_{cn} \end{bmatrix}^T$, $s_0 = \mathrm{Pr}\{\text{state } 0\}$ and $s_{t1} = \mathrm{Pr}\{\text{state } t_1\}$ etc. Then when the iteration converges to a value of s, the tracking probabilities are given by:

$$\mathrm{Pr}\{\text{no confirmed track}\} = s_0 + s_{t1} + s_{t2}, \text{ and}$$

$$\Pr\{\text{track}\} = \sum_{i=1}^{n} s_{ci}.$$

6.2.6.2 Fusion States and System Performance

We can extend this simply to describe the Track Fusion example by applying the track life-cycle model independently to each sensor and calculating the conditional probability of the states: no track, single sensor track (two possibilities) and fused track.

In general, the performance of data fusion, in each state, i, will be different and it is proposed that the overall performance be modelled by a linear combination of the performance of the individual states with coefficients equal to probabilities of the states, p_i.

$$H = \sum_i p_i H_i$$

For the track fusion example, the effect would be that shown in figure 6.13. The figure covers the approach of an object from a start position, where $pd \approx 0$ for both sensors, to a final position where $pd \approx 1$ for both sensors. The dotted lines represent the performance levels associated with individual states. Each state is represented by a distinct line and the information held for any individual object will have an entropy which is approximated by one of the lines.

During periods when the system performance line is approximately coincident with a Fusion State line, that state dominates the performance of the system. In such circumstances, the performance of the system for an individual object is readily predictable. For instance, there are three intervals in figure 6.13 when this occurs. Initially, it is very rare for any track to exist at all and the average system performance coincides closely with the state of ignorance. For a brief period in the middle of the sequence system performance coincides with the sensor 2 alone Fusion State. At this point, Pd for sensor 2 is sufficiently high to give $\Pr\{\text{track}\} \approx 1$. At the same time, sensor

1 still has Pr{track} ≈ 0. At the end of the sequence, Pr{track} ≈ 1 for both sensors and Track Fused information is available with probability approximately unity.

When the system performance line does not coincide with a fusion state line, an object may be observed in one of several states. When the system performance progresses from ignorance to sensor 2 alone, the information quality for individual tracks does not adopt some intermediate value. Instead, a random sample of objects would contain a mixture of (predominantly) objects with no track and objects tracked by sensor 2. The value of the average system performance line derives from the proportions of the mixture. Similarly, during the transition from sensor 2 alone to track fused, a mixture of these two states is found in a random sample of objects.

Figure 6.13: A Mixture of Fusion States
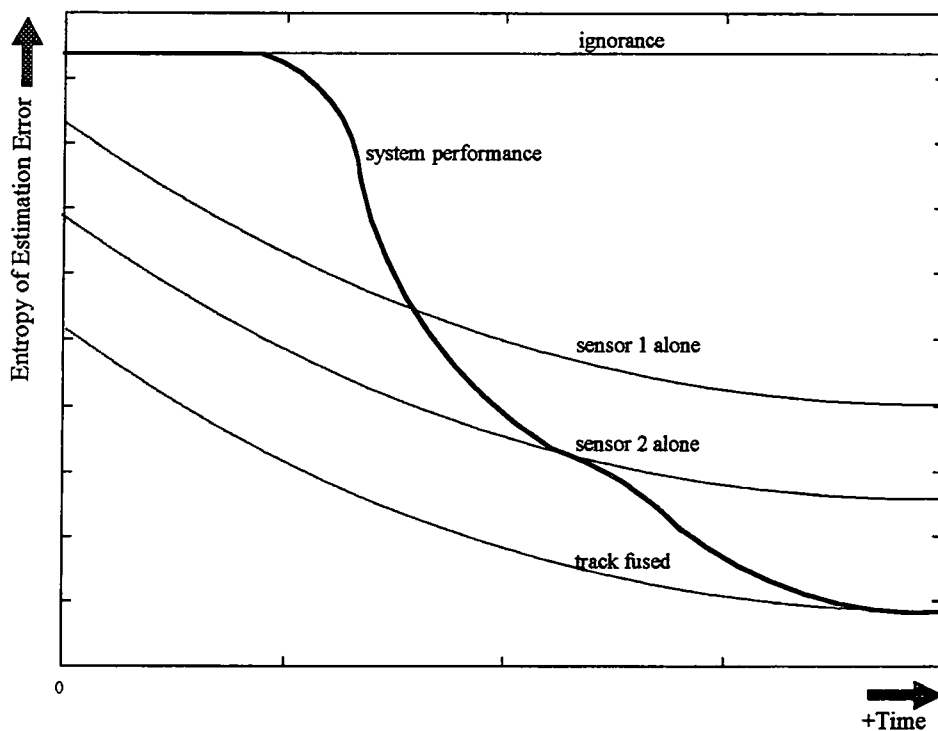
The performance of a Track Fusion process coincides with the Track Fusion model only when both sensors are tracking with high probability.

The sensor 1 alone Fusion State is highly unlikely to occur because it requires sensor 1 to have formed a track simultaneously with sensor 2 having failed. The physical properties of the sensors, in this hypothetical situation, effectively rule this out.

6.2.6.3 Data Association Fusion States

Recalling the discussion of Data Association models, it was stated that the Fusion States for data association could produce a complicated and dynamic mixture of behaviours when $pd \neq 1$. It is proposed to model this mixture of behaviours by extending the track life cycle Markov chain. If the Markov chain of Figure 6.12 was redrawn for two objects, and was made more general by allowing $\Pr_c \neq 1$ the result is a new Markov chain. The first few transitions are represented in Figure 6.14. State 0 represents no tracks in the system and in state 0 four possibilities exist:

- No new observation, with probability $p(0, 0) = (1 - pd_a)(1 - pd_b)$, in which case the model stays in state 0;

- A single new observation, with probability $p(a, 0) = pd_a(1 - pd_b)$, or $p(0, b) = (1 - pd_a)pd_b$, in which case the model moves to a single tentative track state, t1a or t1b respectively.

- Two new observations, with probability $p(a, b) = pd_a pd_b$ in which case the model moves to the double tentative track state, t1a t1b.

From any state where tentative or confirmed tracks exist, the correct data association probability, $\Pr_c$, must be accounted for. We noted in the discussion of Data Association models that $\Pr_c$ could change depending on whether one or more tracks were present in the system and on whether one or more observations were available to update them. This changed model for $\Pr_c$ is denoted by $\Pr_c'$ in Figure 6.14. Also, pd is affected when a track is updated on successive cycles by observations from different objects. Sudden increases in velocity error that are introduced make the object more

difficult to find subsequently. Tracks affected in this way are denoted *cix* (e.g. c1x in Figure 6.14).

From the state t1a, seven transitions are possible. The next state depends on the new observations made and, except in the case of no new observations, on the correct association probability.

Figure 6.14: Track Life-Cycle Markov Chain (two objects)
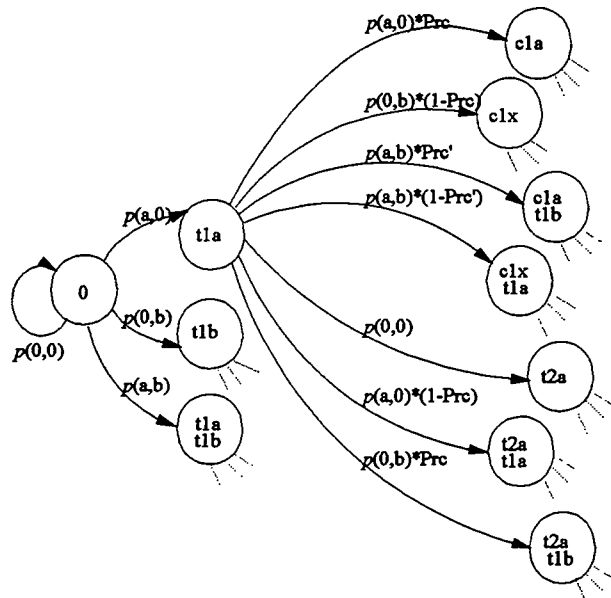
The track life-cycle Markov chain becomes much more complex in the presence of multiple objects and the possibility of Data Association errors.

Whilst this model of a relatively simple situation is complicated, it is by no means intractable. With careful examination of the possible transitions, the models proposed here allow an accurate representation of the mixture governing the Fusion States to be created.

7 Experiments and Case Studies

In this Chapter, the results of three validation experiments and a case study will be given. The validation experiments were comparisons of the models proposed in Chapter 6 with detailed Monte Carlo simulations. Their purpose was to confirm the models and investigate their accuracy. The case study was of a real system.

In the validation experiments, computer Monte Carlo simulations were created. The details of each will be given in the following sections. For the real system, performance measurements were made. In all the experiments a performance model of the system was created. Details of these models are given below also. The results take the form of tables and graphs of predicted versus simulated or measured performance.

7.1 Validation of the Data Association Models

The data association models proposed in Chapter 6 were validated. The details of the simulations, models and results are as follows.

7.1.1 Simulation Definition : Data Association

Two simulations were produced: one for a 2-dimensional attribute and observation space, nominally the Cartesian pair (x, y), and a second for a 4-dimensional space, nominally (x, y, u, v). The scenario in each case was simply that two objects at locations x_1 and x_2 were separated by the vector amount δ such that:

$$\lambda = \delta^T [P_1 + P_2]^{-1} \delta ,$$

was the normalised distance square of the separation $\delta = x_1 - x_2$. The notation $\hat{x}_i$ denotes the estimated object state and y_i denotes a new observation where $\hat{x}_1, \hat{x}_2 \sim N(0, P_1)$ and $y_1 y_2 \sim N(0, P_2)$. Normalisation was with respect to the covariance of the relative errors between an estimate and an observation of an object, namely $P_1 + P_2$.

P_1 and P_2 were the estimation error and observation error covariance matrices respectively. For simplicity, P_1 and P_2 were simulated as diagonal matrices so that the errors in the individual state vector elements would be independent. This approximation implies that the singular value decomposition (SVD) axes for the estimation and observation error covariance matrices for the two objects were assumed to be aligned. For close objects observed by a single sensor this approximation is usually an accurate one.

Estimates of $y_1 - \hat{x}_1$ and $y_2 - \hat{x}_2$ were generated using the Box - Muller approximation to introduce pseudo-random Gaussian-distributed errors [65]:

$$x \sim N(0, \sigma) \approx \sigma \times \cos(2\pi \times \text{rand}_1) \times \sqrt{2\ln(\text{rand}_2)} \qquad (7.1.1.1)$$

where rand_1 and rand_2 are a pair of random numbers drawn from the approximate uniform distribution $0 \leq \text{rand} < 1$. Then x provides a value for a single element of a simulated vector of MVGD errors.

The decision process of a NNSF algorithm was carried out and two pools of observations were created, one for each object. For a sequential NNSF logic, the right or wrong observation y_1 or y_2 was allocated to the pool for existing estimate $\hat{x}_1$ based on which of z and w was the smaller respectively, where:

$$z = (y_1 - \hat{x}_1)^T [P_1 + P_2]^{-1}(y_1 - \hat{x}_1) \text{ and}$$

$$w = (y_2 - \hat{x}_1)^T [P_1 + P_2]^{-1}(y_2 - \hat{x}_1).$$

The remaining observation was then allocated to $\hat{x}_2$. For a simultaneous NNSF logic, the observations were allocated correctly or incorrectly based on which of z' and w' was the smaller respectively, where:

$$z' = (y_1 - \hat{x}_1)^T [P_1 + P_2]^{-1}(y_1 - \hat{x}_1) + (y_2 - \hat{x}_2)^T [P_1 + P_2]^{-1}(y_2 - \hat{x}_2) \text{ and}$$

$$w' = (y_2 - \hat{x}_1)^T [P_1 + P_2]^{-1}(y_2 - \hat{x}_1) + (y_1 - \hat{x}_2)^T [P_1 + P_2]^{-1}(y_1 - \hat{x}_2).$$

In this manner, pools of 10^6 observation were allocated to each object.

The simulation was performed for a range of values of λ from 0 to 36 for the sequential algorithm and 0 to 18 for the simultaneous algorithm. The following results were recorded:

- Pr(false allocation) the probability that an observation is falsely allocated;

- $\bar{y} = E[y \, | y \in y(\hat{x}_i)]$ the expected value of the observations allocated to object i; and

- $\bar{\sigma}_y^2 = E\left[(y - \bar{y})^2 \, | y \in y(\hat{x}_i)\right]$ the variance of the observations allocated to object i.

Initially these results were obtained for $|P_1| \ll |P_2|$ and the (observation error : relative error) error square ratio, $\dfrac{|P_2|}{|P_1+P_2|} \approx 1$. A second set of results was obtained for

$|P_1| \gg |P_2|$ and $\dfrac{|P_2|}{|P_1+P_2|} \approx 0$. The models proposed in Chapter 6 correspond to these extreme cases. Finally, for various values of $\sqrt{\lambda}$, the simulations were repeated for a range of values of error square ratio from 0.1 to 0.9 in steps of 0.1.

7.1.2 Model Definition : Data Association

Models of data association performance as proposed in Chapter 6 section 6.2.1 were produced. If the theory proposed in Chapter 6 was correct, the models would predict the statistics of the pools of allocations generated during the simulations. For the situation where the error square ratio ≈ 1 the system behaviour was dominated by the observation errors and the models were:

- Probability of correct association was: $\mathrm{Pr}_c = \mathrm{erf}(a) = \dfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{a} \exp\left(-\dfrac{t^2}{2}\right) dt$.

- The expected offset in apparent object position (in the direction away from the interfering object) was b. For a sequential algorithm the bias $b \approx \mu_t$ and for a simultaneous algorithm $b \approx \dfrac{\mu_t}{\sqrt{2}}$ where:

$$\mu_t = \frac{1}{\sqrt{2\pi}}\left[\int_{-\infty}^{a} t\exp\left(-\frac{t^2}{2}\right)dt + \int_{-\infty}^{-a}\left(t+\frac{a}{2}\right)\exp\left(-\frac{t^2}{2}\right)dt\right].$$

- The expected pool standard deviation (along the line joining the true and interfering objects) was σ'. For a sequential algorithm $\sigma' \approx \sqrt{\sigma_t^2}$ and for a simultaneous algorithm $\sigma' \approx \sqrt{\frac{\sigma_t^2 + 1}{2}}$ where:

$$\sigma_t^2 = \frac{1}{\sqrt{2\pi}} \left[\int_{-\infty}^{a} (t-b)^2 \exp\left(-\frac{t^2}{2}\right) dt + \int_{-\infty}^{-a} \left(t + \frac{a}{2} - b\right)^2 \exp\left(-\frac{t^2}{2}\right) dt \right] .$$

$$\text{where } \quad a = \begin{cases} \frac{\sqrt{\lambda}}{2} & \text{for a sequential algorithm} \\ \\ \sqrt{\frac{\lambda}{2}} & \text{for a simultaneous algorithm} \end{cases}$$

For the situation where error square ratio ≈ 0 the system behaviour was dominated by the estimation errors and the models were:

- Probability of correct association was as above.

- The expected offset in apparent object position (in the direction towards the interfering object) was: $b = (1 - \Pr_c)\sqrt{\delta^T [P + R]^{-1} \delta}$.

- The expected pool variance (along the line joining the true and interfering objects) was obtained from: $\bar{P}_2 = P_2 + (\Pr_c - \Pr_c^2)[\delta\delta^T]$.

For other values of error square ratio a linear combination of the models was used with coefficients $\frac{|P_2|}{|P_1 + P_2|}$ and $1 - \frac{|P_2|}{|P_1 + P_2|}$ respectively.

7.1.3 Comparison of Results : Data Association

The following table 7.1 gives the simulation results of Pr(false association) which were used to validate the model of probability of correct association in the sequential Data Association algorithm. In figure 7.1 these data are plotted as point markers over the curve produced by the model output. Both simulations agree well with the simulation results.

Table 7.1: Simulated Pr(false association) for the Sequential Algorithm

λ	2-D simulation	4-D simulation
0.1	0.4370	—
0.2	0.4114	
0.3	0.3922	
0.4	0.3759	—
0.6	0.3492	
0.8	0.3275	
1.0	0.3087	0.3081
1.4	0.2773	
1.8	0.2514	
2.4	0.2195	
2.5		0.2138
3.0	0.1934	
4.0	0.1585	0.1579
5.0	0.1317	
6.0	0.1102	
7.0		0.0921
8.0	0.0787	
10.0	0.0567	0.0565
14.0	0.0305	
16.0		0.0226
18.0	0.0169	
22.0	0.0095	
29.0	0.0035	

Figure7.1: Probability of False Allocation (sequential data association algorithm)
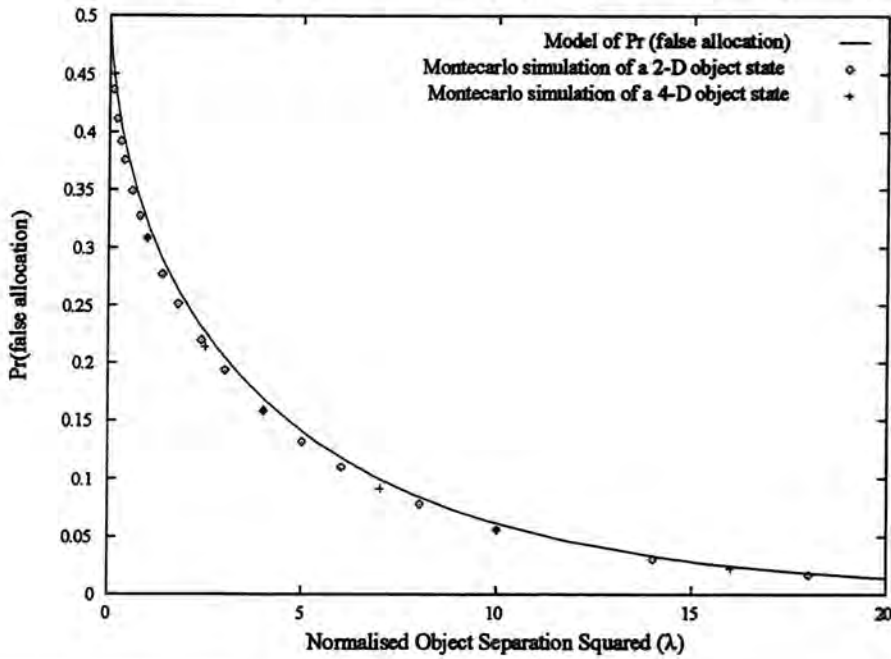
For the sequential algorithm, the model corresponds well with the probability of false allocation obtained in simulation. The dimensionality of the system does not affect the model and a value of λ in a 4-Dimensional system has the same significance as it does in a 2-Dimensional system

Table 7.2 gives the corresponding simulation results from the simultaneous Data Association algorithm. In figure 7.2 these data are plotted as point markers over the curve produced by the model output. Again, both simulations agree well with the model results.

Table 7.2: Pr(false association) for the Simultaneous Algorithm

λ	2-D simulation	4-D simulation
0.1	0.4111	
0.2	0.3754	
0.3	0.3489	
0.4	0.3270	
0.6	0.2915	
0.8	0.2635	
1.0	0.2395	0.2395
1.4	0.2012	
2.0	0.1713	
2.4	0.1365	
2.5		0.1319
3.0	0.1104	
4.0	0.0787	0.0790
5.0	0.0567	
6.0	0.0414	
7.0		0.0307
8.0	0.0228	
10.0	0.0129	0.0128
14.0	0.0042	
16.0		0.0024
18.0	0.0014	

Comparison of the graphs in figures 7.1 and 7.2 shows a distinct performance advantage of the simultaneous algorithm over the sequential algorithm. This was predicted in Chapter 6 and is borne out by these results. For any value of normalised object separation, the probability of false allocation is less for the simultaneous algorithm and the probability reduces to an insignificant level for smaller values of normalised object separation. The simultaneous algorithm benefits from a decision made on the basis of more information.

Both algorithms produce results which are unaffected by the dimension of the system. Again this result was predicted in Chapter 6. This means that all available data

dimensions should be used in the Data Association process. Consideration of an extra

dimension must give a normalised separation square which is greater than or equal to the

lower-dimensional value and equality occurs only when there is no-object separation in

the new dimension. These results show that performance improves as separation

increases.

Figure 7.2: Probability of False Allocation(simultaneous data association algorithm)
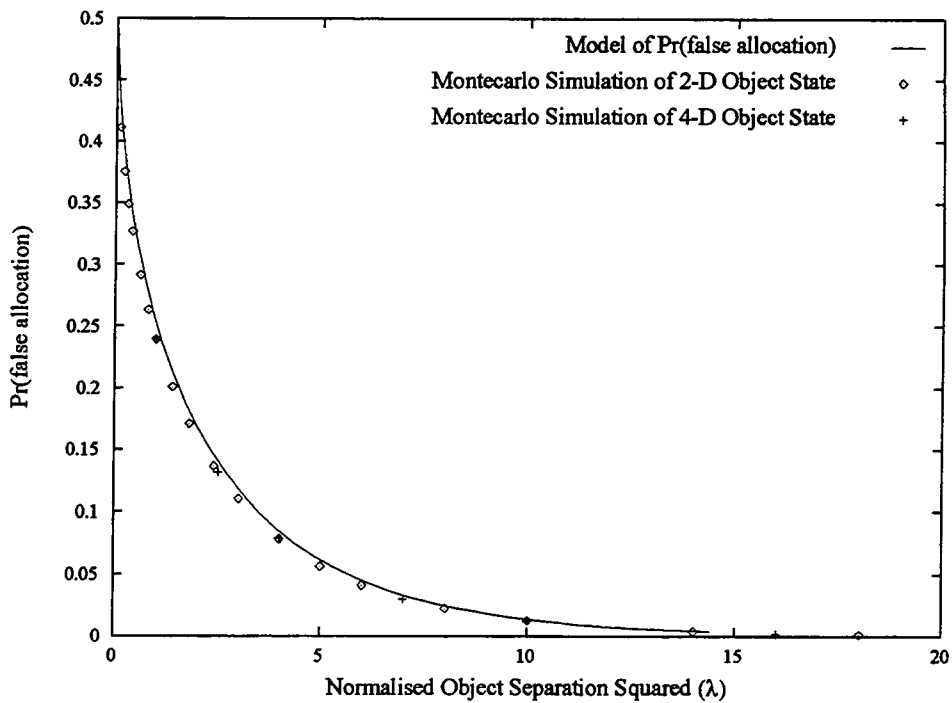
For the simultaneous algorithm, the model shows equally good correspondence and the same independence of system dimension.

Table 7.3(a) gives the simulation and model results for estimation bias in the data

sets allocated by the sequential Data Association algorithm in a 2-dimensional system.

The lower line corresponds to $|P_1| \ll |P_2|$ and the association process dominated by

observation errors. The upper line corresponds to $|P_1| \gg |P_2|$ and the association

process dominated by estimation errors. In figure 7.3(a) the simulation data are plotted

as point markers and the model output as a curve. The simulations agree well with the

model results. Correlation error free operation corresponds to a bias of zero and as the

object separation increases, the simulation and the model show the bias and hence the

correlation errors diminishing and eventually dying out.

Table 7.3(a): Min. and Max. Estimation Bias Due to Data Association Error

$\sqrt{\lambda}$	bmin - simulation	bmin - model	bmax - simulation	bmax - model
0.000	-0.5641	-0.5641	0.0000	0.0000
0.100	-0.5155	-0.5156	0.0475	0.0473
0.200	-0.4697	-0.4698	0.0891	0.0890
0.300	-0.4266	-0.4269	0.1252	0.1251
0.500	-0.3487	-0.3492	0.1814	0.1814
0.700	-0.2818	-0.2821	0.2175	0.2180
1.000	-0.1995	-0.2000	0.2400	0.2406
1.500	-0.1047	-0.1053	0.2169	0.2174
2.000	-0.0501	-0.0509	0.1576	0.1581
2.500	-0.0218	-0.0226	0.0963	0.0968
3.000	-0.0085	-0.0094	0.0509	0.0512
3.500	-0.0030	-0.0039	0.0232	0.0235
4.000	-0.0009	-0.0018	0.0092	0.0092
5.000	0.0000	-0.0008	0.0009	0.0009
6.000	0.0000	-0.0007	0.0001	0.0000

Figure 7.3(a): Upper and lower bounds of Estimation Bias due to Allocation Errors
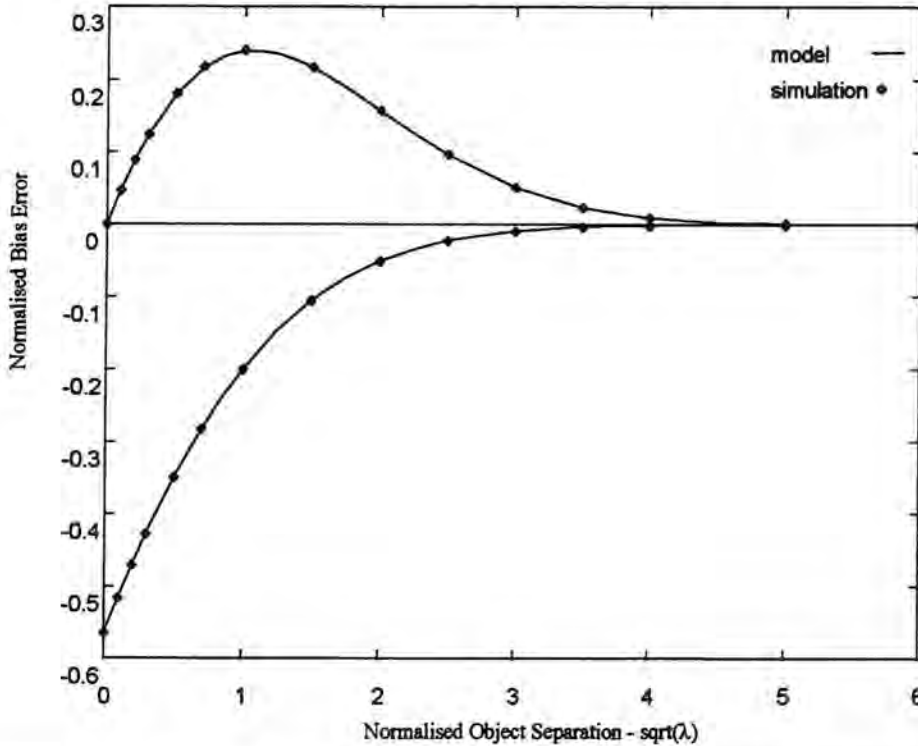
Expected estimation bias due to Data Association error is approximately bounded above where $|P_1| \gg |P_2|$ and below where $|P_1| \ll |P_2|$. The expected bias in any situation is predictable from the ratio of observation to estimation error variance and the normalised object separation.

Table 7.3(b) gives the simulation and model results for apparent normalised observation error standard deviation in the direction of the interfering object in the data sets allocated by the sequential Data Association algorithm in a 2-dimensional system. This time, unlike table 7.3(a), the lower line corresponds to $|P_1| \gg |P_2|$ and an association process dominated by estimation errors. The upper line corresponds to $|P_1| \ll |P_2|$ and the association process dominated by observation errors. In figure 7.3(b) the simulation data are plotted as point markers and the model output as a curve. The simulations agree well with the model results.

Table 7.3(b): Min. and Max. Sample Standard Deviation Due to Data Association Error

$\sqrt{\lambda}$	σ'_{min}-simulation	σ'_{min}-model	σ'_{max}-simulation	σ'_{max}-model
0.000	0.0000	0.0000	0.8253	0.8250
0.100	0.0499	0.0500	0.8259	0.8255
0.200	0.0994	0.0995	0.8275	0.8272
0.300	0.1479	0.1480	0.8303	0.8299
0.500	0.2404	0.2404	0.8387	0.8381
0.700	0.3239	0.3242	0.8501	0.8497
1.000	0.4271	0.4274	0.8718	0.8714
1.500	0.5275	0.5280	0.9118	0.9116
2.000	0.5388	0.5395	0.9469	0.9466
2.500	0.4812	0.4822	0.9718	0.9715
3.000	0.3875	0.3886	0.9868	0.9865
3.500	0.2840	0.2857	0.9944	0.9942
4.000	0.1920	0.1921	0.9979	0.9977
5.000	0.0653	0.0671	0.9996	0.9996
6.000	0.0290	0.0141	0.9997	0.9998

Correlation error free operation corresponds to a normalised observation error standard deviation of zero at the lower bound and unity at the upper bound. Thus, the effects on data fusion performance are different at the lower and upper bounds. At the lower bound, the true value of normalised observation error approaches zero and there is more variation in the allocated data than the true data. The effects of this are likely to be destructive with divergent estimates and lost tracks. Recalling the discussion of Chapter

3 an estimator produced in this way is unlikely to be conservative and the true distribution of errors in any estimate will have a wider spread than that predicted by the fusion process. This will occur even if a robust estimation technique is applied because the problem lies in the allocated set of observations. At the upper bound, there is less variation in the allocated data set than in the true data and the effects on data fusion performance are likely to be benign. Again, as the object separation increases, the simulation and the model show the correlation errors diminishing and eventually dying out.

Figure 7.3(b): Upper and lower bounds of Sample Standard Deviation due to Allocation Error:
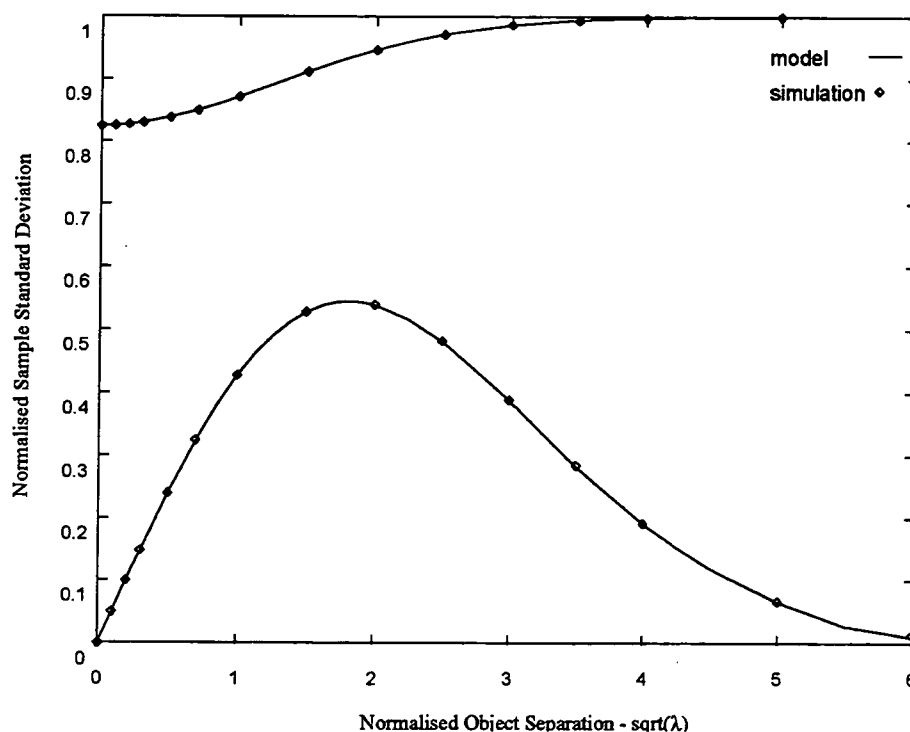
Expected change in error standard deviation due to Data Association error is approximately bounded above where $|P_1| \ll |P_2|$ and below where $|P_1| \gg |P_2|$.

Table 7.3(c) gives the normalised estimation bias and observation standard deviation of the pool of allocated observations for the range of values of error square ratio, $\dfrac{|P_2|}{|P_1 + P_2|}$, from 0 to 1. Zero corresponds to $|P_1| \ll |P_2|$ and unity to $|P_1| \gg |P_2|$. Results are given for $\lambda = 1.0$ and 3.0.

Table 7.3(c): Normalised Estimation Bias and Observation Standard Deviation versus the Error Square Ratio

$\sqrt{\lambda}$	$\dfrac{\|P_2\|}{\|P_1+P_2\|}$	b – model	b – simulation	σ' – model	σ' – simulation
1.0	0.0	0.2400	0.2406	0.4271	0.4274
	0.1	0.1961	0.2449	0.4901	0.5338
	0.2	0.1521	0.2342	0.5458	0.6158
	0.3	0.1082	0.2139	0.5964	0.6840
	0.4	0.0642	0.1869	0.6430	0.7425
	0.5	0.0203	0.1524	0.6864	0.7927
	0.6	-0.0237	0.1096	0.7273	0.8347
	0.7	-0.0677	0.0557	0.7660	0.8670
	0.8	-0.1116	-0.0161	0.8028	0.8846
	0.9	-0.1556	-0.1117	0.8380	0.8806
	1.0	-0.1995	-0.2000	0.8718	0.8714
3.0	0.0	0.0509	0.0512	0.3876	0.3886
	0.1	0.0450	0.0406	0.4822	0.4701
	0.2	0.0390	0.0309	0.5612	0.5410
	0.3	0.0331	0.0218	0.6303	0.6051
	0.4	0.0271	0.0140	0.6926	0.6657
	0.5	0.0212	0.0078	0.7497	0.7244
	0.6	0.0153	0.0029	0.8027	0.7811
	0.7	0.0093	-0.0004	0.8525	0.8368
	0.8	0.0034	-0.0031	0.8995	0.8899
	0.9	-0.0026	-0.0060	0.9442	0.9398
	1.0	-0.0085	-0.0094	0.9868	0.9865

Figure 7.3(c) gives graphs of these results. The graphs indicate the effects of Data Association errors on estimation performance along the vertical lines $x = 1$ and $x = 3$ drawn on figures 7.3(a) and 7.3(b) between the upper and lower performance bounds.

In figure 7.3(c) the simulation data are plotted as point markers and the model output as a curve. The simulations agree well with the model results for $\sqrt{\lambda} = 3$ and for larger values of $\sqrt{\lambda}$ the correspondence is found to improve further. As the separation increases the model and simulation results for the allocated data set approach the true population values and the probability of data association error diminishes such that the bias approaches the line $y = 0$ and the standard deviation approaches the line $y = \sqrt{x}$ with

$0 < x < 1$ and $y \geq 0$. For $\sqrt{\lambda} = 1$, the correspondence is less good but is still a useful indicator of the parameters of Data Association performance.

Figure 7.3(c): Estimation Bias and Sample Standard Deviation due to Allocation Errors

simulated estimation bias	◆
modelled estimation bias	———
simulated apparent observation sigma	+
modelled apparent observation sigma	——

i) Normalised Object Separation = 1   ii) Normalised Object Separation = 3
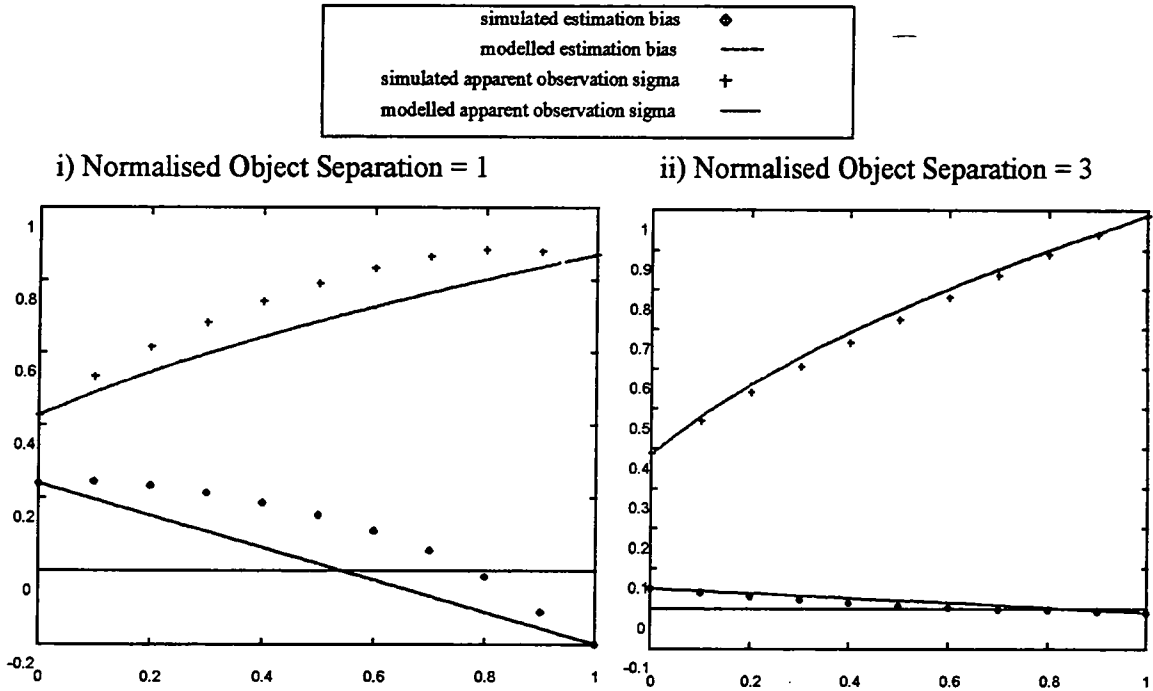
The models of bias and standard deviation indicate the effects of Data association error on estimation over the range of error ratios, $\dfrac{|P_2|}{|P_1+P_2|}$. 0 on the horizontal axis coincides with the upper bound of figure 7.3(a) and 1 with the corresponding lower bound. For the bounds of figure 7.3(b), the opposite relationship is true.

The values of $\sqrt{\lambda}$ given in the tables represent the lower end of the range of interest. For Tracking applications of Data Association, separations less than three (3-sigma) are not of interest because an observing sensor would be highly unlikely to discriminate between separate objects. Lower values are of interest for Track Fusion applications of Data Association only.

7.2 Validation of the Tracking and Track Fusion Models

The tracking and track fusion models proposed in Chapter 6 were validated. The details of the simulations, the models and the results are given in the following sections. These results were first reported at the IEE Data Fusion Conference in 1996 [59].

Two sensors, separated by several miles, observed, independently, a pair of distant approaching objects. An object tracking process was performed at each sensor and the resulting tracks were communicated, periodically, to a third party. Immediately after each communication, the third party performed track fusion. The entropy of the track fusion result was recorded.

The objects were simulated as if capable of autonomous manoeuvres. Thus X, the system under observation, was stochastic by nature. A model of this system and the accompanying observation and data fusion processes was produced and compared with a detailed simulation.

7.2.1 Simulation Definition: Tracking and Track Fusion

The simulation comprised three main elements: two similar Radar models represented the sensors, an EKf algorithm was used to represent the tracking processes and a MMSE was used to represent track fusion.

Data association was assumed perfect and communication was assumed to occur immediately after each tracker update. These simplifications were adopted to isolate tracking and track fusion performance from other effects and avoid complicated interactions which would make it difficult to draw clear conclusions.

The Radar model was as defined in Chapter 6. Spatial coverage followed a raster scan pattern. Detection behaviour was governed by the parameters:

- $P_t G A_e$ - transmitted power $\times$ antenna gain$\times$ effective antenna receiving area;

- σ - the mean RCS;

- NL - the sum of the environmental and systematic noise $\times$ propagation losses; and

- p_{fa} - the probability of a detection due to noise .

The Radar equation gave:

$$\frac{S}{N} = \frac{P_t G A_e \sigma}{(4\pi)^2 r^4 N L} \quad \text{and} \quad P_d \approx P_{fa}^{\overline{\left(1+\frac{S}{N}\right)^{-1}}} .$$

A detection was made when $P_d >$ rand and rand was a random number uniformly distributed so that $0 \leq$ rand < 1. Once an object was detected an observation y was created with $y \sim N(h(x), R)$. These MVGD errors were simulated as described in section 7.1 using the Box - Muller approximation of equation (7.1.1.1) [65].

The EKf algorithm executed the recursive calculations of equations (3.2.1.2 a to e):

$$x(k, k-1) = \Phi x^i(k-1, k-1)$$

$$P(k, k-1) = \Phi P^i(k-1, k-1)\Phi^T + Q$$

$$K = P(k, k-1)H^T[HP(k, k-1)H^T + R]^{-1}$$

$$x^i(k, k) = x(k, k-1) + K[y_k - h(x(k, k-1))]$$

$$P^i(k, k) = [I - KH]P(k, k-1)$$

where $H = \frac{\partial}{\partial x} h(x(k, k-1))$. For the sake of conciseness, the superscript i has been omitted from all but the main outputs of the algorithm, $x^i(k, k)$ and $P^i(k, k)$.

The MMSE Track Fusion algorithm was as defined in equations (3.2.2.1) and (3.2.2.2) and for conciseness the (k, k) qualifier is omitted:

$$\hat{x} = \hat{x}^i - P^i[P^i + P^j]^{-1}[\hat{x}^j - \hat{x}^i]$$

$$P = \left(I - P^i[P^i + P^j]^{-1}\right)P^i$$

The entropy of the MVGD posterior error distribution, $H(\varepsilon) = \frac{1}{2} \ln\{|P|\} + G$, was stored and the average over 500 repetitions of a 90 second tracking sequence was calculated.

7.2.2 Model Definition: Tracking and Track Fusion

Models of tracking and track fusion performance as proposed in Chapter 6 section 6.2.2 were produced. Again, if the theory proposed in Chapter 6 was correct, the

models would predict the entropy of the MVGD posterior error distribution recorded during the simulations.

The EKf algorithm was modelled by iterating the equation (6.2.2.5).

$$P_n^i \approx A_{n-1} \sum_{l=0}^{n-2} (\Phi A_{n-2} \ldots \Phi A_l) Q (\Phi^T)^{n-2-l} + A_{n-1} Q$$

This expression was iterated until $p_{n-1}^i - p_n^i < \delta_p \, \forall p^i$ where δ_p is the convergence threshold and p^i are the elements of P^i. The entropy was then calculated:

$$H(\varepsilon^i) = \tfrac{1}{2}(\ln|P_n^i| + G)$$

The MMSE track fusion algorithm was modelled using equation (6.2.3.2):

$$H = H(\varepsilon^i) + H(\varepsilon^j) - H(\varepsilon^{ij})$$

where $H(\varepsilon^{ij}) = \tfrac{1}{2}\ln\{|P^i + P^j|\} + G$. When multiple Fusion States were modelled, an ergodic Markov process model of the form:

$$s(n) = \psi s(n-1)$$

which was iterated until $s_{n-1} - s_n < \delta_\mu \, \forall s$

where the s are the individual state probabilities.

7.2.3 Comparison of Results: Tracking and Track Fusion

Figure 7.4 is the graph of relative entropy of the information error versus scenario time sampled at one second intervals. It is the entropy of the information error experienced by the receiving third party in its track-fused estimates. Three lines are reproduced.

- The relative entropy of a single simulation.

- The mean relative entropy over many simulations.

- The relative entropy predicted by the model.

The relative entropy of a single simulation demonstrates the characteristic "saw tooth" shape of figure 3.7. Between updates, the relative entropy rose as the observer's uncertainty regarding unseen manoeuvres increased. As each track fusion update was

made, a step decrease in relative entropy of the information error occurred. These steps were not uniform because *Pd* was modelled realistically and sometimes only one sensor observed the target and communicated new information and the other sensor missed an update. These missed updates were experienced as smaller steps in relative entropy.

Figure 7.4: A Track Fusion Scenario
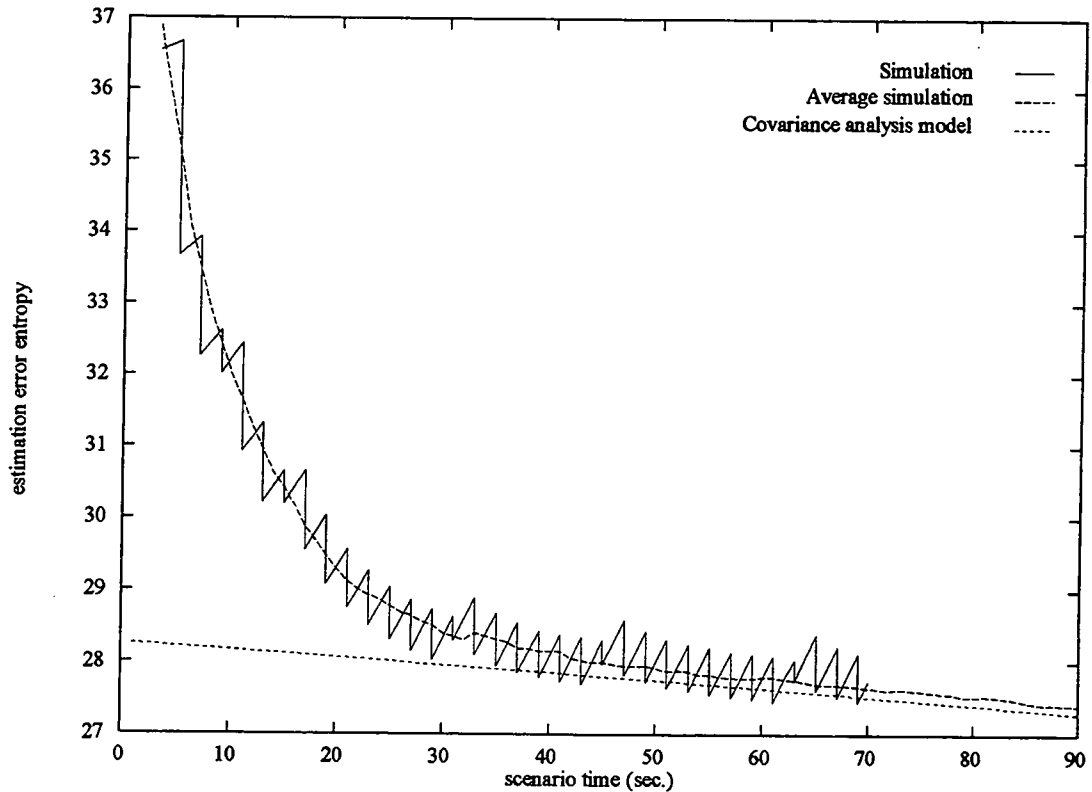
The modelled information entropy of a Track Fusion process is confirmed through simulation. But convergence takes 40 seconds and the behaviour prior to convergence demonstrates the presence of other Fusion States.

The mean entropy over 500 simulations was calculated in order to enable direct comparison with the model prediction. The saw-tooth effect is no longer present because the samples were not synchronised.

The entropy predicted by the covariance analysis model produced an almost straight line with a slight negative gradient. The gradient was due to the "shrinking" of the sensor resolution cell of the simulated sensors as the objects approached. If we

consider a measurement error standard deviation m-box, it was defined with respect to spherical polar co-ordinates and the volume was $r^2 \cos \psi \sqrt{|R|}$ where r was object range and ψ was elevation rotation from the horizontal plane. Such a volume "shrinks" as it gets closer to the origin or its elevation rotation above or below the horizon increases. The origins for the trackers were the sensor locations respectively. As a result, when the objects got nearer to the sensors, the observations got more accurate and the entropy reduced with respect to attribute space.

It is clear that, in the early part of the scenario, up to the 40 seconds mark, the model was not predicting the behaviour of the simulation. This was due to two factors. Firstly, the simulation started with no prior information of the system state at time zero with neither sensor having established tracks. It required time to converge whereas the model started with the assumption of a fully converged system. Secondly, the model represented only one of the fusion states present in the system. In the simulation four fusion states were possible:

- Neither sensor tracking.

- The first sensor tracking alone.

- The second sensor tracking alone.

- Both sensors tracking.

The entropies of the states were different, and the simulation produced the average entropy of a mixture of them. However, the model predicted the value of the lowest-entropy, "both sensors tracking" state. This is a key finding because the effect predicted in figure 6.13 is thus demonstrated by these results. Prior to the 40 second mark, the simulation produced a mixture of the four states and as the objects approached closer and the probability of track acquisition improved, the mixture became purer and the information entropy approached to the modelled state.

From 40 seconds of simulation, onwards, the probabilities of detection and tracking for the sensors had risen to the point where the "both sensors tracking" state dominated and the model became an acceptable approximation to the performance of the system. Even then, the model underestimated the information error entropy by between 0.1 and 0.15. Assuming the underestimate to affect all attributes equally, this would correspond to an underestimate of, at most, 1.25% of error standard deviation.

A variation of the above experiment was performed. Multiple fusion state models were applied to different sensors in the same track fusion scenario. This time the sensors were TV trackers, capable of providing line of sight (direction) data only. In figure 7.5 a spatial plot of information (= -1*entropy) over the x, y plane is given. The measure has been negated to improve clarity. We will refer to this representation as the Information Map of the data fusion process. The information map may be represented alternately as a surface with the highest points corresponding to the best information or as a set of contours.

The information map in figure 7.5 has four significant features:

•a "crease" or depression in the information quality on the line joining the sensors, this is a feature of tracking with two line of sight sensors where no triangulation is possible in this region and the system cannot resolve the three dimensional location of any object;

•an outer stable region where there are few contours and the information quality changes slowly, this corresponds to regions covered by a single sensor and again the system cannot resolve the three dimensional location of any object;

•two roughly semicircular inner regions where there are no contours and information quality changes slowly, this corresponds to regions covered by both sensors with $Pd \approx 1$ and here the system can resolve the three dimensional location of any object; and finally

•a ring of close contours where the average information quality changes rapidly, in this region a mixture of single and double sensor information would be encountered and the ratio of the mixture would change rapidly as contours were crossed.

Figure 7.5: Multiple Fusion States

An Information Map   An Effectiveness Map
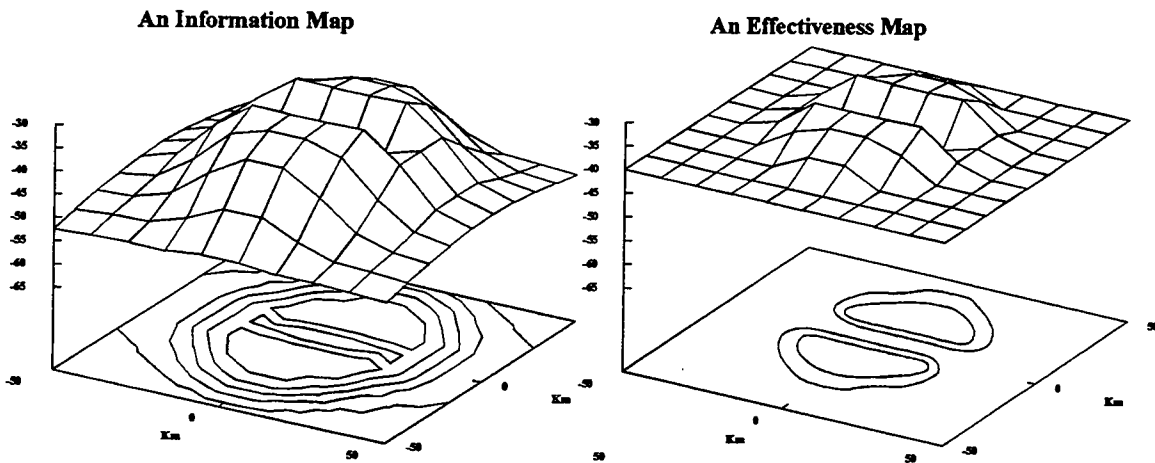
> Spatial representation of the model outputs is also possible. The information map predicts the data fusion performance against an object at any location within a region of interest. The effectiveness map isolates those regions where the system is effective.

Where full, 3-dimensional data could not be resolved, a model of ignorance, of the type proposed in Chapter 6, was used which treated the missing data as uniformly distributed over the full width of the region of interest along the line of the deficiency.

Thus, the key characteristics of figure 6.13 are present in figure 7.5. There are stable, predictable regions where the data fusion performance corresponds to a single Fusion State and regions where mixtures of states are observed.

Figure 7.5 also includes an Effectiveness Map. Again, this may be represented as a surface or a set of contours. The Effectiveness Map is created by applying the SoP H_0, which corresponds to the performance level at which the data fusion system is deemed to become effective:

$$H_E = \min[H(\varepsilon), H_0]$$

In figure 7.5, the contours correspond to

$\Pr(H(\varepsilon) < H_0) = 0.5$ and 0.95 respectively. Again the data has been negated for clarity. There are two effective regions in this example.

7.3   Validation of the Object Classification Models

The object classification models proposed in Chapter 6 were validated. There was a model of the entropy of classification belief and another model of the distribution of classification decisions. The details of the simulations, the models and the results are given in the following sections.

A single sensor observed an object repeatedly and produced an independent classification each time in the form of the favoured object class. In the experiment to validate the model of the entropy of classification belief, the object came from one of three classes and the probability that any individual classification was correct was 0.6. In the experiment to validate the model of the distribution of classification decisions, the object came from one of two classes and the probability that any individual classification was correct was 0.75.

In both experiments, incorrect classifications always chose one of the wrong classes. A "no statement" outcome was not possible. The objects were incapable of changing their class. Thus $\mathbf{X}$ was non-random.

7.3.1 Simulation Definition: Object Classification

Sensor observations were simulated using a thresholding process. Given the observation error distribution $c_i = \begin{bmatrix} \cdots c_{ij} \cdots \end{bmatrix}$ for the true object class i resulting in reported class j and a random number from the uniform distribution $0 \le \text{rand} < 1$, the reported class j is the lowest indexed class for which:

$$\sum_{l=0}^{j} c_{il} > \text{rand} .$$

A recursive Bayesian inference process was performed as each new observation became available and an entropy measure of the joint object class was recorded [8] [51]. The recursive Bayesian process takes the form:

$$p(x|Y^k) = \frac{p(y_k|x) \times p(x|Y^{k-1})}{p(y_k|Y^{k-1})}$$

where $p(y_k|Y^{k-1}) = \sum p(y_k|x) \times p(x|Y^{k-1})$, y_k is the current measurement and Y^k is the set of measurements up to and including step k.

The entropy measure depended on the model which was being evaluated. In the experiment to validate the model of the entropy of classification belief, the measure was simply:

$$H_k = \sum_x p(x|Y^k) \ln\left\{ \frac{1}{p(x|Y^k)} \right\},$$

calculated after observation k and averaged over 10^4 repetitions of the simulation. For the experiment to validate the model of the distribution of classification decisions, the decision was recorded after each observation for reported class j:

$$d_{k,n} = \left[\cdots \delta_{jl} \cdots \right], \text{ where } \delta_{jl} = \left\{ \begin{array}{ll} 1 & j = l \\ 0 & j \neq l \end{array} \right. \text{ is the Dirac delta.}$$

The distribution of decisions after observation k after 10^4 repetitions of the simulation is then:

$$d_k = 10^{-4} \times \sum_n d_{k,n} \text{ and } H_k = \sum_x d_k \ln\left\{ \frac{1}{d_k} \right\}.$$

7.3.2 Model Definition: Object Classification

The model of belief is based on the entropy of the recursive Bayesian classifier given in equations (6.2.4.1) and (6.2.4.2). It is assumed that the single observation distribution of classification, c_i, is known and for the purposes of this experiment is:

$$c_i = \left[\begin{array}{ccc} 0.6 & 0.2 & 0.2 \end{array} \right]^T,$$

where the probabilities have been reordered for clarity such that the true identity corresponds to the leftmost element. Belief entropy after k observations is modelled:

$$H[p(x|Y^k)] = H[p(y_k|x)] + H[p(x|Y^{k-1})] - H[p(y_k|Y^{k-1})]$$

where $H[p(x|Y^{k-1})]$ is the result of the previous iteration and $H[p(y_k|x)$ is enumerated

from c_i. The third term is calculated from $p(y_k|Y^{k-1}) = \sum_x p(y_k|x)p(x|Y^{k-1})$ where an

estimate of the distribution of $p(x|Y^{k-1})$ is obtained from a numerical solution of:

$$H_{max} = -c_{max} \ln[c_{max}] - (1 - c_{max}) \ln\left[\frac{(1-c_{max})}{q-1}\right] ,$$

and assumption of the maximum entropy distribution form as per Figure 5.4. The

required entropy can then be enumerated from the distribution of $p(y_k|Y^{k-1})$ so

obtained. This model predicted the belief entropy directly.

The model of decision entropy was based on the relationship given in equation

(6.2.4.3). The distribution of decisions after k observations was where the subscript i for

true class i has been omitted for the sake of clarity:

$$\hat{d}_k = \begin{bmatrix} \hat{d}_{1,k} & \cdots & \hat{d}_{q,k} \end{bmatrix}^T$$

and each element obeys:

$$\hat{d}_{j,k} = r_{j,k} \cdot b_{ij,k} .$$

The vector r is the vector of probabilities of selecting class j when zero to up to k

observations from the k observations to date favour class j. The vector b is the binomial

distribution of k trials with success probability c_{ij}, element j of c_i. In this case:

$$c_i = \begin{bmatrix} 0.75 & 0.25 \end{bmatrix}^T$$

The model entropy was then $H_k = \sum_x \hat{d}_k \ln\left\{\frac{1}{\hat{d}_k}\right\}$.

7.3.3 Comparison of Results: Object Classification

7.3.3.1 Model of Belief Entropy for Object Classification

In table 7.6 the simulated and modelled belief entropy for zero up to 20

observations of the three-class example are given. Also given is the corresponding

decision entropy measured in the simulation. Figure 7.6 is the graph produced from these data.

The model prediction showed the characteristic, stepped shape of figure 3.3 represented as bars here. The simulation, represented here as diamond markers, and the model produced acceptable, if not exact, correspondence. As the number of observations increased, the model underestimated the performance that was measured in the simulation. However, at its widest, the discrepancy corresponded to an error of less than 1% in the value of c_{max} (the declared belief in the favoured class).

Table 7.6: Results of Belief Entropy Experiment

Number of Observations	Model of Belief Entropy	Simulated Belief Entropy	Simulated Decision Entropy
0	1.099	1.099	1.099
1	0.950	0.950	0.950
2	0.826	0.825	0.950
3	0.723	0.719	0.826
4	0.635	0.630	0.764
5	0.561	0.553	0.705
6	0.498	0.483	0.632
7	0.443	0.424	0.573
8	0.396	0.373	0.528
9	0.355	0.328	0.475
10	0.319	0.286	0.431
11	0.288	0.251	0.394
12	0.261	0.224	0.355
13	0.236	0.199	0.321
14	0.215	0.175	0.292
15	0.196	0.154	0.263
16	0.179	0.134	0.237
17	0.163	0.119	0.215
18	0.150	0.105	0.194
19	0.137	0.091	0.175
20	0.126	0.079	0.159

The time scale is measured in numbers of observations. Object Class is usually a non-random attribute which cannot change in the interval between observations and was assumed to be so in this experiment. Thus information quality did not change between observations irrespective of the elapsed time.

Figure 7.6 : Entropy of Object Classification Belief (cii=0.6, q=3)
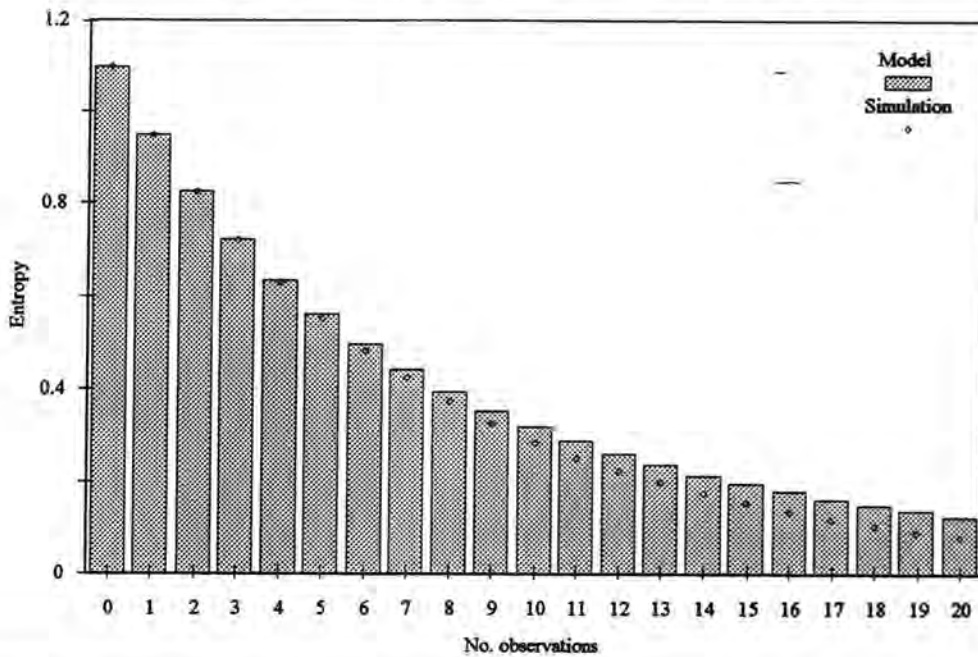
The modelled information entropy of Object Classification belief is confirmed through simulation. The Object Class is non-random so the length of the interval between observations does not affect the information quality.

7.3.3.2 Model of Decision Entropy for Object Classification

In table 7.7 the simulated and modelled decision entropy for zero up to 20 observations of the two-class example are given. Figure 7.7 is the graph of object classification decision entropy versus number of observations. The model prediction again showed the characteristic, stepped shape of figure 3.3 again represented by bars. The simulation results are given as diamond markers. Clearly, the correspondence is very close. An interesting feature of the graph was that entropy reduced only for odd numbers of observations. If the error distributions for all classes are identical maximum entropy distributions, the Bayesian approach is identical in its behaviour to voting and with an even number of observations, a tie was a possibility. Until such a tie was resolved, the information in the system did not increase. Thus, the model behaviour was as expected.

Table 7.7: Results of Decision Entropy Experiment

Number of Observations	Model of Decision Entropy	Simulated Decision Entropy
0	0.693	0.693
1	0.562	0.560
2	0.562	0.557
3	0.433	0.435
4	0.433	0.435
5	0.333	0.328
6	0.333	0.332
7	0.255	0.246
8	0.255	0.259
9	0.195	0.199
10	0.195	0.203
11	0.149	0.145
12	0.149	0.147
13	0.114	0.110
14	0.114	0.108
15	0.087	0.086
16	0.087	0.087
17	0.067	0.061
18	0.067	0.059
19	0.051	0.049
20	0.051	0.046

Figure 7.7 : Entropy of Object Classification Decisions (cii=0.75, q=2)
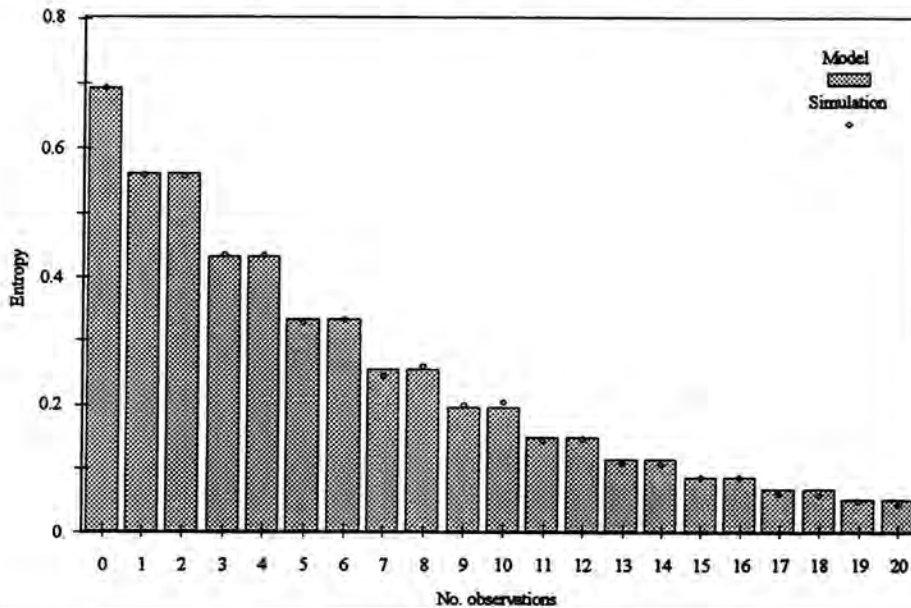
The model of Object Classification decisions is confirmed through simulation. There are two classes to choose from and the sensor's classification capability is identical for both classes. As a result, information quality improves only for odd numbers of observations.

In figure 7.8, the decision entropy and belief entropy from the simulation of experiment 7.3.3.1 are plotted together. It can be seen clearly that the decision entropy

did not improve between the first and second observation but after that, every update gave an improvement.

Figure 7.8 : Entropy of Classification Belief and Decision (cii=0.6, q=3)
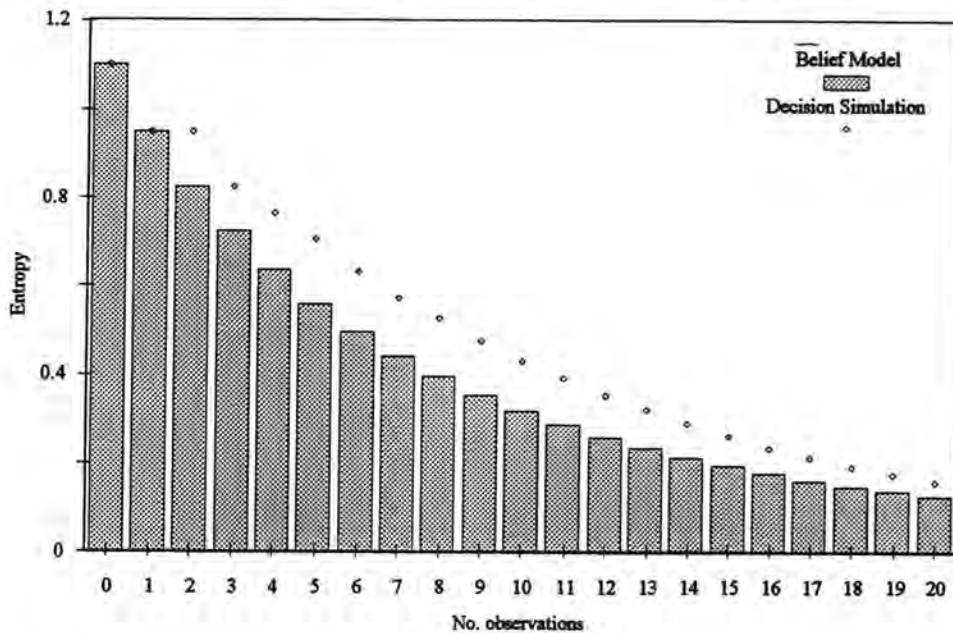
> **The distribution of belief in the outcome of a Bayesian classification process and the distribution of the classification decision are not the same. The belief distribution is overly optimistic.**

More importantly, the entropy of belief was lower than that of the decision, implying that the belief distribution gave an overoptimistic view of information quality. This is the object classification equivalent of a non-conservative estimate and is a recognised drawback of the Bayesian inferential approach. (For estimation applications, the Covariance Intersection Working Group defined a conservative estimate as one where the true error distribution is enclosed by the predicted distribution [14].) This was predicted in Chapter 6 and was due to wrong answers making a positive contribution to the entropy because their entropy was lower than that of ignorance.

7.4   BAE SYSTEMS Technology Demonstrator

ICIDS is a laboratory based data fusion technology demonstrator system developed by the BAE SYSTEMS, Sowerby Research Centre in Bristol in collaboration

with the department of Engineering at Oxford University. Initially, SKIDS was developed to investigate aspects of workplace automation. It was described by B. S. Y. Rao, Durrant-Whyte and Sheen in a 1992 publication. More recently, it has evolved into the ICIDS system and has been used by Deaves et. al. to investigate more general issues associated with distributed fusion networks [18] [19] [66].

Four ceiling-mounted video cameras observed an area of a room. Objects which entered the area were tracked and, if they belonged to a known class (vehicle type 1, vehicle type 2 or human), were identified. All sensor measurements were pooled in an optimal data fusion algorithm. The algorithm comprises a distributed Kalman filter (DKf) as defined in equations 3.2.1.3 a to d and a distributed object classifier.

Define $z(k, k) = P^{-1}(k, k)x(k, k)$, then the DKf algorithm may be written:

$$P(k.k-1) = \Phi P(k-1, k-1)\Phi^T + Q$$

$$z(k, k-1) = P(k, k-1)^{-1} \Phi P(k-1, k-1) z(k-1, k-1)$$

$$z(k, k) = z(k, k-1) + \sum_i H_i^T R_i^{-1} y_i(k)$$

$$P^{-1}(k, k) = P^{-1}(k, k-1) + \sum_i H_i^T R_i^{-1} H_i$$

where the index i denotes sensor.

A case study was undertaken to create a model of the distributed tracker in the ICIDS system, using published parameters and descriptions [66]. The model predicted the levels of performance, achievable in theory, from the combination of sensors and processing centres available to the tracker.

To enable the model to be validated, colleagues at the Sowerby Research Centre measured the performance of ICIDS. This section describes the model and the results of the validation.

7.4.1 Measured Performance: BAE SYSTEMS Technology Demonstrator

Two characteristics of performance were measured:

- The spatial coverage of the sensors,

- The accuracy of object tracks.

The observed region was a 6 metre by 3 metre rectangle. For recording purposes, it was divided into one metre square divisions, labelled A1 to C6 in three rows A to C with six divisions in each row. Figure 7.9 shows the measured map of sensor coverage. Within each square is a list of sensor numbers found to observe an object placed in that square.

The corresponding measured performance is given in table 7.9.

Table 7.9: BAE SYSTEMS Technology Demonstrator Measured Data

Row	Division	Number of sensors	Covariance Determinant	Information Entropy
A	d1	2	4,508.9	9.88
	d2	3	1,760.7	9.41
	d3	3	1,717.5	9.40
	d4	4	753.3	8.98
	d5	3	1,487.6	9.33
	d6	2	3,997.4	9.82
B	d1	3	1,709.5	9.40
	d2	3	1,713.6	9.40
	d3	4	739.2	8.98
	d4	4	734.1	8.98
	d5	4	741.1	8.98
	d6	3	1,581.4	9.36
C	d1	3	4,841.2	9.92
	d2	4	738.3	8.98
	d3	4	732.1	8.97
	d4	4	732.1	8.97
	d5	3	1,471.2	9.32
	d6	3	2,111.7	9.50

Figure 7.9: ICIDS Measured Sensor Coverage

The distribution of sensor coverage over a measurement grid of one metre squares. Data fusion performance was measured with respect to this grid

Object track accuracy was measured in terms of the determinants of the tracking error covariance matrices, $|P|$, in each division of the room. These were converted to information entropies assuming MVGD errors and using the relationship:

$$H = \tfrac{1}{2}(\ln|P| + m\ln 2\pi e) \text{ with } m = 4.$$

These measurements were plotted as a stepped surface. Views from camera 1 and camera 3 are reproduced in figures 7.10(a) and (b). Each measured covariance determinant is the average of 100 measurements. The standard deviation of the 100 measurements was also provided.

Figure 7.10: ICIDS Measured Performance

(a): Entropy View from Camera 1

(b): Entropy View from Camera 3
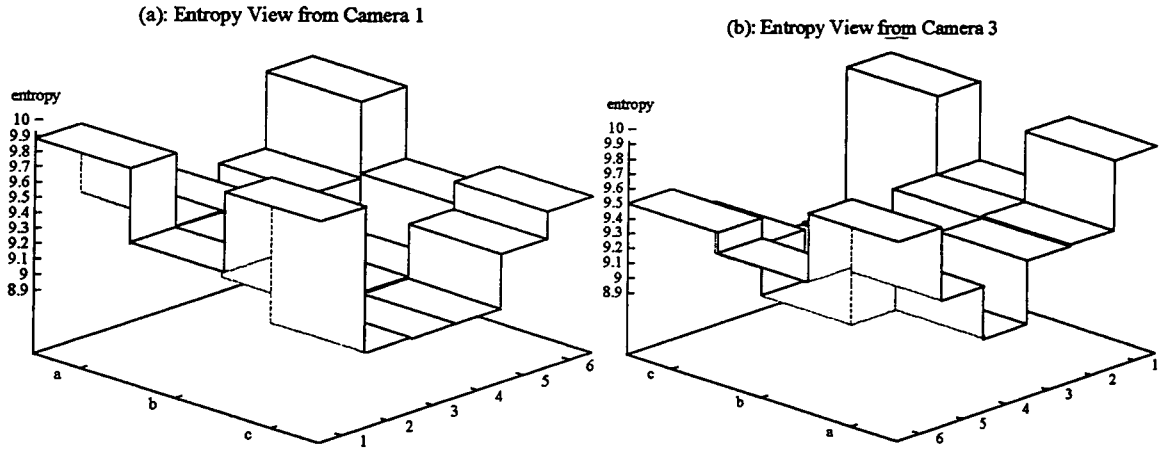
The entropy map of measured data fusion performance for the BAE SYSTEMS Technology Demonstrator

7.4.2 Model Definition: BAE SYSTEMS Technology Demonstrator

Models were created of two, three and four sensors observing a division. The architecture of the system was of the form of figure 6.6 and an appropriate single tracker model was used as defined in equation 6.2.2.5.

$$P_n \approx A_{n-1} \sum_{i=0}^{n-2} (\Phi A_{n-2} \dots \Phi A_i) Q (\Phi^T)^{n-2-i} + A_{n-1} Q$$

This expression was iterated until $p_{n-1} - p_n < \delta_p \forall p$ where δ_p is the convergence threshold and p are the elements of P. The entropy was then calculated:

$$H = \tfrac{1}{2}(\ln|P_n| + m \ln 2\pi e)$$

The parameters of the model were taken from B. S. Y. Rao and Sheen's description of the SKIDS system with some amendments resulting from discussions with colleagues at the Sowerby Research Centre, the current users of the system [66].

7.4.3 Comparison of Results: BAE SYSTEMS Technology Demonstrator

Table 7.11 gives the modelling results alongside the values calculated from the measured covariance determinants. The graphs of entropy per division, reproduced in figures 7.11 (a to c), were obtained from these data .

Table 7.11: BAE SYSTEMS Technology Demonstrator Modelling Results

Row	Division	Entropy of model	Entropy of Measured Data
A	d1	10.02	9.88
	d2	9.23	9.41
	d3	9.23	9.40
	d4	8.67	8.98
	d5	9.23	9.33
	d6	10.02	9.82
B	d1	9.23	9.40
	d2	9.23	9.40
	d3	8.67	8.98
	d4	8.67	8.98
	d5	8.67	8.98
	d6	9.23	9.36
C	d1	9.23	9.92
	d2	8.67	8.98
	d3	8.67	8.97
	d4	8.67	8.97
	d5	9.23	9.32
	d6	9.23	9.50

These results were disappointing. If the discrepancy between the modelled and simulated results were assumed to affect all attributes equally, this would reflect an error of 4% in standard deviation. This is more than three times the largest errors experienced in the validation experiment. Also, the application examined during the validation was considerably more dynamic than this one and most of the error in the validation experiment might be explained by low representational fidelity for system dynamics. In short, the errors should have been smaller, not larger.

In an attempt to understand this, measurement sequences were examined in detail. Figure 7.12 is plotted from the 100 measurements made at one of the divisions observed by four sensors. The measurements were provided with a warning that sensor

processing equipment sometimes "lost" observations before they reached the tracking algorithm. This was reflected as a variation in the measured covariance determinant and quantified as a standard deviation for each measurement.
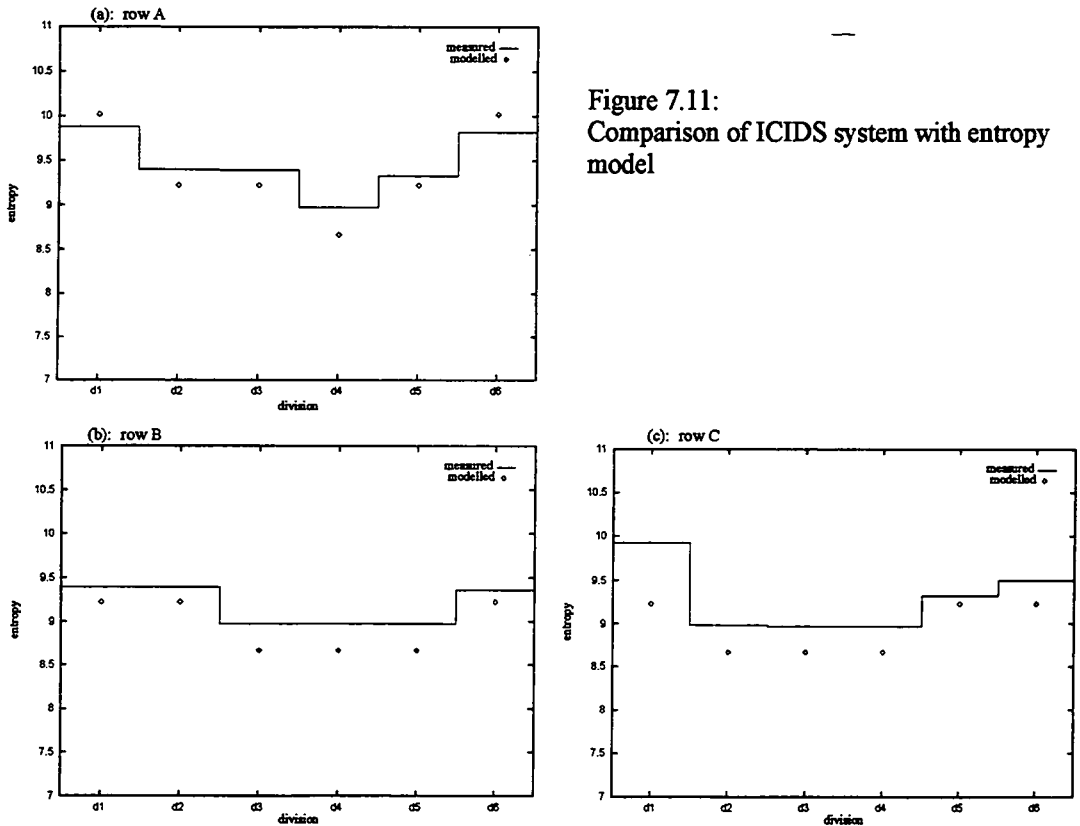
Figure 7.11:
Comparison of ICIDS system with entropy model

Comparison of the modelled and measured performance of the BAE SYSTEMS Technology Demonstrator.

Inspection of the data showed that the distribution of measured values was not a random one. There were only four values and the measured sequence switched between them. Thus standard deviation could not be used to infer a likely spread of values.

However, the switching appeared to follow a random sequence. Further inspection showed that all values were between the performance levels predicted by the models for three and four sensors and the average of the 100 measurements lay approximately mid way between those two levels.

The "lost" observations should have been reflected in the model parameters as a detection probability $Pd < 1$ but the appropriate value was unknown. All modelling had

been performed assuming $Pd = 1$. Had this been true, very little variation would have been experienced in the measured performance. If the four levels were interpreted as fusion states corresponding to the elapsed time since the last lost observation. The simplest interpretation of this would assume a lost observation $1, 2, 3$ and more than 3 sensor cycles before the current measurement. Under this assumption, the proportions of these states implied $Pd = 0.86$.

Figure 7.12: Detailed Results: observation by four sensors
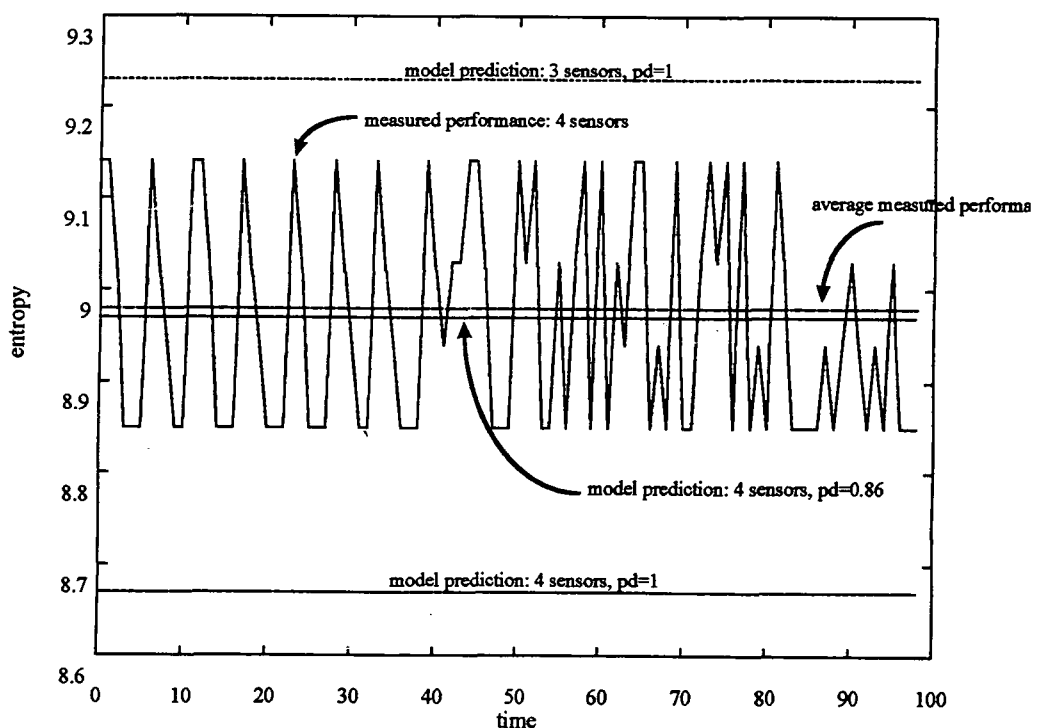
The measured information entropy switched between four values, suggesting that four distinct system states were being observed. The assumption that the states related to the elapsed time since the last lost observation yielded an estimate of Pd of 0.86.

The model for four sensors was rerun using this value and the correspondence with the measured data improved dramatically. In terms of standard deviation the error reduced to less than 0.15%. Whilst this is very promising, it is a result based on speculation. Also, the behaviour in the three and two sensor divisions is likely to be much more complex and it is unlikely that a few simple assumptions would lead to a justifiable "correction" to the results. The measurements must be repeated with detailed

note taken of the behaviour of the system before reliable conclusions can be drawn. Unfortunately, the system has been dismantled and the lab is currently in use for other experiments. As a result, this will not be possible in the immediate future.

7.5   Experiments and Case Studies: Summary and Discussion

In this Chapter the results of three validation experiments and a case study were given. The validation experiments were comparisons of the models proposed in Chapter 6 with detailed Monte Carlo simulations. Their purpose was to confirm the models and investigate their accuracy. The case study was of a real system. In the validation experiments, computer Monte Carlo simulations were created specifically for this work. In all cases at least one new performance model of each system was created.

The data association models proposed in Chapter 6 were validated. Two simulations and two models were produced: one simulation for a 2-dimensional attribute or observation space, nominally the Cartesian pair (x, y) and a second for a 4-dimensional space, nominally $(\ x, \ y, \ u, \ v\ )$; and a model for a sequential and a simultaneous data association logic. It was shown that:

- The models predicted accurately the statistics of the data associated with a known object.

- The same model applied equally to a 2-dimensional and 4-dimensional attribute space and that data association performance for a given normalised separation square was independent of the dimension of the attribute space. This has the consequence that an increase in the number of independent attributes used in Data Association decisions improve performance provided that the objects are separate in the new attributes.

- Given $Pd \approx 1$ and an association process dominated by observation errors, correlation error produces biased estimates but the corresponding reduction in apparent observation error covariance results in an apparent distribution of

Page 224

observations which is conservative with respect to the true observation data. In such circumstances the effect of Data Association errors is benign.

• Given $Pd \approx 1$ and an association process dominated by estimation errors, correlation error produces biased estimates and an increase in apparent observation error covariance $\overline{}$ which results in an apparent distribution of observations which is non-conservative with respect to the true observation data. In this case the effect of Data Association errors is destructive leading to unstable estimation and divergent behaviour.

• Given $Pd \approx 1$ and neither estimation errors nor observation errors dominant, a linear combination of the models with coefficients $\dfrac{|P_2|}{|P_1+P_2|}$ and $1 - \dfrac{|P_2|}{|P_1+P_2|}$ applies.

Other features of data association models proposed in Chapter 6 were left as topics for future research. Namely, when $Pd \approx 1$ the expected improvement in data fusion performance against objects in groups compared with single objects and the expected degraded performance in the same situation when $Pd \neq 1$. Also left as a topic for future research was the complex interactions of these behaviours which might be observed and confirmation of the ergodic Markov process models proposed.

The use of the error square ratio, $\dfrac{|P_2|}{|P_1+P_2|}$, to represent the relationship between observation and estimation errors produces high fidelity models for the range of normalised object separations meaningful in Tracking applications of Data Association. When Track Fusion applications of Data Association are modelled, smaller normalised object separations arise and the accuracy of the models reduces. Detailed investigation of the modelling of Data Association for Track Fusion and is left as a topic for future research.

The models of Tracking and Track Fusion estimation performance were validated. In a simulation, two sensors, separated by several miles, observed, independently, a pair of distant approaching objects. An object tracking process was performed at each sensor and the resulting tracks were communicated, periodically, to a third party. Immediately after each communication, the third party performed track fusion. The entropy of the track fusion result was recorded and compared with models of the data fusion process. It was shown that:

•The model of track fusion produced acceptable correspondence with the latter stages of the simulation with an average error in the attribute standard deviations of at most 1.25%.

•During the earlier stages of the simulation, other fusion states that were not modelled dominated the simulation performance. Thus the effects of fusion states, predicted in Chapter 6 were observed in these results.

Multiple Fusion State models were applied to TV tracker sensors in the same track fusion scenario. The results were represented as an Information Map of the data fusion system. The information map proposed here may be represented alternately as a surface with the highest points corresponding to the best information or as a set of contours.

The information map had an outer stable region where there are few contours and the information quality changed slowly, two roughly semicircular inner regions where there were no contours and information quality changed slowly and a ring of close contours where the average information quality changed rapidly. The first two regions corresponded to stable fusion states and the third, the transition between them. The relationship between the states was governed by an ergodic Markov process model. Where full, 3-dimensional data could not be resolved, an ignorance model was used

which treated the missing data as uniformly distributed over the full width of the region of interest along the line of the deficiency.

The data were also represented as an Effectiveness Map. The Effectiveness Map is another representation developed for this work. It was created by applying the SoP, $\overline{H_0}$, which corresponded to the performance level at which the data fusion system was deemed to become effective. The Effectiveness Map allows the effective operating regions of a system to be visualised.

The object classification models proposed in Chapter 6 were validated. There was a model of the entropy of classification belief and another model of the distribution of classification decisions. A single sensor observed an object repeatedly and produced an independent classification each time in the form of the favoured object class. Simulations and models of this process were compared.

It was seen that the distributions of belief and decisions produced by a recursive Bayesian Object Classifier did not correspond. The belief distribution was overly optimistic and predicted a higher probability of a correct decision than was actually achieved. This occurred because the classifier was capable of choosing the wrong class with a high level of belief. Such an occurrence caused the belief distribution to improve and the decision distribution to worsen.

8   Conclusions and Future Work

8.1   Conclusions of This Thesis

In Chapter 1, the aim of this work is defined as a systematic approach to concise measurement of the effectiveness of data fusion processes. This is achieved.

In Chapter 2, the nature of information systems and situation awareness, and the role of embedded data fusion processes in providing that information, are examined. The benefits of objective prediction and evaluation of effectiveness are stated. Future application of such measures to the control and monitoring of autonomous systems is anticipated.

Also in that chapter, in section 2.8, a measure of data fusion performance is defined. Given the information requirement of some task, the measure of performance is the likely error in that information as provided by data fusion; the information error indicating a capacity to support wrong decisions and inappropriate actions. The standard of performance (SoP) is defined as the maximum level of information error which can be tolerated and the task still be carried out with confidence. The measure of effectiveness is then the probability that the SoP is achieved. *The central proposition of this thesis is the entropy of information error provides a concise expression of these quantities.*

Chapter 3 examines the data fusion process. It considers information error in terms of a region of uncertainty around a sensor observation or a fused object estimate. Candidate data associations, connecting observations and estimates about the same real world object, arise where the uncertainty regions surrounding two or more observations and estimates intersect. After associated observations and estimates are fused, the new uncertainty lies within the intersection. Well known algorithms for associating and fusing data are discussed.

The subject of Chapter 4 is information entropy. The history of entropy is considered; first encountered as a measure of the disorder in an ensemble of physical

particles and eventually adopted as a measure of the information content of stored, transmitted or computed data. Standard results of information theory are given in this chapter. At the end of that chapter, in section 4.5, *a new analytic approach to the calculation of combined entropy of continuous and discrete information is defined*.

Chapter 5 considers the properties of entropy in the context of data fusion and situation awareness. Changes in spatial "point-of-view" do not affect information entropy but change in viewing scale does. The effect of a change viewing time depends on whether the system being observed is stochastic or deterministic and on whether new information is available to a later observer. In stochastic systems, uncertainty grows during any period where no new observation is made and entropy increases accordingly. In a deterministic system, entropy is static during any period without new observations. In either case, relevant new observations cause entropy to decrease. It is also noted in Chapter 5 that entropy discriminates between samples with identical covariance matrices drawn from different probability distributions. Thus, the entropy measure discriminates shape as well as spread of the distribution of information error.

In Chapter 6 the ways in which simulations of the data fusion processes can be constructed are described. A simulation is defined here as a mixture of real system elements and abstractions giving the appearance of the complete and full system in operation. The performance and effectiveness of a system in simulation are measured as if the real process were taking place. The simulations used for this work comprised the real fusion process, sometimes organised into simplified architectures, receiving data from abstractions of sensors operating in an abstracted environment.

The aims of this work include prediction as well as measurement and evaluation and, to predict the effectiveness of data fusion, a set of interrelated models is produced. In the context of this work a model is defined as an abstraction of the entire system and its environment, where no attempt is made to give the appearance of a real system in

Page 229

operation and only the outcome (its performance and effectiveness) is predicted. The models too are described in Chapter 6. They represent the processes of data association, estimation and maintenance of the situation awareness "picture". –

In section 6.2.1, *a new Data Association model is proposed* which represents the clustering process that groups together observations of the same real-world object. The model is novel because it operates in a space where distances are normalised with respect to the distribution of relative errors between new information and information already accumulated. The main advantages of this compared with known approaches are greater accuracy and applicability to a wider range of situations. The model predicts the probability that the estimate for a particular real-world object will be updated with true observations - the alternatives being false observations caused by clutter or observations from other nearby objects. Given this probability, the bias (apparent change in the mean of the observations) and the distortion (the apparent change in the spread of the observations) are calculated.

Estimation process models then predict the information quality obtained when the observations in associated groups are fused. Models are given for Tracking, for Track Fusion and for Object Classification. In addition to these, the effect, on situation awareness, of the absence of an estimate is considered.

The tracking models, described in section 6.2.2, predict the parameters of a track's posterior error distributions and use these to calculate entropy. The well known steady-state solution of the matrix Riccati equation is used, but *new formulations of the solution as a series expansion and a nested computation are offered.*

In sections 6.2.3 and 6.2.4 respectively, *new models of Track Fusion and Object Classification, are defined directly in terms of the propagation of information entropy as new observations are received and fused.* The concept of *a quantifiable*

information penalty of ignorance is proposed in section 6.2.5 and a model is offered

for the information error of having no estimate.

Given the statistics of associated sets of observations and the resultant fused

information quality, our model of the data fusion process is incomplete. Automatic

decision processes operate within data fusion to create, maintain and delete elements of

the situation awareness "picture". A particular sequence of observations may lead to an

estimate of predictable information quality, but is the sequence sufficiently coherent for

that estimate to be maintained? A pair of nearby objects may cause quantifiable

disruption to the data association and estimation processes, but does this lead to a simple

reduction in fused information quality or a divergent estimate irrevocably separated from

the true object?

In section 6.2.6, *the new concept of fusion states is proposed* to account for

step changes in information quality and mixtures of system behaviours that arise in these

and similar circumstances and *a new model to predict their influence on fused

information quality is offered.* The effects on data fusion of these errors and distortions

are modelled as an ergodic Markov process. Using such a model it is possible to predict

the probability of each state in which an object estimate could be found.

Chapter 7 gives the results of a series of case studies performed to validate these

models. They confirm that the models are indicative of the performance and effectiveness

of data fusion. In all cases the modelled performance predicted that of a detailed Monte

Carlo simulation of the same situation. When the models were compared with a real

system, the results were promising but inconclusive. Equipment problems during the trial

were not allowed for in the modelling and resulted in differences between the modelled

and measured performance. A detailed discussion of all the results is given in Chapter 7.

Throughout this work information entropy is found to be a good scalar measure

of the spread of errors in fused information. As such it is capable of summarising

numerous and complex uncertainties into a single value. When effectiveness is evaluated, information entropy has the advantage of conciseness. Greater precision could be achieved by examining different aspects of performance and effectiveness separately, but at the cost of a more fragmented and possibly qualified statement of effectiveness.

Entropy alone is not sensitive to bias errors. However, when bias is reflected in the Markov chain Fusion State models, it is possible to predict effects such as discontinuities in estimation which are symptomatic of this type of error. During any break in estimation, the "lost" object is represented by the information entropy of ignorance and thus the degraded information quality is evident.

Many of the examples and the references used throughout this thesis relate to Radar tracks of objects which were usually aircraft. However, this work is relevant to estimation of the world-line of any object in space and time. For example, it is common for similar tracking techniques to those used in Radar surveillance of aircraft to be applied to financial forecasting. This work has applications outside the field from which the examples are drawn.

8.2   Further Research

The tracking models described in Chapter 6 predict optimal data fusion process performance where systems are approximately linear and errors approximately white and Gaussian. Models of tracking for highly non-linear, non-Markov and non-Gaussian systems will be left as topics for future research.

In the course of this work we concentrate on a Bayesian system of beliefs as an approach to Object Classification. The alternative Dempster - Shafer Evidential Reasoning approach might offer a fruitful area for further research [20] [74].

The work in this thesis was concerned with the entropy of the equilibrium mixture of states as an expression of the likely behaviours that could arise from a data fusion process in a given situation. Ekroot and Cover defined the entropy of a Markov

trajectory between two states of a Markov process as an entropy measure of the process or path by which a future state of affairs could arise given the current known starting point [23]. These ideas in combination could be of particular interest in future autonomous systems where an automatic controller might select actions, using entropy measures, to promote desirable outcomes and avoid undesirable ones.

The subject of Data Association and the models proposed in this thesis should be investigated further. The work done to date has just scratched the surface of the subject and investigation of these simple models suggests complex (but predictable) behaviours are present. In situations when $Pd \neq 1$ we predict complex stochastic processes in action. The applicability of the ergodic Markov model to these processes needs to be confirmed.

This thesis offers models for nearest neighbour and optimal assignment data association algorithm performance. The question of whether the approach proposed can be extended to conveniently represent other algorithms (in particular, MHT performance and JPDAF performance) will be left as a topic for future research [2] [69].

The study of the ICIDS laboratory based data fusion technology demonstrator system needs to be repeated or a similar study with some other data fusion system must be carried out. Demonstration of correspondence between a model and a detailed Monte Carlo simulation provides a measure of confidence but the only way to achieve the levels of confidence necessary if investment decisions are to be made on the basis of these tools is to demonstrate correspondence with a real system.

8.3   Potential Future Applications

The prospect exists to use MoPs and MoEs within systems to optimise some aspect of their own behaviour. The system would be constructed to incorporate a model of effective behaviour and a model of the processes it executes. Any actions under internal control would then be selected and initiated to maximise effectiveness.

Systems are already emerging which operate under (automatic) management. It is possible to construct a Sensor system in this way so that tasking is performed to maximise the effectiveness of the system. The effectiveness model has to date been based on the implementer's judgement. This work will provide an analytical basis for the development of such a model and a means of predicting whether or not effective operation will be achievable in a realistic scenario.

It will be possible to create a system capable of monitoring its own information gathering performance. Based on the reported situation, a model of expected system behaviour would be used to predict a performance level. If this level were not achieved equipment malfunction might be indicated.

There is current interest in self learning systems [34]. Such a system would have some model of desired and undesired behaviour and would modify itself iteratively in an attempt to optimise its own behaviour. The great difficulty with these systems is the creation of concise, verifiable behavioural models which are based in theory. The techniques described here might be applied for this purpose.

Looking further ahead, Conscious systems would be implemented in applications which required some robot agent to carry out actions autonomously with little or no immediate supervision [57]. The agent would supervise itself by comparing the outcomes of its actions with simulation or model of itself. Any significant divergence between the model and the real world would trigger remedial actions or alarms. Again there is a requirement for reliable, concise modelling founded in theory and this presents a possible application for this work.

Appendix - Glossary of Acronyms and Abbreviations

ACM	Association for Computing Machinery
AFCEA	American Forces Communications and Electronics Association
AGARD	Advisory Group for Aerospace Research and Development
ARIMA	Auto-regressive Integrated Moving Average (time series)
ARMA	Auto-regressive Moving Average (time series)
CBA	Cost/benefit analysis
C^2	Command and Control
C^3	Command, Control and Communications
CI	Covariance Intersection
C^3I	Command, Control, Communications and Intelligence
CIWG	Covariance Intersection Working Group
COBP	Code of best practice
CRLB	Cramer-Rao Lower Bound
DKf	Distributed Kalman Filter (tracker)
DLM	Dynamic Linear Model (time series)
EKf	Extended Kalman Filter (tracker)
ESA	Electronically Steered Array (radar)
ESM	Electronic Support Measures
ICIDS	Investigation into Communications Issues in Decentralised Systems
IEE	Institute of Electrical Engineers
IEEE	Institute of Electrical and Electronic Engineers
IMM	Interacting Multiple Models (tracker)
JPDAF	Joint Probabilistic Data Association Filter
Kf	Kalman Filter (tracker)

MAP	Maximum a Posteriori (estimator)
MCES	Modular Command and Control Evaluation Structure
MHT	Multiple Hypothesis Tracking
MIS	Management Information Systems
MMI	Man-machine Interface
MMSE	Minimum Mean-square Estimator
MoE	Measure of Effectiveness
MoFE	Measure of Force Effectiveness
MoP	Measure of Performance
MVGD	Multivariate Gaussian Distributed
NATO	North Atlantic Treaty Organisation
NN	Nearest Neighbour
NNSF	Nearest Neighbour Standard Filter
ORSA	Operations Research Society of America
Pd	Probability of Detection
PDAF	Probabilistic Data Association Filter
Pfa	Probability of False Alarm
RCS	Radar Cross-section
SKIDS	Signal and Knowledge Integration with Decisional Control for Multi-sensory Systems
SoP	Standard of Performance
SPIE	The International Society for Optical Engineering
SVD	Singular Value Decomposition
TSF	Track Splitting Filter

References

[1]   J. H. Atkinson, (1990), *Modelling and Simulation of the Test and Evaluation Process*, USA: Simulation, pp. 127.

[2]   Y. Bar-Shalom and T. E. Fortmann, (1988), *Tracking and Data Association*, San Diego, CA: Academic Press.

[3]   A. L. Barker, D. E. Brown and W. N. Martin, (1990), *A Neural Implementation of a Data Association Algorithm*, ORSA Journal on Computing, Operations Research Society of America, pp. 100.

[4]   J. J. Baroudi, M. H. Oslon and B. Ives, (1986), *An Empirical Study of the Impact of User Involvement on System Usage and Information Satisfaction*, New York: Communications of the ACM, Vol 24, No 3, pp. 232.

[5]   D. A. Bell, (1968), *Information Theory*, London: Pitman.

[6]   A. Beltran (M. Brown - translator), (1982), *Rock Art of the Spanish Levant*, New York: Cambridge University Press.

[7]   J. V. Black and C. M. Reed, (1996), *A Hybrid Parametric, Non-Parametric Approach to Bayesian Target Tracking*, London: IEE, Target Tracking and Data Fusion, IEE Digest No: 96/253, Paper No. 5.

[8]   S. S. Blackman, (1986), *Multiple-Target Tracking with Radar Application*, Norwood MA: Artech House.

[9]   S. S. Blackman, (1990), *Association and Fusion of multiple Sensor Data*, Mulltitarget-Multisensor Tracking: Advanced Applications (Y. Bar-Shalom editor), Norwood MA: Artech House, pp. 187.

[10]   H. A. P. Blom and Y. Bar-Shalom, (1988), *The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients*, New York: IEEE Transactions on Automation Control, Vol 33, No. 8, pp. 780.

[11]   F. Bourgeois and J. C. Lassalle, (1971), *An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices*, New York: Communications of the ACM, vol 14, pp. 802.

[12]   R. Burke, (1992), *Project Management, Planning and Control - second edition*, New York: John Wiley and Sons.

[13]   M. J. Caputi, (1995), *A Necessary Condition for the Effective Performance of the Multiple Model Adaptive Estimator*, New York: IEEE Transactions on Aerospace and Electronic Systems, Vol. 31 No. 3, pp. 1132.

[14]   The Covariance Intersection Working Group, (1998), *A culminating Advance in the Theory and Practice of Data Fusion, Filtering and Decentralised Estimation*, http://ait.nrl.navy.mil/people/uhlmann/covint.html.

[15]   S. A. W. Cyrus, (1991), *Measuring the Effectiveness of Information Systems*, Monterey, CA: US Naval Postgraduate School.

[16]   E. Daeipour and Y. Bar-Shalom, (1995), *An Interacting Multiple Model Approach for Target Tracking with Glint Noise*, New York: IEEE Transactions on Aerospace and Electronic Systems, Vol. 31 No. 2, pp. 706.

[17]   L Dams, (1996), *Prehistoric Art*, Grolier Multimedia Encyclopaedia, Grolier Electronic Publishing.

[18]   R. H. Deaves et. al., (1998), *Simulation Based Evaluation of Communications Management Within Battlespace Scenarios*, SPIE Orlando: 12th Annual International Symposium on Aerospace/Defence Sensing, Simulation and Control, Ref. 3374/07, pp. 76.

[19]   R. H. Deaves, D. Nicholson, P. Greenway and D. R. Bull, (1997), *Evaluation of Communications Management in a Simulated Decentralised Tracking System*, Orlando: SPIE, 11th Annual International Symposium on Aerospace/Defence Sensing, Simulation and Control, 3068(30).

[20]   A. P. Dempster, (1968), *A Generalisation of Bayesian Inference*, London: Journal of the Royal Statistical Society, Series B, Vol 30, pp. 205.

[21]   N. V. Duffy, (1996), *Thermodynamics*, Grolier Multimedia Encyclopaedia, Grolier Electronic Publishing.

[22]   J. A. Edminster, (1979), *Theory and Problems of Electromagnetics*, New York: McGraw-Hill.

[23]   L. Ekroot and T. J. Cover, (1993), *The Entropy of Markov Trajectories*, New York: IEEE Transactions on Information Theory, Vol. 39, No. 4, pp. 1418.

[24]   M. Feder and N. Merhav, (1994), *Relations Between Entropy and Error Probability*, New York: IEEE Transactions on Information Theory, Vol. 40, No. 1, pp. 259.

[25]   K. Fielden, (1996), *Cuniform*, Grolier Multimedia Encyclopaedia, Grolier Electronic Publishing.

[26]   R. A. Fisher, (1921), *On the Mathematical Foundations of Theoretical Statistics*, London: Philosophical Transactions of the Royal Society of London, Sec A(222), pp. 309.

[27]   D. F. Galletta and A. L. Lederer, (1989), *Some Cautions on the Measurement of User Information Satisfaction*, Georgia State University: Decision Sciences Journal, Volume 20, No. 3, pp. 419.

[28]   I. J. Gelb and R. M. Whiting, (1996), *Evolution of Writing Systems*, Grolier Multimedia Encyclopaedia, Grolier Electronic Publishing.

[29]   P. E. Girard, (1989), *A Function-Based Definition of C² Measures of Effectiveness*, The Science of Command and Control Volume II (S. E. Johnson and A. H. Levis editors), USA: AFCEA International Press, pp. 112.

[30]   A. M. Haimovich, J. Yosko, R. J. Greenberg, M. A. Parisi and D. Becker, (1993), *Fusion of Sensors with Dissimilar Measurement/Tracking Accuracies*, New York: IEEE Transactions on Aerospace and Electronic Systems, Vol. 29, No. 1, pp. 245.

[31]   R. Harris, (1986), *The Origin of Writing*, London: Duckworth.

[32]   P. J. Harrison and C. F. Stevens, (1976), *Bayesian Forecasting (with discussion)*, London: Journal of the Royal Statistical Society B, 38, pp 205.

[33]   R. V. L. Hartley, (1928), *Transmission of Information*, Murray Hill, NJ: Bell Systems Technical Journal No 7, pp. 535.

[34]   J. N. H. Heemskerk and N. Sharkey, (1996), *Learning Subsumptions for an Autonomous Robot* , London: IEE, Self Learning Robots, IEE Digest No: 96/026, Paper No. 8.

[35]   S. A. Horvanessian, (1988), *Introduction to Sensor Systems*, Norwood MA: Artech House.

[36]   B. Houston and J.Kraus, (1989), *Advanced data correlation techniques*, Department of Physics, University of Durham.

[37]   T. O. Jackson, (1993), *Neural Networks for Data Association Tasks*, Hull, Humberside: British Aerospace Internal Report BAe-BSE-FE-NNS-300086

[38]   R. E. Kalman, (1960), *A New Approach to Linear Filtering and Prediction Problems*, Transactions ASME: J. Basic Engineering, Vol. 82, pp. 34.

[39]   P. Keen, (1981), *Value Analysis: Justifying Decision Support Systems*, Minnesota, Dublin and Melbourne: MIS Quarterly, pp. 1.

[40]   M. G. Kendall, A. Stuart and J. K. Ord, (1987), *Kendall's Advanced Theory of Statistics*, Vol I, 5th Edition, London, Griffin and Co.

[41]   M. G. Kendall, A. Stuart and J. K. Ord, (1991), *Kendall's Advanced Theory of Statistics*, Vol II, 5th Edition, London: Edward Arnold.

[42]   M. G. Kendall, A. Stuart and J. K. Ord, (1983), *The Advanced Theory of Statistics*, Vol III, 4th Edition, London, Griffin and Co.

[43]   D. J. Kershaw and R. J. Evans, (1996), *A Contribution to Performance Prediction for Probabilistic Data Association Tracking Filters*, New York: IEEE Transactions on Aerospace and Electronic Systems, Vol. 32 No. 3, pp. 1143.

[44]   M. M. Kharbouch and D. P. Atherton, (1990), *Performance Evaluation of Target Trackers*, School of Engineering & Applied Sciences, University of Sussex, Falmer, Brighton

[45]   G. A. Korn and T. M. Korn, (1968), *Mathematical Handbook for Scientists and Engineers* - second edition, New York: McGraw-Hill.

[46]   B. P. Lahti, (1968), *Random Signals and Communication Theory*, Scranton, Pennsylvania: International Textbook Company.

[47]   R. J. Linn, D. L. Hall and J. Llinas, (1991), *A Survey of Multi-Sensor Data Fusion Systems*, Orlando: SPIE Vol. 1470, Data Structures and Target Classification, pp. 13.

[48]   J. Llinas, (1992), *System Engineering, Design & Development*, Lecture Notes, London: Advanced Technology International Ltd.

[49]   Longman, (1996), *The Dictionary of the English Language*, Infopedia, SoftKey International Inc.

[50]   D. T. Magill, (1965), *Optimal Adaptive Estimation of Sampled Stochastic Processes*, New York: IEEE Transactions on Automation Control, Vol. AC-10, pp. 434.

[51]   J. Manyika and H. F. Durrant-Whyte, (1994), *Data Fusion and Sensor Management: a decentralised information-theoretic approach*, West Sussex: Ellis Horwood Limited.

[52]   A. Mason, (1984), *Computers and Productivity*, Public Productivity Review, Volume 8, No 1, pp. 70.

[53]   J. C. Maxwell, (1871), *Theory of Heat*, London: Longmans, Green.

[54]   P. S. Maybeck, (1979), *Stochastic Models, Estimation and Control*, Vol. I, San Diego, CA: Academic Press.

[55]   S. Mori, K.-C. Chang and C.-L. Chong, (1992), *Performance Analysis of Optimal Data Association with Applications to Multiple Target Tracking*, Mulltitarget-Multisensor Tracking: Applications and Advances (Y. Bar-Shalom editor), Volume II, Norwood MA: Artech House, pp. 183.

[56]   M. Moruzzis, N. Colin and G. Milhem, (1995), *Applications of Multisensor Data Fusion to Target Recognition*, Paris: NATO, AGARD Conference Proceedings No. 557, Tactical Aerospace C3I in the coming Years, Paper No. 15.

[57]   M. Nilsson, (1996), *Toward Conscious Robots*, London: IEE, Self Learning Robots, IEE Digest No: 96/026, Paper No. 12.

[58]   C. A. Noonan, (1993), *Co-operative Sensor Data Fusion Study - Proposed Algorithms*, Warton Lancashire: British Aerospace Internal Report BAe-WSE-RP-RDB-DF-086

[59]   C. A. Noonan and K. J. Orford, (1996), *Entropy Measures of Multi-Sensor Fusion Performance* , London: IEE, Target Tracking and Data Fusion, IEE Digest No: 96/253, Paper No. 15.

[60]   C. A. Noonan, (1996), *Sensor Data Fusion for Air to Air Situation Awareness Beyond Visual Range* , Paris: NATO, AGARDograph 337 Multi-sensor Multi-target Data Fusion, Tracking and Identification Techniques for Guidance and Control Applications, pp 268.

[61]   C. A. Noonan, (1997), *Sensor Fusion Effectiveness and System Life-Cycle Management* , London: ERA Technology, Avionics 97 Conference, Paper No. 10.5.

[62]   C. A. Noonan and M. Pywell, (1997), *Aircraft Sensor Data Fusion: an Improved Process and the Impact of ESM Enhancements*, Paris: NATO, Agard Conference on Data Fusion, Lisbon, Paper No. 2.

[63]   N. Pal and S. Pal, (1991), *Entropy, a New Definition and its Applications*, New York: IEEE Transactions on Systems Management and Cybernetics, 21(5), pp. 1260.

[64]   R. Popoli, (1992), *The Sensor Management Imperitive*, Mulltitarget-Multisensor Tracking: Applications and Advances (Y. Bar-Shalom editor), Volume II, Norwood MA: Artech House, pp. 325.

[65]   W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, (1992), *Numerical Recipes in C - the art of scientific computing*, Second Edition, the Press Syndicate of the University of Cambridge

[66]   B. S. Y. Rao, H. F. Durrant-Whyte and J. A. Sheen, (1991), *A Fully Decentralised Multi-Sensor System for Tracking and Surveillance*, Robotics Research Group, University of Oxford: Report No. OUEL 1886/91.

[67]   C. R. Rao, (1965), *Linear Statistical Inference and its Applications*, New York: John Wiley.

[68]   K. Reichard and T Stephenson, (1997), *Teach Yourself Netscape Communicator 4.0*, New York: MIS Press.

[69]   D. B. Reid, (1979), *An Algorithm for Tracking Multiple Targets*, New York: IEEE Transactions on Automatic Control, Vol. AC-24 No. 6, pp 843.

[70]   The Research and Technology Organisation (1999), *Code of Best Practise (COBP) on the Assessment of C2*, Paris: NATO

[71]   G. S. Rushbrooke, (1949), *Introduction to Statistical Mechanics*, Oxford University Press.

[72]   R. K. Saha, (1996), *Track-to Track Fusion With Dissimilar Sensors*, New York: IEEE Transactions on Aerospace and Electronic Systems, Vol. 32 No. 3, pp. 1021.

[73]   D. Sengupta and R. A. Iltis, (1988), *Neural Solution to the Multi-Target Tracking Data Association Problem*, New York: IEEE Transactions on Aerospace and Electronic Systems Volume AES-25 No. 1, pp. 96.

[74]   N. E. Shafer, (1976), *A Mathematical Theory of Evidence*, New Jersey: Princeton University Press.

[75]   C. Shannon, (1948), *A Mathematical Theory of Communication*,Murray Hill, NJ: Bell Systems Technical Journal No 27, pp. 393, 623

[76]   R. A. Singer, (1970), *Estimating Optimal Tracking Filter Performance for Manned Manoeuvring Targets*, New York: IEEE Transactions on Aerospace and Electronic Systems, Vol. AES 6, No. 4, pp. 473.

[77]   M. I. Skolnik, (1962), *Introduction to Radar Systems*, Tokyo: Mc Graw - Hill.

[78]   R. Sweet, (1989), *The MCES and the Search for Generic Measures*, The Science of Command and Control, Vol II (S. E. Johnson and A. H. Levis editors), USA: AFCEA International Press, pp. 106.

[79]   B. J. Thomas, (1997), *The Internet for Scientists and Engineers* - third edition , WA: SPIE

[80]   G. van Keuk, (1995), *Multihypothesis Tracking with Electronically Scanned Radar*, New York: IEEE Transactions on Aerospace and Electronic Systems, Vol. 31 No. 3, pp. 916.

[81]   H. L. Van Trees, (1989), *C^3 Systems Research: A Decade of Progress*, The Science of Command and Control Volume II (S. E. Johnson and A. H. Levis editors), USA: AFCEA International Press, pp. 24.

[82]   E. Waltz and J Llinas, (1990), *Multisensor Data Fusion*, Norwood MA: Artech House.

[83]   M. West and J. Harrison, (1997), *Bayesian Forecasting and Dynamic Models*, second edition, New York: Springer.

[84]   A. E. R. Woodcock and J.T. Dockery, (1988), *Models of Combat with Embedded C^2. I: Catastrophe Theory and Lanchester Equations*, International CIS Journal, Vol. 2, (3), pp. 34.

[85]   P. M. Woodward, (1953), *Probability and Information Theory, with Applications to Radar*, London, Pergamon Press.

[86]   M. Wu and W. J. Fitzgerald, (1995), *Analytical Approach to Change Point Detection in Laplacian Noise*, London: IEE Proceedings - Visual Image Signal Processing Vol. 142 No. 3, pp. 174.