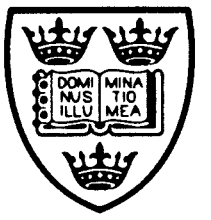
An Information-Theoretic Approach to Data Fusion and Sensor Management

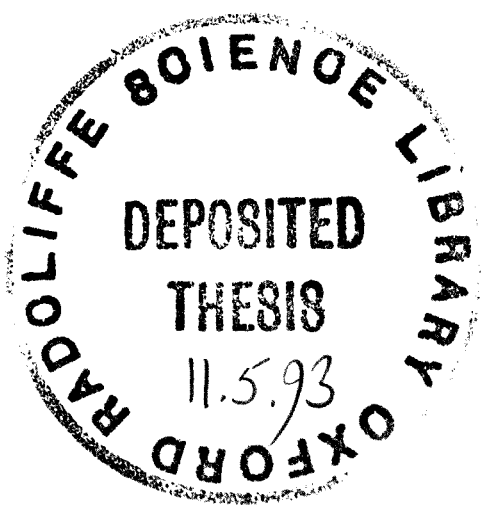
by

James Munyaradzi Manyika



Department of Engineering Science
University of Oxford

Hilary Term, 1993



This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfillment of the requirements for the degree of Doctor of Philosophy. The thesis is entirely my own work, and except where otherwise stated, describes my own research.

James M. Manyika, Balliol College, Oxford

James Munyaradzi Manyika
Balliol College

Doctor of Philosophy
Hilary, 1993

An Information-Theoretic Approach to Data Fusion and Sensor Management

Abstract

The use of multi-sensor systems entails a *Data Fusion* and *Sensor Management* requirement in order to optimize the use of resources and allow the synergistic operation of sensors. To date, data fusion and sensor management have largely been dealt with separately and primarily for centralized and hierarchical systems. Although work has recently been done in distributed and decentralized data fusion, very little of it has addressed sensor management. In decentralized systems, a consistent and coherent approach is essential and the *ad hoc* methods used in other systems become unsatisfactory.

This thesis concerns the development of a unified approach to data fusion and sensor management in multi-sensor systems in general and decentralized systems in particular, within a single consistent information-theoretic framework. Our approach is based on considering information and its gain as the main goal of multi-sensor systems. We develop a probabilistic information update paradigm from which we derive *directly* architectures and algorithms for decentralized data fusion and, most importantly, address sensor management. Presented with several alternatives, the question of how to make decisions leading to the best sensing configuration or actions, defines the management problem. We discuss the issues in decentralized decision making and present a *normative* method for decentralized sensor management based on information as expected utility. We discuss several ways of realizing the solution culminating in an iterative method akin to bargaining for a general decentralized system. Underlying this is the need for a good sensor model detailing a sensor's physical operation and the phenomenological nature of measurements *vis-a-vis* the probabilistic information the sensor provides. Also, implicit in a sensor management problem is the existence of several sensing alternatives such as those provided by agile or multi-mode sensors. With our application in mind, we detail such a sensor model for a novel *Tracking Sonar* with precisely these capabilities making it ideal for managed data fusion. As an application, we consider vehicle navigation, specifically localization and map-building. Implementation is on the OxNav vehicle (JTR) which we are currently developing. The results show, firstly, how with managed data fusion, localization is greatly speeded up compared to previous published work and secondly, how synergistic operation such as sensor-feature assignments, hand-off and cueing can be realised decentrally. This implementation provides new ways of addressing vehicle navigation, while the theoretical results are applicable to a variety of multi-sensing problems.

Acknowledgements

My most sincere thanks and gratitude go first and foremost to my supervisor Dr. Hugh Durrant-Whyte who through his limitless energy and enthusiasm gave me the motivation and encouragement to carry out this research and also proved to be an inexhaustible source of ideas and information. I would also like to thank Prof. Mike Brady for his interest in my work and for the numerous fruitful discussions and suggestions. My gratitude also goes to my closest colleagues, Tom Burke, Stewart Grime, Michael Stevens, Peter Ho, Billur Barshan and Tim Berg for their many useful comments, constructive criticism and support. My gratitude also goes to the OxNav crowd for creating *Joey* and in particular Ian Treherne for building the Tracking Sonars. I would also like to thank John Leonard and Bobby Rao who laid the foundations for much of this work, for the numerous fruitful discussions and helpful comments. Thanks are due to Dr. Paul Schenker of NASA Jet Propulsion Labs, and Dr. Alec Cameron of Philips from whom, through visits and discussions, I benefited immensely. The Robotics motley crew in particular, Arthur, Charlie, Chris, Steve, Dave, Doug, Huosheng, Ian, Mariano, Martin, Simukai and Yang contributed to making this an enjoyable experience. I owe an immense debt to my "home-girl" Simukai and to Pete for their assistance in proof-reading this thesis.

I must thank the many members of my three colleges, Keble, St Hugh's and Balliol for making my Oxford experience extremely varied and enjoyable. Thanks go to my many friends especially, David, Liz, Michele, and Phil for their companionship, tolerance and understanding throughout my time at Oxford. I would like to thank my mother and Veldore Young for their ever present love and support. Most importantly, I wish to pay tribute to my father James M.D. Manyika who died as I wrote this. I miss him and thank him for everything.

Finally I would like to thank the Rhodes Trustees who through the generosity of a Rhodes Scholarship funded my study at Oxford. In addition, thanks are due to St Hugh's College for the grant of a Senior Scholarship and to the Fellows of Balliol College for election to a Junior Research Fellowship.

This work is dedicated to my Mother and to the memory of my Father and all that he stood for.

Contents

List of Figures

Nomenclature

Notation

$(\cdot)^T$	Matrix transpose
$(\cdot)^{-1}$	Matrix inverse
$\bar{(\cdot)}$	Mean of $(\cdot)$
$\hat{(\cdot)}$	Estimate of $(\cdot)$
$\tilde{(\cdot)}$	Partial estimate of $(\cdot)$
$E\{\cdot\}$	Expected value of $\{\cdot\}$
$a \triangleq b$	a is defined as b
$\{(\cdot)\}$	Set whose elements are $(\cdot)$
$< \cdots >$	Ordered set
$(\cdot)(k)$	$(\cdot)$ at time k
$(\cdot)(k \mid l)$	$(\cdot)$ at time k given l

General Symbols

The following symbols are used consistently throughout this thesis. However, in some chapters there are some locally defined symbols whose scope is limited to those chapters only.

Symbol	Interpretation
k	Time step
$\mathcal{P}$	Probabilistic set
$p(\cdot)$	Probability distribution where $p \in \mathcal{P}$
λ	Likelihood function
Λ	Likelihood vector
$\mathcal{X}$	State space
$\mathbf{x}$	State vector $\mathbf{x} \in \mathcal{X}$
X_l	lth element in a discrete state vector
$\mathbf{P}$	State covariance matrix
$\mathbf{y}$	Information state vector
$\mathcal{Z}$	Observation space
$\mathbf{z}$	Observation vector

General Symbols(cont)

Symbol	Interpretation
$\mathbf{Z}^k$	Set of observation vectors $\mathbf{z}$ up to time k
$\mathbf{Z}_j^k$	Set of observation vectors up to k for sensor j
$\mathcal{V}$	Observation noise space
$\mathbf{v}$	Observation noise vector
$\mathbf{R}$	Observation noise covariance
$\mathcal{M}$	General observation model
$\{M\}$	Set of observable parameters
M_l	lth observable parameter
$\mathbf{H}$	Linear observation matrix
$\mathbf{h}[\cdot]$	Non-linear observation model
$\mathbf{w}$	State transition noise
$\mathbf{Q}$	State transition noise covariance
$\mathbf{F}$	State transition matrix
$\mathbf{f}[\cdot]$	Non-linear state transition model
$\mathbf{u}$	Control input
$\mathcal{N}$	Set of information sources (sensors)
N	Number of information sources (sensors) in system
ϵ	Generalized error symbol
$s(\cdot)$	Score function
∇	Vector Gradient
$\mathcal{A}$	Action set
a_l	lth action
a^{ir}	sensor i's rth preference action
$U(\cdot)$	Utility function
$L(\cdot)$	Loss function
$\beta(\cdot)$	Expected Utility
$\mathcal{B}(\cdot)$	Total expected utility
p_l	Outcome probability distribution function for action a_l
$\rho_l, \{p\}_l$	Set of probabilistic outcomes corresponding to action a_l
$\{\rho\}_l$	Set of probabilistic outcomes for *all* sensors in $\mathcal{N}$ corresponding to a_l
$\succeq$	Preferred to or equal in preference to
$\Rightarrow$	Implies
$h(\cdot)$	Entropy
$i(\cdot)$	Mutual information
$\mathbf{J}$	Fisher information
$\mathcal{I}(\cdot)$	General information metric
$\mathbf{I}$	Entropy based posterior information metric
$\mathbf{i}$	Entropy based likelihood information metric
$\tilde{\mathbf{I}}$	Entropy based partial information metric
$\mathcal{T}$	Target or feature set
r	Range to feature

General Symbols(cont)

Symbol	Interpretation
σ_r^2	Range variance
θ	Bearing to feature
σ_θ^2	Bearing variance
δ	Differential path length
Δ	Differential time of flight
β_{RCD}	Angular width of Region of Constant Depth
r_{RCD}	Range of Region of Constant Depth
β_{min}	Angular criteria
ϵ_r	Range error criteria
ζ	Sensor control parameters
d	Seperation (base-line) of Polaroid devices on Tracking Sonar
σ_δ^2	Observation noise variance in differential estimator filter
σ_u^2	State transition noise in differential estimator filter

Abbreviations

AEP	Asymptotic equipartition property
CCD	Charged coupled device
DKF	Decentralized kalman filter
DSN	Distributed Sensor Networks
JPDAF	Joint probabilistic data association filter
KF	Kalman filter
MAP	Maximum a posteriori estimate
ML	Maximum likelihood estimate
MMSE	Minimum mean square error estimate
OxNav	Oxford Navigator research vehicle
PDF	Probability distribution function
PDAF	Probabilistic data association filter
RCD	Region of constant depth
R	Receiver
T	Transmiter
TOF	Time of flight

Where is the life we have lost in living?
Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?

- T.S. Eliot (The Rock 1934)

Chapter 1

Introduction

In sensing, the problem is no longer one of shortage of information but one of making sense of diverse and vast amounts of it. An intuitive approach is to seek and extract only that information which is necessary and relevant for the task at hand. Therefore, this work is very much about actively seeking "...the knowledge that we have lost in information...".

1.1 An Overview of Multi-Sensor Systems

Recent years have witnessed a proliferation of multi-sensor systems. This stems from the following; despite advances in sensor technologies and the myriad computational methods and algorithms aimed at extracting as much information as possible from a given sensor, the irrefutable fact remains, no single sensor is capable of obtaining *all* the required information *reliably* at all times in varying environments.

Sensing is used extensively in robotics to tackle the problem of perception. Sensors exploit physical phenomena to measure quantities. A sensor is considered appropriate to a task when a relationship exists between the measured quantity and the state of nature. The precision with which this relation is known depends on how well understood the measurement is in as far as it relates to the state. **Models** or descriptions of device physics are inherently only approximations owing to our lack of complete understanding of the principles governing the actual measurement. This is exacerbated in some sensors

due to incomplete geometric models for the interpretation of data. Sensor measurements are always **uncertain** and, occasionally, spurious and incorrect. This, coupled with the occasional **failure** of sensor devices, greatly compromises reliability. Spatial and physical limitations of the devices mean that only **partial information** can be provided by a single sensor. This results in a sensor having limited capabilities to resolve ambiguities and to provide consistent descriptions of the sensed environment.

The perception requirements of complex robotics systems necessitate the measurement of diverse and complementary quantities under various constraints. Such functionality far exceeds the repertoire of any single sensor. Motivated by biological systems, intelligent robotic systems make use of a multiplicity of sensors in order to extract as much information as possible about a sensed environment. Multi-sensor systems aim to overcome the shortcomings of single sensors through:

- **Redundancy.** It is well known that redundancy reduces uncertainty. This can be appreciated from the fact that for two sensors, the signal relating to the measured quantity is correlated whereas the uncertainty associated with each sensor tends to be uncorrelated. Also, redundancy is desirable if sensor failure is anticipated so that system performance is degraded gracefully.

- **Diversity and Complementarity.** Physical diversity is based on using different sensor technologies together and spatial diversity offers differing viewpoints of the sensed environment. Such diversity reduces uncertainty and is invaluable in resolving ambiguities. Complementarity results if the sensor suite is made up of sensors each of which observes a subset of the environment state space.

Examples of such systems are that described by Mitchie and Aggarwal [102] which obtains

complimentary information from visual, thermal and range sensors and that described by Flynn [57] which combines sonar and infra-red sensors. The literature is replete with examples, such as the work by Allen and Bajcsy [3] using vision and touch, Elfes [51] using sonar and stereo range data, Nandhakumar and Aggarwal [105] using thermal and visual images, Terzopolous [131] using spatially diverse visual information. Such trends towards the use of multi-sensor systems are reported in the surveys by Giralt [60] and Luo [93] and that by Durrant-Whyte [48]. In the development and use of multi-sensor systems, the following issues arise:

1. How can the diverse and sometimes conflicting information be combined in a consistent and coherent manner and the requisite states inferred from it?

2. How can such systems be optimally configured, utilised and controlled to provide the required information in often dynamic environments in the best possible manner?

The answer to these questions defines the *Sensor Data Fusion* problem.

1.2 Data Fusion and Sensor Management

What sensor data fusion encompasses depends on how it is defined. Definitions vary in extent but the theme is the same as these representative ones show:

> "Data fusion is the process by which data from a multitude of sensors is used to yield an optimal estimate of a specified state vector pertaining to the observed system." Richardson and Marsh [118].

> "..the problem of *sensor fusion* is the problem of combining multiple measurements from sensors into a single measurement of the sensed object or attribute, called the *parameter*." McKendall and Mintz [100].

> "Data fusion deals with the synergistic combination of information made available by various knowledge sources such as sensors, in order to provide a better understanding of a given scene." Abidi and Gonzales [1].

> "*Multisensor fusion*,..., refers to any stage in the integration process where there is an actual combination (or fusion) of different sources of sensory information into one representational format." Luo [92].

Of these definitions, that by Abidi and Gonzalez is probably the most comprehensive as it incorporates the notion of *Sensor Synergy.* This has also been termed *Sensor Management* [133][114], Sensor coordination [46][66] and Sensor planning or control [65][27][37]. Sensor Management seeks to manage or coordinate the use of sensor resources in a manner that improves the process of perception synergistically. The alternate term Managed Data Fusion, perhaps better captures both the concepts of data fusion and sensor management.

1.2.1 The Issues

Irrespective of application specifics, the main issues in sensor data fusion can abstractly be summarized as follows:

- **Interpretation and Representation.** Firstly, sensor measurements need to be interpreted and understood. This is based on a detailed description of the physical nature of the measurement and a geometrical description of the sensed environment. Secondly, a common representation of the information contained in different measured quantities obtained from diverse sensors is required. The representation should encapsulate sensor uncertainty and performance as related to the observation of the state of nature.

- **Fusion Methodologies.** Required are methods for combining diverse information using a common representation in a manner that is consistent and coherent and avoids distortion or biases caused by malfunctioning sensors or outlier measurements.

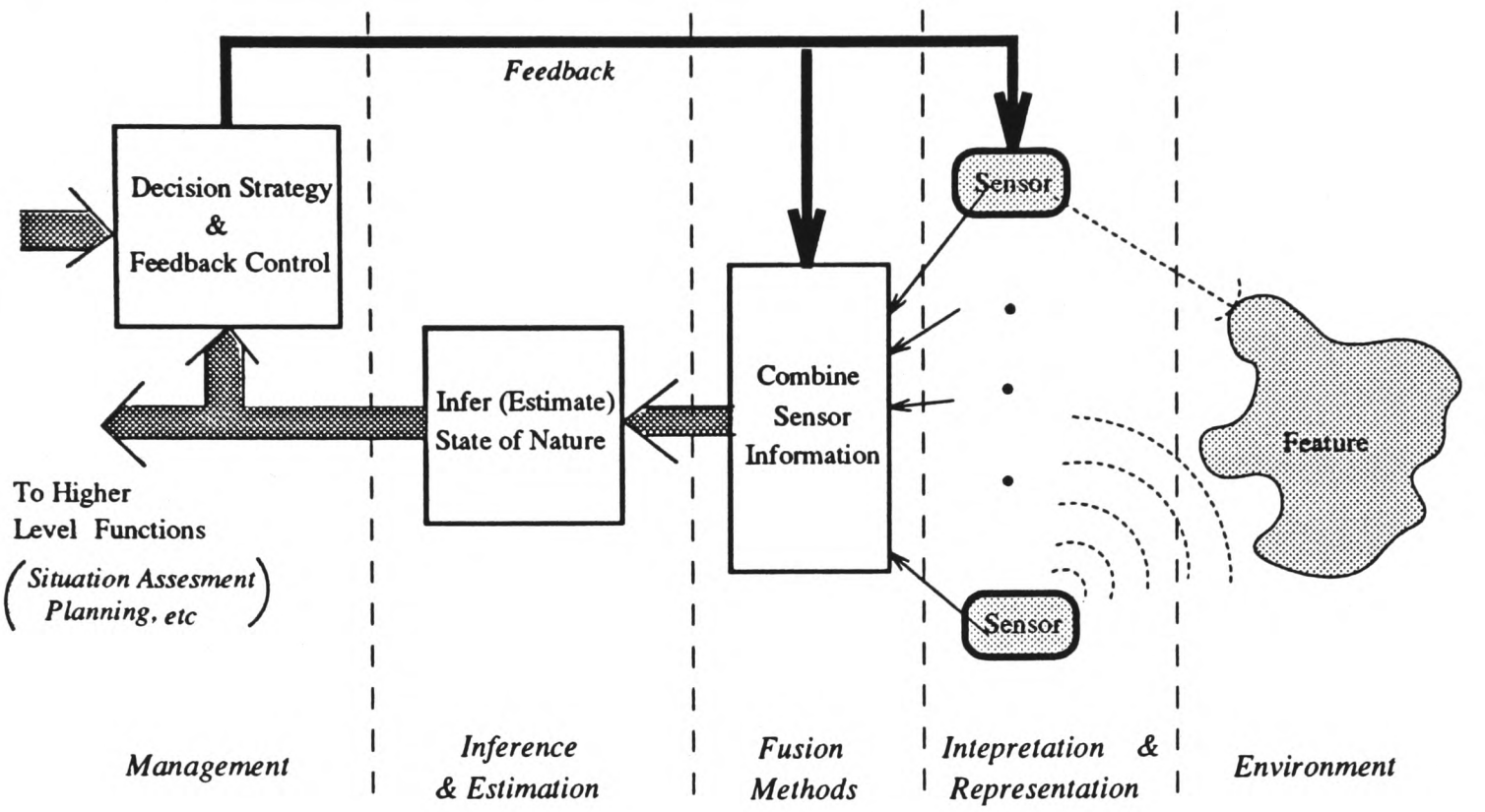


Figure 1.1: A Taxonomy of Issues in Multi-Sensor Data Fusion.

- **Inference and Estimation.** Having combined all the information, we need to infer the state of nature from it in a manner that is optimal by some criteria, given the inherent uncertainty in observations.

- **Sensor Management.** When presented with several sensing options or configurations, the option making the best use of sensor resources to achieve system goals must be chosen. Implicit is a requirement for an *a priori* understanding of the perceptional goals of the system. These goals provide a basis for the criteria used to evaluate the efficacy of alternative strategies and configurations.

While these issues are *fundamental* in any data fusion problem, a major consideration which determines the methods which can be employed, is the multi-sensor **architecture**.

1.2.2 Background

For the most part, the main issues in sensor data fusion have been addressed separately, often in an *ad hoc* manner and in the context of specific systems and architectures. We now discuss some of this work with respect to the issues summarized above. The methods applied generally fall under quantitative and qualitative techniques. We emphasize quantitative methods because quantitative analysis provides a common language and embodies a *corpus* of well-understood techniques while also providing a means of performance evaluation. However, mention will be made, where pertinent, of qualitative methods.

Sensor Models and Representation. Sensor models (descriptions) are, in the first instance, aimed at the interpretation of measurements based on developing an understanding of the sensor and the sensed environment. Such an approach is exemplified in the sensor models by Kuc and Siegel [85] and Leonard [88] for sonar and Hillis [71] for a touch sensor. Models of this form, while invaluable in providing good physical and geometrical descriptions of sensors, do not quantitatively describe the uncertainty of the sensor measurements. In addition, measurements are not described in a common representation appropriate for a multi-sensor system. Probability theory [112] is the most widely used method for describing and representing uncertainty in a way that abstracts from a sensor's operational details. Durrant-Whyte [47] addresses the representation problem by developing probabilistic models and describes such models as "..enabling different sensors to communicate to each other in a common dimensionless language". Quantitative methods for developing noise models (see Gelb [59]) have been used by Faugeras and Ayache [52] among others, to evaluate and model uncertainty in vision sensing. More recently, probabilistic models for sensors have been developed by Barshan [13, 14] for several inertial navigation sensors, based on tech-

niques described by Jenkins and Watts [79]. Such probabilistic descriptions are extremely useful in augmenting physical models such as those of Kuc and Seigel [85] and Leonard [88] as shown in [97, 98] thus providing a way of objectively evaluating sensors and the information they provide in a common representation for data fusion.

Qualitative methods [36], have also been used to describe sensors as exemplified by Flynn [57] for sonar and infra-red and the logical sensor descriptions of Henderson [70]. Such descriptions are useful when it is intended to address data fusion using qualitative reasoning methods.

Sensor Fusion and Inference. Much work has been done in developing methods of combining information from different sensors. The basic approach has been to pool the information using what are essentially "weighted averaging" techniques with varying degrees of complexity. Stone [130] proposes such a pool based on a probabilistic representation of information. There are several variations on the theme as discussed by Berger [18] and Zidek [138] provides a survey of these methods. Non-probabilistic methods such as Dempster Shafer evidential reasoning [137] have also been used for fusion as described by Garvey *et al* [58]. Inferring the state of nature given a probabilistic representation is, in general, a well understood problem in classical statistical estimation (See Gelb [59]). Representative are methods such as Bayesian estimation [18][113], Kalman Filtering [12][127] *et cetera.* Application of the Kalman filter and its derivatives has been widely reported as typified in the work by Willner [42], Hashemipour *et al* [69], Willsky *et al* [136], Castanon and Teneketzis [28] and Durrant-Whyte *et al* [116, 24, 63] where it is used for both fusion and inference. Non-probabilistic methods for inferring the state based on multi-sensor information have also been developed [58] in addition to methods in artificial intelligence

making use of Neural networks [86] and Expert systems [3].

Sensor Management. Sensor coordination or management is invaluable as a means of reducing uncertainty and resolving ambiguities in comparison with passive multi-sensing. To date, sensor management has largely been dealt with on an *ad hoc* basis using descriptive techniques as made evident in the survey by Waltz [133] of several implemented systems. Much of the work done in sensor management has been in the area of tracking radar systems, for example the work by McKenzie and Mullens [101] and Weinberg [135]. In these systems, the approach has been to develop models of sensor behaviour and performance and then manage sensors on this basis, this being facilitated by the centralized or hierarchical nature of the systems. Ikeuchi and Kanade [76] describe a system which automatically generates sensor control programs based on feature detection performance. More formal attempts to manage sensors are represented in the work described in [125][68] and recently [83]. A large proportion of sensor allocation schemes are based on determining cost functions and performance trade-offs *a priori*, as described by Hovanessian [74] and more recently by Balchen and Dessen [10]. Nash [107] uses cost-payoff assignment matrices to allocate sensors to targets. A demonstrably rational way of making decisions such as those required in sensor management, is through the use of *normative* or decision-theoretic techniques [114]. Normative techniques are desirable because they are based on an axiomatic framework with a performance that can be analysed objectively. Making decisions in this way is hard [18] because of the required rigour and the associated computational difficulties in finding the optimal action and hence is a research area in its own right [134]. Tsitsiklis and Athans [132] highlight these difficulties for a distributed detection problem. Efforts have been made at using normative methods in the pre-selection of sensors as exemplified in the work by

Hager [65] and Fleskes [56] and in the active pursuit of uncertainty reduction by Hager [64]. A particularly relevant example is the work by Blackman [19] that makes use of a utility approach for real-time sensor selection where the utility is related to the predicted covariance matrix in a Kalman filter. Most functional implementations of managed data fusion systems are in combat aircraft where achieving sensor synergy is of the essence given timing, operational and performance constraints. Both Waltz [133] and Popoli [114] describe such systems which use a mixture of the techniques already mentioned.

Architectures. Architectures have traditionally been centralized, utilizing a central processor responsible for implementing data fusion. The need to relieve computational burdens at the central processor leads to Hierarchical systems [110][34] which have the advantage of several levels of abstraction. While ideal for coordination and control, hierarchical architectures are vulnerable to processor failure, computational bottlenecks and inflexibility. The recent trends towards *autonomous* systems have led to the development of distributed architectures and Distributed Sensor Networks (DSNs), for example those described by Chong [35]. Iyengar [77] provides a recent survey of DSNs. A more formally defined refinement of these are the *Decentralized* architectures described by Hashemipour *et al* [69], Tsitsiklis and Athans [132] and Durrant-Whyte [50, 49] and formally specified in [95]. Decentralized systems offer:

- **Modularity.** This is a result of the fact that local sensing and global data fusion takes place at the sensor node itself. This is facilitated by each sensor node being fully autonomous with its own sensing, processing and communications facilities.

- **Scalability and Flexibility.** Because all the functionality is localized in the sensor, scaling the system is simply a matter of adding or removing sensors.

- **Survivability.** Due to the absence of a central processor, the system can withstand the loss of nodes and performance is gracefully degraded.

These are discussed in detail by Durrant-Whyte [49]. There is currently much research in this area as exemplified by the work at Oxford [116, 17, 63, 94].

The above survey illustrates the eclectic nature of methods that have been developed for data fusion and sensor management. This has resulted in implemented systems using a mixture of methods[1] . There are hardly any reported systems making use of a single consistent framework to address *all* the issues of representation, fusion and management, in a way which can be applied to a variety of architectures. The work by Hager [65] and that by Blackman [19] and most recently [20], comes closest to this although not directly applicable to decentralized systems. This is the point at which the work presented in this thesis makes a significant contribution.

1.3 An Information-Theoretic Approach

The Problem

The purpose of the work described in this thesis is to develop a **consistent methodology** for addressing the issues of representation, fusion and management in general, and in *fully decentralized* systems in particular. Although the issues of representation and fusion have, to some extent, already been addressed for decentralized systems, the significant Sensor Management problem in such systems has hardly been addressed.

Fully decentralized sensor management presents several potential problems; (i) guaranteeing consistency and consensus amongst decision-makers, (ii) nature of the criteria

[1] As examples: The system described by Kuczewski [86] uses a Neural network to manage Kalman filter based multi-target tracking in a hierarchical system. Popoli [114] describes a fuzzy decision tree approach utilizing heuristic criteria to manage multi-target tracking and emission control.

for optimality and the question of group or individual optimality, and not least (iii) the maintenance of coherence and rationality in the decisions made. Given these problems, a normative approach to management is the only reasonable way to proceed because the resulting decision structure lends itself to quantitative and objective analysis. A number of difficulties are encountered in the development of a normative approach to management:

1. The information upon which decisions will be based must be formally described and modelled to ensure objectivity and rationality in the decision making.

2. A basis for preference and optimality founded on the perceptional aims of the system, must be developed so that decisions and their outcomes may be evaluated accordingly.

3. Given the autonomous nature of the decision-makers (sensor nodes) in a decentralized system, consistency must be guaranteed and consensus achievable.

The subject of decentralized decision theory is itself a major research area with considerable unresolved issues for which generally prescribed solutions have thus far been elusive [18][134]. However, for sensing systems, some assumptions can be made which simplify the general problem. For a general solution to be developed, it is imperative that the issues of representation, fusion and management be re-addressed within a consistent framework and that requisite tools for overcoming the difficulties of decision-making developed axiomatically. This thesis develops, from first principles, a single consistent framework for addressing Data Fusion and Sensor Management in general, and for decentralized architectures in particular.

Proposed Approach

Our approach is predicated on perception primarily being about a need for information, thus making information and its gain the main goal of multi-sensor systems. Therefore, sensing (for purposes of perception) reduces to the simple process where knowledge or belief concerning a given state is updated based on the occurrence of an observation containing *relevant* information about the state. Implicit in this is an assumption that the observation can be modelled in terms of its information value, which requires detailed knowledge of the source of the observation. This basic model can be refined to include uncertainty in the knowledge representation and the observation itself, and also the possibility of having several observations. We adopt a probabilistic approach to represent sensor uncertainty. Using these ideas together with Bayes Theorem, we develop a **probabilistic information update** relation capturing the essence of multi-sensor systems. From the probabilistic information update, we develop and present the following:

1. **A Complete Data Fusion Model.** This model provides the necessary tools and represents a fundamental description of all the theoretical considerations required to address data fusion and sensor management within a single consistent framework.

2. **Data Fusion Algorithms.** Building on the data fusion model, algorithms for inference in multi-sensor systems are derived.

3. **Information Metrics.** These are methods used to evaluate and compare the amount of information available in a multi-sensor system in general, and in decentralized systems in particular.

4. **Normative Sensor Management Methods.** These methods are based on a decentralized decision-theoretic approach, making use of information based utility functions.

Due to our use of information, information theory and decision-theoretic methods we term our approach **Information-Theoretic**.

1.4 Application to Mobile Robotics

While the theoretical work in this thesis is applicable to a wide variety of multi-sensing problems, we address the specific problem of multi-sensor vehicle localization and map-building.

It has been suggested by Leonard [88] and others, that vehicle localization on the basis of a map or, conversely, map-building on the basis of location information, can be greatly improved using directed sensing strategies such as *focus of attention.* Focus of attention aims at reducing the overall quantity of information while increasing its value by providing only correctly associated measurements [2]. If applied to localization, where a sensor on a vehicle focusses attention on a known feature in the environment, the high band-width of correctly associated measurements from such a sensor can be used *directly* to determine the location of the vehicle. This is in contrast to current methods of obtaining vast amounts of mostly unnecessary measurements then correlating these with map information each localization cycle. Conversely, when applied to map-building, such a sensor provides information concerning an un-mapped feature. Thus a vehicle equipped with such sensors can both localize and map-build simultaneously, given an initial knowledge of some environment feature(s).

Implementation of navigation in this way is, however, incumbent on the availability of a sensor with the ability to focus attention on a given geometrical feature such as the ones

[2] Focus of attention can be found in natural vision systems such as saccade eye motion. Vision research has attempted to mimic this as described by Andersson [5].

naturally occurring in indoor environments. Towards this end, we develop a model for a sonar sensor based on the *phase monopulse* principle. The resulting monopulse-sonar sensor is an important new development which represents a significant improvement in sonar data processing in that it provides for the first time accurate centering information using standard Polaroid devices[3] . Exploitation of this, results in the development of a novel *Tracking Sonar* capable of focussing attention on a given feature while producing *correctly-associated* measurements at rates of up to $30Hz$ in a working range of up to $4m$. Implementation of a localization algorithm at the sensor itself thus turns each Tracking Sonar into a modular guidance sensor capable of obtaining location estimates at approximately the measurement rate of $30Hz$. We make use of the data fusion algorithms that we develop, to fuse the information from these sensors and estimate the position of the vehicle decentrally. We also implement a feature classification algorithm which compliments the tracking of un-mapped features for map-building.

The need for sensor management in such an implementation is self-evident, given that each sensor can track any one of several features in the environment that are in its field of view. Decisions, therefore, need to be made about which features need to be tracked and in what sensor configuration at a given moment. This is necessary in order to obtain the most useful information for whatever task the sensors are performing, be it localization or map-building. In addition, it may become necessary as the vehicle moves to assign and re-assign features (targets) to sensors and to cue sensors and hand over features as they move out of view. Therefore, in summary, our application and implementation are intended to demonstrate the following:

[3] Accurate centering has been achieved before using modified Polaroid sensors capable of measuring amplitude [15].

1. **A Sensor Model.** The detailed modelling of a sensor in terms of its operation and the probabilistic information that it provides, as required by our normative approach.

2. **An Autonomous Data Fusion Sensor.** Development of a single sensor into a self contained guidance sensor suited for application in multi-sensor systems in general and decentralized systems in particular.

3. **Decentralized Directed Guidance Algorithms.** A scheme for decentralized localization and feature classification using the data fusion algorithms that we develop.

4. **Sensor Management.** How feature-sensor assignments, sensor cueing and hand-off can be achieved on a navigating robot.

1.5 Thesis Summary

We commence in **Chapter 2** by presenting a probabilistic model for data fusion and sensor management. We present a Bayesian model of the observation process for a sensor. We describe how the state may be inferred from the probabilistic information obtained from the measurement. We then consider the problem of combining probabilistic information from several sensors so that the state may be inferred from the combined information. We present a section on multi-Bayesian decision-theory in which we highlight the issues and the constraints upon which solutions depend, followed by a brief introduction to utility theory. Finally, we introduce information-theoretic considerations culminating in a statement of probabilistic information update.

In **Chapter 3**, we are concerned with developing architectures and algorithms for data fusion based on the information update paradigm of Chapter 2. Decentralization of the information update leads to a decentralized architecture, having developed centralized and

hierarchical architectures along the way. We present an alternate derivation of the information filter form of the decentralized Kalman filter directly from the information update. We compliment this algorithm with an equivalent one for the classification of discrete states. Similar algorithms have been presented previously starting from different considerations [63][116]. What is presented here, represents an alternative approach based on information-theoretic considerations. This also serves to illustrate how other previously published work fits into our overall framework.

Chapter 4 presents a normative approach to Data Fusion Management. We consider information gain and the reduction of uncertainty as the implicit goals in sensor management. The requisite information for such a decision-theoretic approach is provided by the algorithms of Chapter 3. We then justify and construct utility functions based on information as expected utility. We present entropy and Fisher information based metrics for the information provided by data fusion. We then justify the direct comparison of the information-based utilities leading to a solution for the problem of making decentralized sensing decisions. We discuss the realization of sensor management, culminating in an iterative algorithm akin to bargaining which reduces the computational overheads. We also discuss issues of optimality and rationality together with the management of multiple data fusion algorithms.

Having thus far emphasized the crucial nature of sensor models, we address in **Chapter 5** precisely that issue for sonar. We present a physical model in which we explain sonar measurements based on a Region of Constant Depth (RCD) model. We develop a sonar arrangement based on the phase monopulse principle which provides accurate centering information. This leads to the development of a novel Tracking Sonar with the ability to focus attention on and track RCDs in real time at data rates of up to $30Hz$. This capability

makes the sonar ideal for sensor management applications since RCDs can be selected and tracked at will. We then develop a probabilistic model of the information provided by the Tracking Sonar. We discuss its performance and limitations and, most importantly, its role and use in data fusion.

In **Chapter 6**, we consider vehicle navigation based upon work originally presented by Leonard and Durrant-Whyte [90]. We start with a brief background to put the work in this chapter in context with current work in vehicle navigation. We use the algorithms of Chapter 3 and the sensor model of Chapter 5 to address vehicle localization. This is implemented on the OxNav vehicle (JTR) which has been purpose-built in our laboratory. We present a novel implementation of decentralized localization and present some new results. In addition, we make use of the classification algorithm of Chapter 3 to implement a method of classifying features being tracked by the sensors. This implementation also highlights the need for sensor management in order to maximize the information obtained for localization and generally to coordinate the sensors.

Chapter 7 presents some demonstrations of sensor management for a navigating robot. The chapter starts by presenting typical quantities used for sensor management in the information filter and in the classification algorithm based on the metrics from Chapter 4. The variation of these quantities with several parameters and models is demonstrated. Results are then presented for several feature-sensor assignment problems during localization and feature classification. Sensor hand-off and cueing is then demonstrated by showing the times and points for hand-off for sensors localizing or classifying features. We then conclude this chapter with a brief discussion on the costs of effecting the sensor management actions.

We conclude in **Chapter 8** by summarizing the achievements and contributions of the work presented in this thesis. More importantly, we highlight the limitations and

shortcomings of this work. This leads to a discussion of directions for further research and study.

Summary: *Thesis Road Map*

Figure 1.2 summarizes the work presented in this thesis in the form of a *Thesis Road Map* through the chapters and illustrates the coherence and consistency of our approach.

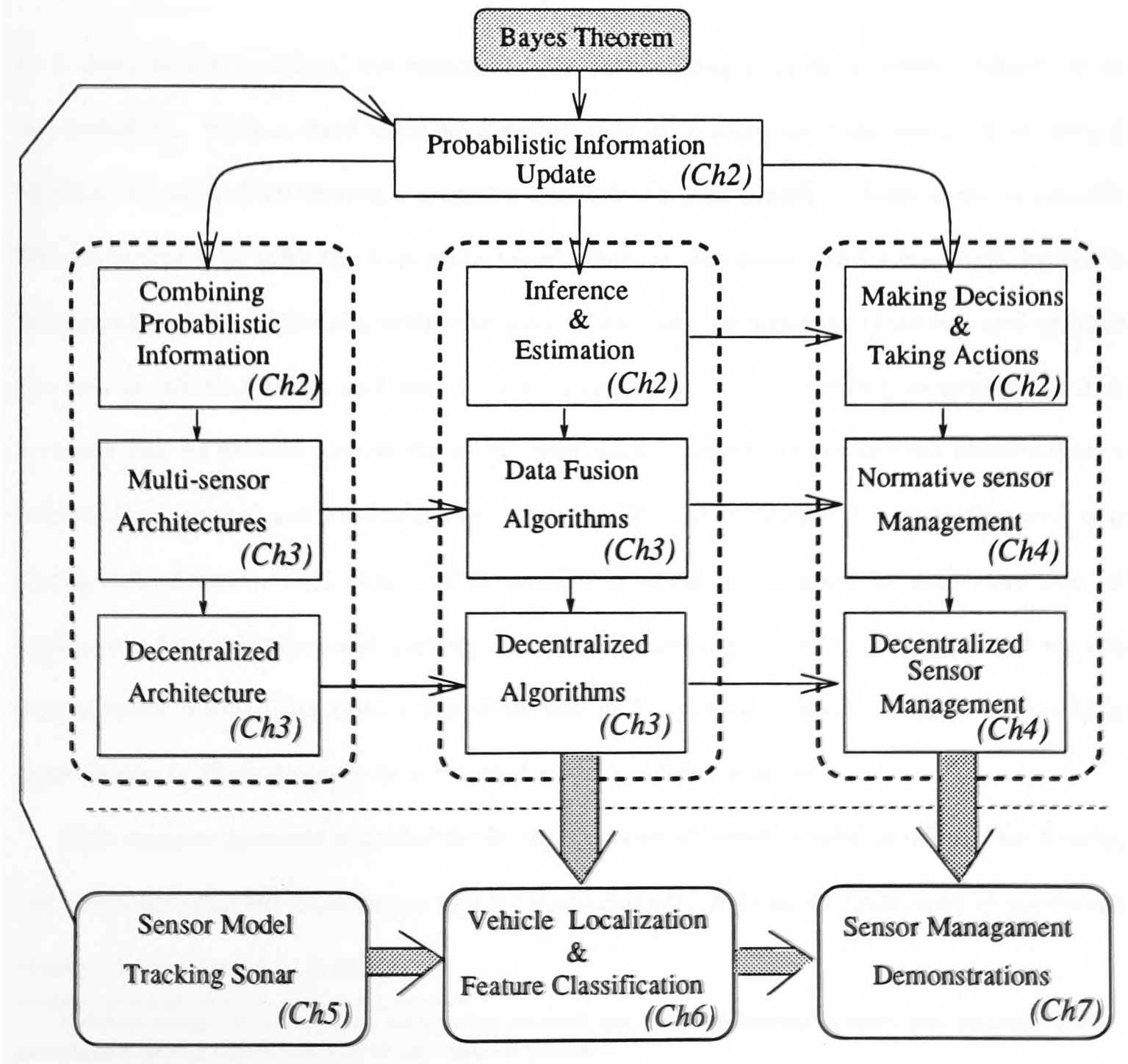


Figure 1.2: A Thesis Road Map. Each *Ch* refers to the appropriate chapter.

Chapter 2

A Probabilistic Model for Managed Data Fusion

In a data fusion problem, we commence by determining a state of nature which we are interested in. Such a state may be a description of the spatial location of an object, its identity in terms of attributes, a complex dynamic state or simply a single numeric quantity. Our objective is to infer the true state based often on incomplete and sometimes conflicting information obtained from a variety of sources and also to influence (control) and optimize the way in which we find and gather such information [1] . In providing measurement data, a sensor can be viewed abstractly as an *information source.* Often several observations or information sources are available thus offering different strategies for gathering and combining information. This necessitates the development of methods for achieving this and techniques for evaluating and making decisions concerning alternate strategies. As we shall demonstrate, probability theory is rich enough and complete in itself, to allow us to address these issues in their entirety in a rational and consistent manner.

This chapter presents a probabilistic *information-theoretic model* as a basis for developing methodologies for data fusion and its management. A detailed treatment of probability theory can be found in [112][80].

[1] Until we define information in Section 2.4 we shall use the term loosely to mean any quantity useful in providing a better understanding of the state of nature.

2.1 Observations, Inference and Estimation

The state of nature is described by an n-dimensional vector $\mathbf{x}$ taken from some state space $\mathcal{X}$, that is, $\mathbf{x} \in \mathcal{X}$. The state is denoted $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and may be discrete or continuous. When the state is continuous the space $\mathcal{X} \subseteq \Re^n$, the n-dimensional Euclidean space. The state $\mathbf{x}$ may also be random or deterministic and in what follows we make no distinction unless otherwise stated. We take m measurements of some physical quantities which give some indication as to the state $\mathbf{x}$. These measurements make up the m-dimensional observation vector $\mathbf{z}$ taken from some observation space $\mathcal{Z}$, that is, $\mathbf{z} \in \mathcal{Z}$. In our context, where we make physical measurements, the space $\mathcal{Z} \subseteq \Re^m$. Each measurement may itself be random or deterministic. The observation is defined by

$$\mathbf{z} \triangleq \mathcal{M}(\mathbf{x}, \mathbf{v}), \tag{2.1}$$

where $\mathbf{v}$ is an unknown observation noise vector of the same dimension as $\mathbf{z}$ and is normally described by a random variable. $\mathcal{M}$ is a generalized *observation model* relating the state space $\mathcal{X}$ to the observation space $\mathcal{Z}$. $\mathcal{M}$ is never known precisely and invariably a practical model of $\mathcal{M}$ always incorporates some approximations. The closeness with which such a model approaches the "true" relation between $\mathcal{Z}$ and $\mathcal{X}$ requires a good understanding of the phenomenological nature of the observations, a physical model of the measurement device itself and also knowledge of the sensed environment.

The probabilistic information contained in $\mathbf{z}$ about $\mathbf{x}$ is described by the probability distribution function (PDF) $p(\mathbf{z} \mid \mathbf{x})$, known as the *likelihood function* [2]. Such information is considered *objective* because it is based on observations. The likelihood function contains all the relevant information from the observation $\mathbf{z}$ required in order to make inferences

[2] Berger gives an intuitive description of the likelihood function that ".. $\mathbf{x}$ would be a more plausible occurrence if $p(\mathbf{z} \mid \mathbf{x})$ were large"[18].

about the true state $\mathbf{x}$. This is termed the *Likelihood Principle* [18]. It can be shown using the factorisation theorem for sufficient statistics as derived by Lehman [87], that the likelihood function is proportional to a function $g(T(\mathbf{z}) \mid \mathbf{x})$, where $T(\mathbf{z})$ is a *sufficient statistic* of $\mathbf{x}$. The statistic $T(\mathbf{z})$ is said to be sufficient for the unknown state $\mathbf{x}$ if the conditional distribution $p(\mathbf{z} \mid T(\mathbf{z}))$ is independent of $\mathbf{x}$. From statistical theory, a sufficient statistic of the state can be used to make inferences and decisions concerning the state $\mathbf{x}$. Using such a sufficient statistic, the raw observed data can be discarded and a minimal sufficient statistic used to make inferences.

2.1.1 Bayes Theorem

The likelihood function does not however tell the whole story if, before measurement, information about the state $\mathbf{x}$ is made available exogenously. Such *a priori* information about the state is encapsulated in the prior distribution function $p(\mathbf{x})$ and is regarded as *subjective* because it is not based on any observed data. How such prior information and the likelihood information interact to provide *a posteriori* (combined prior and observed) information, is solved by Bayes Theorem which gives the posterior conditional distribution of $\mathbf{x}$ given $\mathbf{z}$ as

$$p(\mathbf{x} \mid \mathbf{z}) = \frac{p(\mathbf{z} \mid \mathbf{x})\, p(\mathbf{x})}{\int p(\mathbf{z} \mid \mathbf{x}) p(\mathbf{x})\, d\mathbf{x}} = \frac{p(\mathbf{z} \mid \mathbf{x}) p(\mathbf{x})}{p(\mathbf{z})}, \tag{2.2}$$

where $p(\mathbf{z})$ is the marginal distribution. An equivalent formulation, which often proves quite useful, arises from a consideration of sufficient statistics as follows; if $T(\mathbf{z})$ is a sufficient statistic for $\mathbf{x}$ with distribution $g(T(\mathbf{z}) \mid \mathbf{x})$, then from the factorization theorem we can write the posterior as

$$p(\mathbf{x} \mid \mathbf{z}) = p(\mathbf{x} \mid T(\mathbf{z})) = \frac{g(T(\mathbf{z}) \mid \mathbf{x})\, p(\mathbf{x})}{\int g(T(\mathbf{z}) \mid \mathbf{x}) p(\mathbf{x})\, d\mathbf{x}}. \tag{2.3}$$

Often the distribution $g(T(\mathbf{z}) \mid \mathbf{x})$ is easier to handle and model than the actual likelihood $p(\mathbf{z} \mid \mathbf{x})$ and so Equation 2.3 may be used instead of Equation 2.2. If no subjective prior information is available, a distribution $p(\mathbf{x})$ is required in Equation 2.2 such that the resulting posterior contains no more information than that contained in the likelihood. Such a prior distribution is called a *non-informative prior*, the discussion of which is deferred until Section 2.4.2.

In an attempt to reduce uncertainty and resolve ambiguity, several measurements may be taken over time before constructing the posterior. We define the set of all observations up to time k as follows

$$\mathbf{Z}^k \triangleq \{\mathbf{z}(1), \mathbf{z}(2), \cdots, \mathbf{z}(k)\}. \tag{2.4}$$

The posterior distribution of $\mathbf{x}$ given the set of observations $\mathbf{Z}^k$ is now computed as

$$p(\mathbf{x} \mid \mathbf{Z}^k) = \frac{p(\mathbf{Z}^k \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{Z}^k)}, \tag{2.5}$$

or can be computed recursively after each observation $\mathbf{z}(k)$ as follows

$$p(\mathbf{x} \mid \mathbf{Z}^k) = \frac{p(\mathbf{z}(k) \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{Z}^{k-1})}{p(\mathbf{z}(k) \mid \mathbf{Z}^{k-1})}. \tag{2.6}$$

In this recursive definition, we are not required to store all the observations but need only consider the current observation $\mathbf{z}(k)$ at each kth step. In practise, this is the form of Bayes Theorem which is most commonly used.

2.1.2 Classical Inference and Estimation

Statistical inference is concerned with obtaining an estimate $\hat{\mathbf{x}}$ of the state of nature $\mathbf{x}$ from the posterior distribution $p(\mathbf{x} \mid \mathbf{Z}^k)$ together with some measure of estimation accuracy. A fundamental concept in statistical estimation is that of *expectation*. We define E^p or simply

E, the expected value of some function $f(\mathbf{x})$ with respect to a distribution p as

$$\begin{aligned} E\{f(\mathbf{x})\} &= \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \quad \text{if continuous,} \\ &= \sum_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})p(\mathbf{x}) \quad \text{if discrete.} \end{aligned} \tag{2.7}$$

Using the expectation operator we can obtain the *moments* of a distribution. The posterior mean (1st moment) and covariance (2nd moment) are given by

$$\bar{\mathbf{x}} \triangleq E^{p(\mathbf{x}|\mathbf{Z}^k)}\{\mathbf{x}\}, \quad \text{and} \quad \mathbf{P}_{\mathbf{x}|\mathbf{Z}^k} \triangleq E^{p(\mathbf{x}|\mathbf{Z}^k)}\left\{(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T\right\}, \tag{2.8}$$

respectively. We can show that over all possible estimates $\hat{\mathbf{x}}$, the posterior mean $\bar{\mathbf{x}}$ minimizes the estimate covariance by showing that

$$\mathbf{P}_{\hat{\mathbf{x}}} = E\left\{(\mathbf{x}-\bar{\mathbf{x}}+\bar{\mathbf{x}}-\hat{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}}+\bar{\mathbf{x}}-\hat{\mathbf{x}})^T\right\} = \mathbf{P}_{\mathbf{x}|\mathbf{Z}^k} + (\bar{\mathbf{x}}-\hat{\mathbf{x}})(\bar{\mathbf{x}}-\hat{\mathbf{x}})^T,$$

from which setting $\nabla_{\hat{\mathbf{x}}}\mathbf{P}_{\hat{\mathbf{x}}} = 0$ yields

$$\nabla_{\hat{\mathbf{x}}}\mathbf{P}_{\hat{\mathbf{x}}} = -2\bar{\mathbf{x}} + 2\hat{\mathbf{x}} = 0, \quad \text{therefore} \qquad \bar{\mathbf{x}} = \hat{\mathbf{x}}. \tag{2.9}$$

Showing that $\nabla_{\hat{\mathbf{x}}}\nabla_{\hat{\mathbf{x}}}{}^T\mathbf{P}_{\hat{\mathbf{x}}} > 0$, demonstrates that the covariance $\mathbf{P}_{\hat{\mathbf{x}}}$ is minimized when the estimate equals the posterior mean. Estimators which minimize the posterior covariance are termed *minimum variance* estimators.

Classical inference techniques include the *Maximum Likelihood* (ML) estimate obtained by maximizing the likelihood function

$$\hat{\mathbf{x}}_{ML} = \arg\max p(\mathbf{Z}^k \mid \mathbf{x}). \tag{2.10}$$

For a general distribution, the ML estimate corresponds to the mode of the distribution. The *Maximum a posteriori* (MAP) estimate is obtained by maximizing the posterior distribution

$$\hat{\mathbf{x}}_{MAP} = \arg\max p(\mathbf{x} \mid \mathbf{Z}^k). \tag{2.11}$$

Objectivity in the estimate (or inferred state) is maintained by considering only the observed information, that is the likelihood function. Given a non-informative prior, the MAP estimate is the same as the ML estimate and so in this case the MAP estimate is as objective as the ML estimate. A particularly popular estimator is the Least Squares Estimate which minimizes the sum of the square errors, that is, minimizes the Euclidean distance between the true state $\mathbf{x}$ and the estimate $\hat{\mathbf{x}}$ given the set of observations $\mathbf{Z}^k$. The equivalent estimator for random variables is called the *Minimum Mean Square Error* (MMSE) estimate and is written

$$\hat{\mathbf{x}}_{MMSE} = \arg\min_{\hat{\mathbf{x}}} E^{p(\mathbf{x}|\mathbf{Z}^k)} \left\{ (\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T \right\}. \tag{2.12}$$

Differentiating with respect to $\hat{\mathbf{x}}$ and equating to zero gives the minimizing estimate as

$$\begin{aligned} \nabla_{\hat{\mathbf{x}}} \int (\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}}) p(\mathbf{x} \mid \mathbf{Z}^k)\, d\mathbf{x} &= -2 \int (\mathbf{x} - \hat{\mathbf{x}}) p(\mathbf{x} \mid \mathbf{Z}^k) d\mathbf{x} = 0 \\ \text{and} \quad \hat{\mathbf{x}} &= \int \mathbf{x} p(\mathbf{x} \mid \mathbf{Z}^k) d\mathbf{x} = E\{\mathbf{x} \mid \mathbf{Z}^k\}. \end{aligned} \tag{2.13}$$

Thus an MMSE estimate is given by the conditional mean. That $\hat{\mathbf{x}}$ is an MMSE estimate can also be shown geometrically by a decomposition of the error between $\mathbf{x}$ and an arbitrary estimator of $\mathbf{x}$, $\mathbf{g}(\mathbf{z})$ into 2 orthogonal components, followed by a consideration of the mean-squared error between $\mathbf{x}$ and the arbitrary estimator. The mean-squared error is given by

$$E\left\{ (\mathbf{x} - \mathbf{g}(\mathbf{z}))^T (\mathbf{x} - \mathbf{g}(\mathbf{z})) \right\} \geq E\left\{ (\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T \right\} \tag{2.14}$$

with equality iff $\mathbf{g}(\mathbf{z}) = \hat{\mathbf{x}}$ [121]. It follows from the previous discussion that the MMSE estimate is also a minimum variance estimator. When the mean of the conditional density $p(\mathbf{x} \mid \mathbf{Z}^k)$ coincides with the mode, the MAP estimate is equivalent to the MMSE estimate.

Using these ideas, estimation algorithms can be developed for various applications. The techniques presented thus far are, in general, well understood in terms of classical statistical

theory. However, when there is a multiplicity of information sources in a variety of configurations and topologies, a number of more complex issues arise. In the next and subsequent sections, we develop a model for addressing these issues.

2.2 Probabilistic Information Fusion

In the previous section, the use of multiple observations from a single source was presented as a way of reducing uncertainty. Another approach aimed at reducing uncertainty and obtaining a more complete view of the state of nature, is the *fusion* of information originating from a number of spatially and physically diverse sources. The initial difficulty with using multiple information sources lies in the way information is combined. In tackling the problem the following questions should be addressed:

- How *relevant* to the problem at hand is information from each source i? This entails validating or checking the information from i for its relevance as regards inference and decisions concerning $\mathbf{x}$.

- How *reliable* is source i's information? This entails attaching a measure of value such as a weight to the information provided by each source depending on *a priori* reliability information if this is known.

We start by defining $\mathcal{N}$, the set of all the available information sources

$$\mathcal{N} = \{i\}, \quad \text{for } i = 1,2,\ldots,N. \tag{2.15}$$

We also define the set of *all* observations made by the set of sensors $\mathcal{N}$ up to time step k as

$$\{\mathbf{Z}^k\} = \cup_i \mathbf{Z}_i^k, \quad \forall i \in \mathcal{N}; \quad \text{where } \mathbf{Z}_i^k = \{\mathbf{z}_i(1), \mathbf{z}_i(2), \cdots, \mathbf{z}_i(k)\}, \tag{2.16}$$

and where $\mathbf{Z}_i^k$ is information source i's set of observations up to the time step k. The issue is now to compute the global posterior distribution $p(\mathbf{x} \mid \{\mathbf{Z}^k\})$ using information available from each source. In what follows, we will assume that each information source communicates either a local posterior PDF $p(\mathbf{x} \mid \mathbf{Z}_i^k)$, or likelihood $p(\mathbf{z}_i(k) \mid \mathbf{x})$ in the recursive form of Equation 2.6.

Several solutions have been proposed to the information aggregation problem from a Bayesian perspective as discussed by Berger [18] and Bacharach [8, 9]. Most of the solutions proposed are set in a social context where different assumptions must be made such as reticence possibly borne of adversarial individuals. Such assumptions determine the nature of the solution, hence the importance of Assumption 1. We consider in turn the two approaches generally proposed in the literature, culminating in what we have termed the Independent Likelihood Pool [96].

2.2.1 Linear Opinion Pool

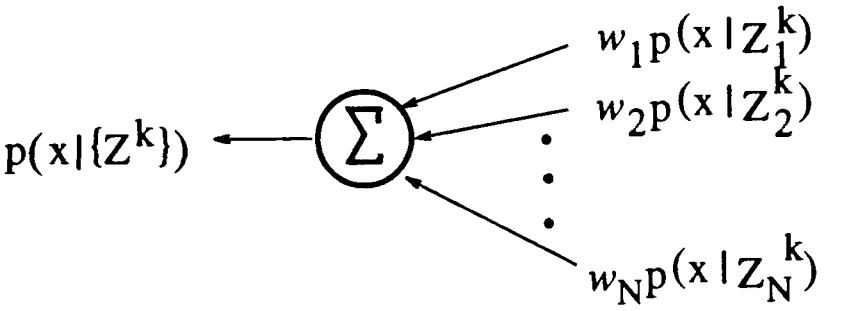


Figure 2.1: Linear Opinion Pool.

The linear opinion pool as described by Stone [130] is a deceptively simple approach to aggregating probability distributions. The posteriors from each information source are combined as follows

$$p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = \sum_j w_j \, p(\mathbf{x} \mid \mathbf{Z}_j^k), \tag{2.17}$$

where w_j is a weight such that $0 \leq w_j \leq 1$ and $\sum_j w_j = 1$. This is illustrated in Figure 2.1. The weight w_i models the significance attached to information source i. The weights can be used to model source reliability or trustworthiness and can be used to "weight out" faulty sensors. However, formulation of Equation 2.17, requires that weight w_i be known before information from i is evaluated [44] which presents difficulties. Though a general methodology for obtaining the weights w_i has not been forthcoming, problem specific and arguably *ad hoc* methods have been developed. Worthy of mention is the entropy-based weighting technique by Basir and Shen [16] which establishes the level of dependence between information sources and then computes appropriate weights before information aggregation. Applying Bayes theorem to Equation 2.17 and assuming each information source has its own subjective prior distribution gives

$$p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = w_1 \frac{p(\mathbf{Z}_1^k \mid \mathbf{x})p(\mathbf{x}_1)}{p(\mathbf{Z}_1^k)} + w_2 \frac{p(\mathbf{Z}_2^k \mid \mathbf{x})p(\mathbf{x}_2)}{p(\mathbf{Z}_2^k)} + \cdots + w_N \frac{p(\mathbf{Z}_N^k \mid \mathbf{x})p(\mathbf{x}_N)}{p(\mathbf{Z}_N^k)}.$$

Such addition of probabilities (assuming equal weights) results in an inability to reinforce opinion. This is because the Linear Opinion Pool does not take appropriate cognizance of the possible availability of independent information locally at each node i. An illustrative example is provided by considering a discrete state $\mathbf{x} = \{x_1, x_2\}$. If $N - 1$ information sources report posteriors of $(0.1, 0.9)$ and one dissenting source j reports $(0.7, 0.3)$, for a large N it would seem sensible for j to be ignored or reinforced towards $(0.1, 0.9)$. The following are the results for a 5 sensor system

$$\frac{1}{5}\sum_5 (0.1,\ 0.9) = (0.1,\ 0.9), \tag{2.18}$$

and when one sensor is dissenting

$$\frac{1}{5}\left((0.7,\ 0.3) + \sum_4 (0.1,\ 0.9)\right) = (0.22,\ 0.78). \tag{2.19}$$

The Linear Opinion Pool would give undue credence to j's opinion and give an erroneous result. The need to redress this leads to the second approach.

2.2.2 Independent Opinion Pool

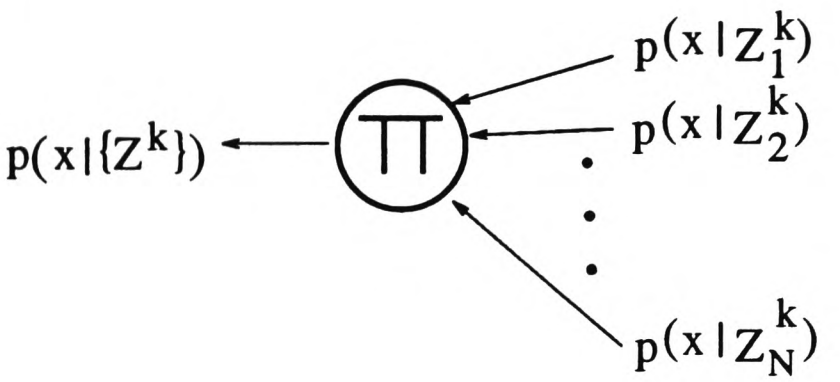


Figure 2.2: Independent Opinion Pool.

The Independent Opinion Pool is defined by

$$p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = \alpha \prod_j p(\mathbf{x} \mid \mathbf{Z}_j^k), \tag{2.20}$$

where α is a normalizing constant. This is illustrated in Figure 2.2. The implicit assumption in the Independent Opinion Pool is that the information obtained conditioned on the observation set is independent. In general, this is a difficult condition to satisfy particularly when the information sources are human! In the realm of measurement and experimentation based on physical laws and principles the conditional independence of the observations can be justified experimentally. This is usually done in the process of modelling the likelihood function $p(\mathbf{z}(k) \mid \mathbf{x})$ by showing that the residual uncertainty in the observation arises from uncorrelated noise terms. The independent pool allows for the reinforcement of opinions which, when each source is assumed to have subjective prior information, is desirable. However, the Independent Opinion Pool can be extreme in its reinforcement of opinion. As a result of the product in Equation 2.20, unwarranted extreme reinforcement of opinion based

on common information would result. This can be seen as follows

$$p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = \alpha \left[\frac{p(\mathbf{Z}_1^k \mid \mathbf{x}) p(\mathbf{x}_1)}{p(\mathbf{Z}_1^k)} \times \frac{p(\mathbf{Z}_2^k \mid \mathbf{x}) p(\mathbf{x}_2)}{p(\mathbf{Z}_2^k)} \times \cdots \times \frac{p(\mathbf{Z}_N^k \mid \mathbf{x}) p(\mathbf{x}_N)}{p(\mathbf{Z}_N^k)} \right]. \tag{2.21}$$

If the prior information is obtained from the same source, then the prior $p(\mathbf{x}_1) = p(\mathbf{x}_2) = \cdots = p(\mathbf{x}_N)$ and this results in unwarranted reinforcement of the posterior through the product of the priors $\prod_j p(\mathbf{x}_j)$. Thus the Independent Opinion Pool is only appropriate when the priors are obtained independently on the basis of subjective prior information at each information source.

2.2.3 Independent Likelihood Pool

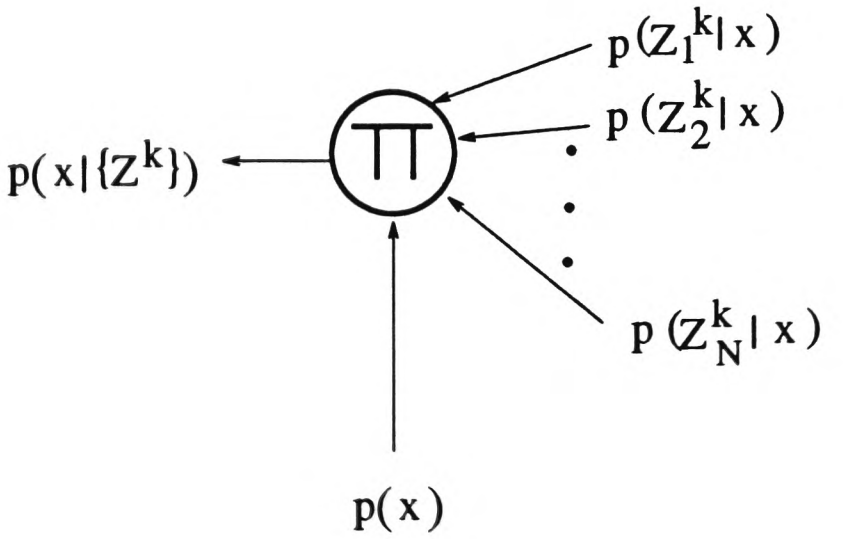


Figure 2.3: Independent Likelihood Pool.

When each information source has common prior information, the opinion pool which more accurately describes the situation is developed as follows: Bayes theorem for the global posterior, that is, the distribution of $\mathbf{x}$ conditioned on *all* the observations up to time k, is given by

$$p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = \frac{p(\{\mathbf{Z}^k\} \mid \mathbf{x})\, p(\mathbf{x})}{p(\{\mathbf{Z}^k\})} = \frac{p(\mathbf{Z}_1^k, \mathbf{Z}_2^k, \ldots, \mathbf{Z}_N^k \mid \mathbf{x})\, p(\mathbf{x})}{p(\mathbf{Z}_1^k, \mathbf{Z}_2^k, \ldots, \mathbf{Z}_N^k)}. \tag{2.22}$$

The distribution $p(\mathbf{Z}_1^k, \mathbf{Z}_2^k, \ldots, \mathbf{Z}_N^k \mid \mathbf{x})$ is difficult to compute in practice if there are dependencies among the elements of each information source's observation set and those of other information sets which do not depend on $\mathbf{x}$. However, for sensor systems, it is reasonable to assume that the likelihoods from each information source i, that is, $p(\mathbf{Z}_i^k \mid \mathbf{x})$ are independent. This is because the only thing the observations have in common is the state and even so, we must show in practice that

$$p(\mathbf{Z}_1^k, \mathbf{Z}_2^k, \cdots, \mathbf{Z}_N^k \mid \mathbf{x}) = p(\mathbf{Z}_1^k \mid \mathbf{x}) p(\mathbf{Z}_2^k \mid \mathbf{x}) \cdots p(\mathbf{Z}_N^k \mid \mathbf{x}). \tag{2.23}$$

If this is the case, we can write Equation 2.22 as

$$\begin{aligned} p(\mathbf{x} \mid \{\mathbf{Z}^k\}) &= \frac{p(\mathbf{Z}_1^k \mid \mathbf{x})\, p(\mathbf{Z}_2^k \mid \mathbf{x}) \cdots p(\mathbf{Z}_N^k \mid \mathbf{x})\, p(\mathbf{x})}{p(\mathbf{Z}_1^k, \mathbf{Z}_2^k, \ldots, \mathbf{Z}_N^k)} \\ &= \frac{p(\mathbf{x}) \prod_j p(\mathbf{Z}_j^k \mid \mathbf{x})}{p(\mathbf{Z}_1^k, \mathbf{Z}_2^k, \ldots, \mathbf{Z}_N^k)}, \end{aligned} \tag{2.24}$$

which can be written recursively as

$$p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = \alpha\, p(\mathbf{x} \mid \{\mathbf{Z}^{k-1}\}) \left[\prod_j \underbrace{p(\mathbf{z}_j(k) \mid \mathbf{x})}_{\text{likelihood}} \right], \tag{2.25}$$

where α is a normalizing constant independent of $\mathbf{x}$. From a communication standpoint, the Independent Likelihood Pool is consistent with the Bayesian update of Equation 2.6 as follows: We can write an expanded observation set $\{\mathbf{Z}^k\}$ according to Equation 2.16 which encompasses all the observations from all the sources and then simply write Equation 2.6 in terms of this expanded observation set as if the observations were from a single source.

2.2.4 Discussion

Of the methods described above, the Independent Opinion Pool and the Independent Likelihood Pool more accurately model the situation in multi-sensor systems where the conditional distributions of the observations can be shown to be independent. The choice between

these depends on the origin of the prior information. The Linear Opinion Pool is useful if there are dependencies between information sources and it is known *a priori* how to assign individual weights to the probabilistic information from each source.

Significant in this thesis is the architectural paradigm which results when the combination of probabilistic information is distributed. Each information source performs the task of combining information from all the other information sources and of computing a global posterior, on the *proviso* that all the information sources in the system are able to communicate with each other directly. This amounts to replicating the independent opinion or likelihood pools (Equations 2.17- 2.25) at every information source with some simplifications resulting. In this way, the posterior obtained is identical at each information source. Our choice of combining information in such a distributed system using the independent opinion and likelihood pools differs from the approach by Chair and Varshney [31] which is based on distributing a form of the Linear Opinion Pool which they proposed in [30].

Inference and estimation in multi-information systems is based on the same methods as described in Section 2.1.2. Instead of using the posterior $p(\mathbf{x} \mid \mathbf{Z}^k)$, we make use of the global posterior $p(\mathbf{x} \mid \{\mathbf{Z}^k\})$. Thus the MAP and MMSE estimators discussed previously can be obtained from the global posterior. However, for the ML we make use of

$$p(\{\mathbf{Z}^k\} \mid \mathbf{x}) = \prod_j p(\mathbf{Z}_j^k \mid \mathbf{x}), \tag{2.26}$$

based on the independence assumption made earlier. An issue of great practical importance to data fusion is that of *data association.* This is the problem of correctly associating data or information from different sensors. In this thesis we do not directly address data association but it can still be solved using a Bayesian approach [129].

2.3 Making Decisions and Taking Actions

Managing the sensor systems modelled above involves making decisions and taking actions under uncertainty. Modelling the sensing process probabilistically gives us access to decision-theoretic techniques developed in Decision theory literature [54][134][65]. Our starting point in making such decisions is the *a posteriori* distribution, $p(\mathbf{x} \mid \mathbf{Z}^k)$ in the case of a single information source and, in the case of multiple information sources, $p(\mathbf{x} \mid \{\mathbf{Z}^k\})$ and its various local components. The posterior PDF encompasses all that is known about the state $\mathbf{x}$ and therefore any decisions concerning $\mathbf{x}$ can be based on it.

2.3.1 Decisions and the Single Bayesian

A decision maker is concerned with taking the best possible actions on the basis of the information available and in the face of uncertainty. We define the action a which is an element of a set of possible actions $\mathcal{A}$. We also define a *Utility* function $U(\mathbf{x}, a)$ defined for all $(\mathbf{x}, a) \in (\mathcal{X} \times \mathcal{A})$, which gives a measure of the utility of taking a given action a when the true state is a particular $\mathbf{x}$. Conversely, we can define a *Loss* function $L(\mathbf{x}, a)$ which gives a measure of the loss incurred in taking action a when the state is $\mathbf{x}$. Hence we can write that $U(\mathbf{x}, a) = -L(\mathbf{x}, a)$. We can compute the expected utility (or loss) β of taking an action a as follows

$$\beta(p(\mathbf{x} \mid \mathbf{Z}^k), a) \quad \triangleq \quad E^{p(\mathbf{x}|\mathbf{Z}^k)}\{U(\mathbf{x}, a)\} \tag{2.27}$$

The Bayes action $\hat{a}$ is the strategy which maximizes the expected utility $\beta(p(\mathbf{x} \mid \mathbf{Z}^k), a)$

$$\hat{a} = \arg\max_a \beta(p(\mathbf{x} \mid \mathbf{Z}^k), a). \tag{2.28}$$

From the Likelihood Principle, it can be shown that this is equivalent to maximizing

$$\int_{\mathbf{x}} U(\mathbf{x}, a)\, p(\mathbf{Z}^k \mid \mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}. \tag{2.29}$$

Analytically, this is a rational framework for making decisions and taking actions, the caveat being an ability to determine appropriate utility functions.

When the action space is the same as the state space the decision problem becomes the same as for the inference problem already discussed. For instance, we can define a loss function such as the squared-error loss and so the minimization of the Loss form of Equation 2.27 results in an alternate statement of the MMSE estimate that we have already encountered, as follows

$$\min_{\hat{\mathbf{x}} \in \mathbf{x}} \int L(\hat{\mathbf{x}}, \mathbf{x}) p(\mathbf{x} \mid \mathbf{Z}^k) d\mathbf{x},$$

where the loss L is defined as the squared error loss

$$\underbrace{L(\hat{\mathbf{x}}, \mathbf{x})}_{\text{mmse}} = (\hat{\mathbf{x}} - \mathbf{x})^2, \quad \text{where } \hat{\mathbf{x}} = \int \mathbf{x}\, p(\mathbf{x} \mid \mathbf{Z}^k) d\mathbf{x}. \tag{2.30}$$

Having already introduced inference, we shall not consider this any further and instead concern ourselves only with action spaces which are not the same as the state space and are countably finite.

2.3.2 Decisions with Multiple Information Sources

When considering decision making with multiple sources of information, it is important to delineate and define the problem precisely in terms of the structural level of the decision making, the nature and level of interaction during the decision making and the presence or absence of a dominant decision maker. We therefore make the following distinctions: Full *cooperation* implies that each information source does not withhold information and is always ready to provide it when required. This is in contrast with *non-cooperation* whereby information sources may be reticent or refuse to provide information altogether [21]. In our case we make the following assumption;

Assumption 1 : Individual information sources cooperate fully and are able to communicate probabilistic information as required with negligible delay.

This is a justifiable assumption for sensor systems quite unlike the general dialogues described by Bacharach [9] where "...opinions are only partially disclosed".

The problem of making decisions involving multiple information sources has been formulated under a variety of assumptions in the literature [21][91]. We shall consider the following representative cases:

Case 1: Consider a system consisting of N information sources and a single overall decision maker. Such a system is illustrated in Figure 2.4. The task of the decision maker is in the first instance to combine probabilistic information from all the sources and then make decisions based on the global posterior. Given that the global posterior is $p(\mathbf{x} \mid \{\mathbf{Z}^k\})$, the Bayes group action is given by

$$\hat{a} = \arg\max_a \beta(p(\mathbf{x} \mid \{\mathbf{Z}^k\}), a), \tag{2.31}$$

where

$$\beta(p(\mathbf{x} \mid \{\mathbf{Z}^k\}), a) = E^{p(\mathbf{x}|\{\mathbf{Z}^k\})}\{\mathbf{U}(\mathbf{x}, a)\}, \tag{2.32}$$

and $\mathbf{U}(\mathbf{x}, a)$ is a group utility function. Implicit in this case is the assumption that individual information sources give up the capacity to make decisions to a single overall decision maker.

Case 2: Consider a system consisting of N Bayesians each able to obtain its own probabilistic information which it shares with *all* the other Bayesians before computing a posterior PDF. From the discussion in Section 2.2, the posterior PDF obtained by each Bayesian is identical and is given by $p(\mathbf{x} \mid \{\mathbf{Z}^k\})$. Each Bayesian is then required to compute an optimal action which is consistent with those of the other Bayesians. It is assumed that the decision

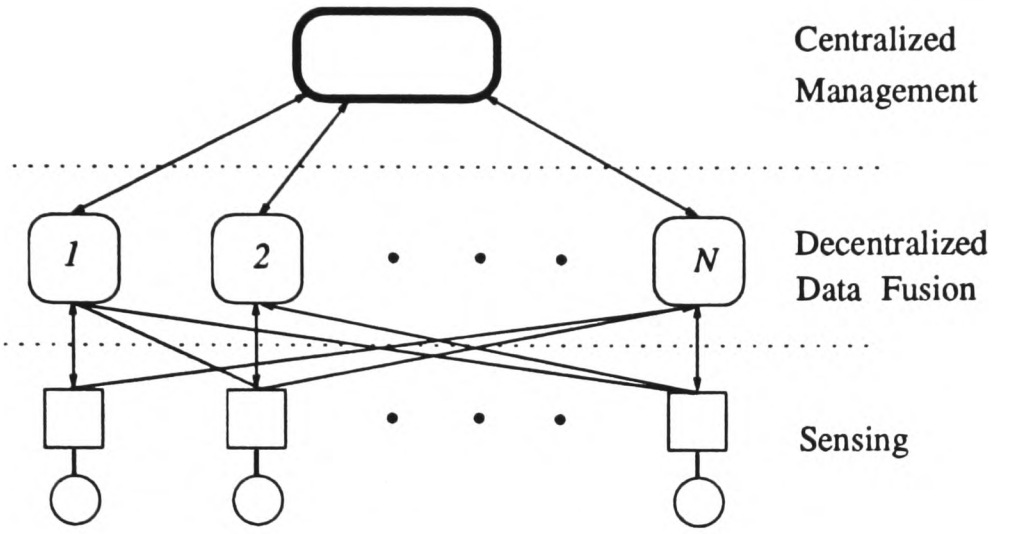


Figure 2.4: "Super" Bayesian approach to management decision making.

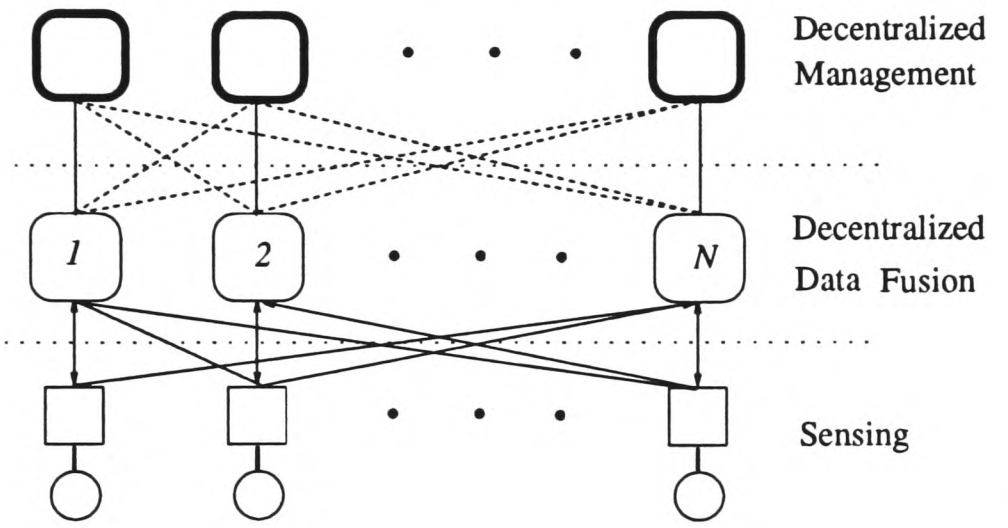


Figure 2.5: Multi-Bayesian approach to management decision making. The connections between the decision-making stage and the data fusion stage between different Bayesians are shown as dotted because they may or may not exist depending on communication requirements for decision-making.

process occurs after all sharing of probabilistic information. This system is illustrated in Figure 2.5.

The solution in the first case is well defined in terms of classical Bayesian analysis. This is facilitated by the presence of a central decision maker and the use of a group utility function. Therefore, this becomes a generalization of the discussion in Section 2.3.1 and is often termed the "super Bayesian" approach [134]. Such a decision-theoretic solution

is desirable because it avoids thresholding in the decision process. This differs from the method of Chair and Varshney [30], where the decisions made by individual sensors are weighted based on reliability and then comparing a weighted sum of individual decisions with a threshold based on the likelihood to give the global decision.

The second case however presents a considerable challenge for which a universally prescribed solution is not possible partly because of the lack of generally applicable criteria for determining rationality and optimality. This is instantiated by the problems associated with deciding between individual and group optimality, the choice of which determines the approach taken [132]. Towards this end, group decision making in our context can be simplified by the non-antagonistic nature of the agents involved in the decision making (Assumption 1) and the implicit goal of *group* information gain or uncertainty reduction.

If the optimal action $\hat{a}_i$ at each Bayesian i happens to be the same this is then the group consensus action. In general it will not be the same and so a solution would proceed as follows; if each Bayesian i computes an acceptable set of actions $A_i \subseteq \mathcal{A}$ and the set $A = \cap_j A_j$ is non-empty, then the group action is selected from the set A. An acceptable class of actions can therefore be obtained by maximizing

$$\beta(p(\mathbf{x} \mid \{\mathbf{Z}^k\}), a) = \sum_j w_j \; E^{p(\mathbf{x}|\{\mathbf{Z}^k\})} \{U_j(\mathbf{x}, a)\}, \tag{2.33}$$

where $0 \leq w_j \leq 1$ and $\sum_j w_j = 1$. Equivalently, from the Likelihood Principle, this could be written as a maximization of

$$\sum_j w_j \int U_j(\mathbf{x}, a) \; p(\mathbf{Z}_j^k \mid \mathbf{x}) \; p(\mathbf{x}) \; \mathbf{d}(\mathbf{x}), \tag{2.34}$$

using the non-recursive form of the information update Equation 2.5. That Equation 2.33 and Equation 2.34 represent acceptable actions is easy to see. What becomes extremely difficult and has been the subject of much research is the choice of an optimal action $\hat{a}$ from

the acceptable set. If the utility functions U_i can be directly compared then maximization of Equation 2.33 or Equation 2.34 gives the optimal group action. The general validity of direct comparisons of utility is questionable given Arrow's impossibility theorem [7]. However, as Savage [120] argues, there are situations where such comparisons seem plausible.

When comparisons can be justified, a general solution is obtained by maximizing Equation 2.33, the solution of which happens to be a special case of a more general solution suggested by Weerahandi and Zidek [134] stated as the maximization

$$\hat{a} = \arg\max_{a} \left\{ \sum_{j} w_j \left[E^{p_j} \left\{ U(\mathbf{x}, a_l) \right\} - c(j) \right]^{\gamma} \right\}^{1/\gamma}, \tag{2.35}$$

where $c(j)$ is regarded as decision maker j's security level which plays the role of "safeguarding j's interests". The weight w_j is as given in Equation 2.33 and $-\infty \leq \gamma \leq \infty$. The case when $\gamma = 1$, gives a solution which minimizes Equation 2.33 and $\gamma = -\infty$ and $\gamma = \infty$ give the so called Bayesian max-min and min-max solutions respectively [18][134].

When direct comparisons cannot be justified, the celebrated Nash [106] solution which makes no direct utility comparisons, can be used. This solution is obtained by maximizing the product of individual expected utilities

$$\hat{a} = \arg\max_{a} \prod_{j} \left[E^{p_j} \left\{ U(\mathbf{x}_j, a_l) \right\} - c(j) \right], \tag{2.36}$$

in which the value of $c(j)$ can be used to maintain some level of individual optimality. The value of $c(j)$ plays no part in the actual derivation of the Nash solution and is thus arbitrary. Weerahandi and Zidek have shown that setting $\gamma = 0$ in Equation 2.35 reduces it to a form of the Nash product known as the Nash-Kalai solution [81], thus establishing an equivalence between these methods. For sensor data fusion applications, we shall endeavour to show that it is reasonable to construct a utility function such that the expected utility for a given action is directly comparable by different decision-makers.

2.3.3 Utility Theory

Having introduced Utility and Loss in the previous section, we now introduce these notions formally. Utility allows the placement of value on the results of decisions thus permitting a subjective preference ordering on actions and their outcomes. The results of decisions can be defined on the set $\Re$. Given uncertainty, these results may be probabilistic and defined on the set $\mathcal{P}$. We can say that action a_l is associated with the state $\mathbf{x}$ having a probability distribution p_l where $p_l \in \mathcal{P}$. An alternative, perhaps more precise, way of writing p_l is by writing it as $p(\mathbf{x} \mid a_l)$ thus making the conditioning on a_l explicit. Because the distribution on $\mathbf{x}$ may also be conditioned on other states or may be a likelihood, we find the notation p_l, which makes the conditioning on a_l implicit, less confusing. The "value" of taking action a_l is given by the expectation

$$E^{p_l}\left\{U(\mathbf{x}, a_l)\right\}, \tag{2.37}$$

where $U(\mathbf{x}, a_l)$ is a real valued function which represents (through expected value) the decision maker's preference over the elements of $\mathcal{P}$. This has meaning if it is possible to state preferences between the elements of $\mathcal{P}$. Thus, if the distribution p_1 over $\mathbf{x}$ is preferred to p_2, we expect that the utility function $U(\mathbf{x}, a_l)$ is such that

$$E^{p_2}\left\{U(\mathbf{x}, a_2)\right\} < E^{p_1}\left\{U(\mathbf{x}, a_1)\right\}. \tag{2.38}$$

Therefore $U(\mathbf{x}, a)$ is a utility function with the same preference structure as the decision maker. This ability to state preferences rationally is the basis of the axiomatic definition of utility [54][18]. In order for $U(\mathbf{x}, a)$ to be a valid utility function, the expected utilities must have the same preference ordering as the true preferences concerning the probabilistic outcomes in $\mathcal{P}$. This necessitates the need for a preference ordering among the elements of $\mathcal{P}$. The following are the axioms on which such a preference pattern is based:

- **Axiom 1.** If p_1 and p_2 are in $\mathcal{P}$ then either $p_1 \prec p_2$, or $p_1 \approx p_2$, or $p_2 \prec p_1$.

- **Axiom 2.** If $p_1 \preceq p_2$ and $p_2 \preceq p_3$, then $p_1 \preceq p_3$.

- **Axiom 3.** If $p_1 \prec p_2$ then $\alpha p_1 + (1-\alpha)p_3 \prec \alpha p_2 + (1-\alpha)p_3$, for $0 < \alpha < 1$.

Axiom 1 implies the ability to express preference and Axiom 2 is a transitivity requirement. Axiom 3 implies that in identical situations where p_1 and p_2 occur with the same probability, the preferred p based on some preference criterion is chosen. The Axioms presented above can be strengthened by the addition of another Axiom which implies that there is 'no heaven or hell', that is, no infinitely good or infinitely bad outcome. Based on the results by Fishburn [55] this fourth axiom is not necessary to guarantee the existence of utility functions. The sufficiency of the above three axioms is the basis of our use and interpretation of utility functions. These axioms guarantee that the decision maker's preference in $\mathcal{P}$ coincides with preferences according to the expected utilities of the elements of $\mathcal{P}$. A rigorous formal proof of the existence of utility functions is given by Von Neumann and Morgenstein [108]; Chernoff and Moses [33] give a more palatable proof of the same. An implicit assumption in the above is that the decision-maker has knowledge of the exact distribution of an element of $\mathcal{P}$. Such exact knowledge may not be available in sensor systems where assumptions and approximations are often made in order to deal with otherwise intractable modelling problems. For such situations we take comfort in the fact that approximate knowledge of the distributions in $\mathcal{P}$ has been shown to be adequate in a proof by Ferguson [54] making use of so-called *personal probabilities* as defined in [6].

Various methods have been described in the literature for constructing utilities and examples of these are found in [82][18][43]. Utility functions can be modelled to give prescribed behaviour such as risk aversion, risk proneness and neutrality. These different behaviours

are modelled analytically based on Jensen's inequality as follows; a risk averse decision profile implies convexity as follows

$$U(E\{\mathbf{x}\}, a) \geq E\{U(\mathbf{x}, a)\}. \tag{2.39}$$

Similarly, risk prone behaviour is implied by the concavity

$$U(E\{\mathbf{x}\}, a) \leq E\{U(\mathbf{x}, a)\}, \tag{2.40}$$

and neutral behaviour follows from the linear relation $E\{U(\mathbf{x}, a)\} = U(E\{\mathbf{x}\}, a)$.

2.4 Information in Probability

As the final piece in the puzzle, we quantitatively consider the information inherent in probability. Probability distributions contain information about the underlying states which they describe. It is useful to quantify this information for the purpose of making comparisons and determining the degree of uncertainty.

2.4.1 Measures of Information

Entropy

Entropy, sometimes known as Shannon information, is a fundamental and generally applicable measure of information in probability distributions. Entropy is the uncertainty associated with a probability distribution and thus gives a measure of the descriptive complexity of a PDF. We shall define entropy as Shannon [123] first defined it; Catlin [29] gives a particularly lucid derivation of the same. Entropy is defined as the expectation of the negative of the log-likelihood [3] of a PDF

$$h(p(\mathbf{x})) \triangleq E\{-ln\ p(\mathbf{x})\}. \tag{2.41}$$

[3] The log-likelihood here is simply the log of a distribution and is quite different from the likelihood function.

We can define the entropy of the posterior distribution of $\mathbf{x}$ given $\mathbf{Z}^k$ at time k as follows

$$\begin{aligned} h(k) \triangleq h(p(\mathbf{x} \mid \mathbf{Z}^k)) &= E\{-ln\, p(\mathbf{x} \mid \mathbf{Z}^k)\} \\ &= -\int p(\mathbf{x} \mid \mathbf{Z}^k)\, ln\, p(\mathbf{x} \mid \mathbf{Z}^k) d\mathbf{x} \qquad \text{if continuous,} \\ &= -\sum p(\mathbf{x} \mid \mathbf{Z}^k)\, ln\, p(\mathbf{x} \mid \mathbf{Z}^k) \qquad \text{if discrete.} \end{aligned} \tag{2.42}$$

With this definition of entropy, a measure of ignorance is given by $ln(1/p(\mathbf{x}))$ [4].

We can develop the entropy relationship for Bayes Theorem as follows; if we negate and take expectations of the log of Bayes theorem (Equation 2.6) then using the definition of Entropy in Equation 2.42 we have

$$\begin{aligned} E\left\{-ln\,[p(\mathbf{x} \mid \mathbf{Z}^k)]\right\} &= E\left\{-ln\,[p(\mathbf{x} \mid \mathbf{Z}^{k-1})]\right\} - E\left\{ln\left[\frac{p(\mathbf{z}(k) \mid \mathbf{x})}{p(\mathbf{z}(k) \mid \mathbf{Z}^{k-1})}\right]\right\} \\ h(k) &= h(k-1) - E\left\{ln\left[\frac{p(\mathbf{z}(k) \mid \mathbf{x})}{p(\mathbf{z}(k) \mid \mathbf{Z}^{k-1})}\right]\right\}. \end{aligned} \tag{2.43}$$

This demonstrates that conditioning with respect to observations reduces entropy, that is, increases information. *Mutual* information is defined as the information about one variable contained in another. In an observation process, we define mutual information at time k as the information about $\mathbf{x}$ contained in the observation $\mathbf{z}(k)$, that is

$$i(k) = \mathbf{I}(\mathbf{x}, \mathbf{z}(k)) \triangleq E\left\{ln\left[\frac{p(\mathbf{z}(k) \mid \mathbf{x})}{p(\mathbf{z}(k))}\right]\right\}. \tag{2.44}$$

With this definition, the entropy relationship for Bayes theorem (Equation 2.43) can be written as

$$h(k) = h(k-1) - i(k), \tag{2.45}$$

which simply says that the entropy following an observation is reduced by an amount equal to the information inherent in the observation.

[4] It can also been argued that ignorance may alternatively be measured by $(1 - p(\mathbf{x}))$. This leads to an alternative definition of entropy as shown recently by Pal and Pal [111] which, although giving similar entropy values as the original definition, has different boundary properties which prove useful in some applications.

Fisher information

Fisher information gives a measure of the amount of information about $\mathbf{x}$ present in the observations $\mathbf{Z}^k$. Like entropy, Fisher information is developed from the log-likelihood. The gradient of the log-likelihood is called the score function s, given by

$$s(\mathbf{Z}^k, \mathbf{x}) \triangleq \nabla_{\mathbf{x}} ln\ p(\mathbf{Z}^k, \mathbf{x}). \tag{2.46}$$

By considering s as a random variable, we obtain its mean from

$$E\{s(\mathbf{Z}^k, \mathbf{x})\} \quad = \quad E\left\{\frac{\nabla_{\mathbf{x}} p(\mathbf{Z}^k, \mathbf{x})}{p(\mathbf{Z}^k, \mathbf{x})}\right\} = \nabla_{\mathbf{x}} \int p(\mathbf{Z}^k, \mathbf{x}) d\mathbf{x} = 0.$$

The Fisher information matrix is defined as the covariance of the score function

$$\mathbf{J}(k) \triangleq E\{\nabla_{\mathbf{x}} ln\ p(\mathbf{Z}^k, \mathbf{x})\ (\nabla_{\mathbf{x}} ln\ p(\mathbf{Z}^k, \mathbf{x}))^T\}. \tag{2.47}$$

This can be rewritten as the negative expectation of the Hessian of the log-likelihood

$$\mathbf{J}(k) = -E\{\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}{}^T ln\ p(\mathbf{Z}^k, \mathbf{x})\}. \tag{2.48}$$

For non-random parameters, the Fisher information matrix is defined in terms of the likelihood as follows

$$\mathbf{J}(k) \triangleq -E\{\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}{}^T ln\ p(\mathbf{Z}^k \mid \mathbf{x})\}. \tag{2.49}$$

Fisher information is useful in estimation; the inverse of the Fisher information matrix is the *Cramer-Rao lower bound* [12] which bounds the mean-squared error of any unbiased estimator of $\mathbf{x}$.

It is more effective to quantify information using entropy for comparative reasons since it can be applied to both discrete and continuous distributions. In continuous state estimation problems, Fisher information is more useful since algorithms can be developed which

maximise it, but this usefulness is restricted to continuous state estimation. Appendix A discusses the relationship between Fisher information and entropy.

2.4.2 Probabilistic Information Update

Since the concept of entropy is the converse of information, a measure of information is given by negating entropy. Thus, we can write from the definition of entropy (Equation 2.41) that information is given by $E\{ln\ p(\mathbf{x})\}$. Whereupon the recursive Bayes update (Equation 2.6) can be written as information update by taking expectations as follows

$$E\left\{ln\left[p(\mathbf{x} \mid \mathbf{Z}^k)\right]\right\} = E\left\{ln\left[p(\mathbf{x} \mid \mathbf{Z}^{k-1})\right]\right\} + E\left\{ln\left[\frac{p(\mathbf{z}(k) \mid \mathbf{x})}{p(\mathbf{z}(k) \mid \mathbf{Z}^{k-1})}\right]\right\}, \quad (2.50)$$

$$posterior\ information = prior\ information + observation\ information.$$

We can write similar information update equations for the multi-information source systems based on the Independent Opinion Pool and the Independent Likelihood Pool as

$$E\left\{ln\left[p(\mathbf{x} \mid \{\mathbf{Z}^k\})\right]\right\} = \sum_j E\left\{ln\ p(\mathbf{x} \mid \mathbf{Z}_j^k)\right\}, \quad (2.51)$$

and

$$E\left\{ln\left[p(\mathbf{x} \mid \{\mathbf{Z}^k\})\right]\right\} = E\left\{ln\left[p(\mathbf{x} \mid \{\mathbf{Z}^{k-1}\})\right]\right\} + \sum_j E\left\{ln\left[\frac{p(\mathbf{z}_j(k) \mid \mathbf{x})}{p(\mathbf{z}_i(k) \mid \{\mathbf{Z}^{k-1}\})}\right]\right\}, \quad (2.52)$$

respectively. A useful device, which is also quite illuminating, is simply to write Bayes algorithm in terms of log-likelihoods as this also brings out the additive nature of the update.

Initialization and Non-informative priors.

According to Bayes theorem (Equation 2.2), when no information is available subjectively about the state $\mathbf{x}$, we require a prior PDF which contains no more information than is

inherent in the likelihood in order to maintain objectivity. A non-informative prior is a PDF that contains no information about the state $\mathbf{x}$. The subject of non-informative priors has deservedly received much attention [78][29] and the difficulties associated with them are generally regarded as one of the main short-comings of Bayesian analysis. Methods do exist for approximating non-informative priors. These are largely based on equating imprecision with ignorance, an example being the use of the *Maximum entropy principle* [18]. Appendix A.2 discusses some of the approaches used to obtain non-informative priors.

However, we argue that in a system where information update is recursive, crude methods which may be eschewed by Bayesian purists such as Maximum entropy priors and Fisher information priors, yield plausible results. This is provided that the system has an adequately modelled likelihood function (observation model).

2.5 Summary

The probabilistic model presented provides a complete and consistent framework for confronting issues in decentralized sensing in particular, and multi-sensor systems in general as discussed in Chapter 1, in terms of:

- Development of architectures based on the methods of combining probabilistic information discussed in Section 2.2.
- Estimation and inference of the true state when presented with probabilistic information from a multiplicity of sources.
- Distributed and decentralized management based on the group decision principles set out in Section 2.3.2, utilizing results and properties from Section 2.4 on information and its update.

In presenting this model, we have attempted to illustrate the completeness and self-sufficiency of the Bayesian approach by demonstrating how the different components of the general problem outlined in Chapter 1 can be solved within the Bayesian paradigm. While the model presented above is not in itself limited to fully-connected systems, there are extensions required with regards to the communication in non-fully connected systems. Grime and Durrant-Whyte [62] have shown that by considering information sets using Bayes Theorem, the additional communication issues can be resolved, following which the model proposed in this chapter can be applied.

Computationally, Bayesian calculations are difficult particularly when considering group decisions. Tsitsiklis and Athans [132] show that minimisations similar to the maximizations of Equations 2.33 - 2.35 are NP-hard for a distributed system. In Chapter 4, we propose methods which, while reducing the computational difficulties in the above formulations, maintain the same rational behaviour in the decision making for purposes of sensor management.

Chapter 3

Deriving Architectures and Algorithms

In this chapter, we take the probabilistic model developed in Chapter 2 and directly develop a consistent methodology for data fusion in terms of architectures and algorithms. There are two themes in this chapter; firstly the architectures for data fusion resulting from our discussion in the previous chapter on probabilistic information fusion, and secondly, methods and algorithms for inference in data fusion.

We develop two representative algorithms. The first of these algorithms is the celebrated Kalman filter [12] which has seemingly countless applications [133][11][25]. The Kalman filter (KF) has been derived using a variety of approaches such as those based on the Gauss-Markov theorem as shown by Scharf [121] making assumptions on the joint distributions of the state and observations[99]. Other approaches make no assumptions on the joint distributions but apply orthogonality results to give least squares estimates [12]. These approaches result in the standard formulation of the Kalman filter. An alternative statement of the Kalman filter has been presented in the form of the *information filter* also called the inverse covariance form. This form has been shown to have some computational advantages and also overcomes some of the difficulties associated with initialization. Formulations of the information form of the Kalman Filter have been presented by Maybeck [99], Grime and Durrant-Whyte [63] starting from the usual formulation of the Kalman filter. What we present here is an alternative *direct* derivation of the information form of the Kalman

filter starting from a notion of probabilistic information and its update. Such a derivation is consistent with the architectures to be developed thus making the architectures a logical extension of the algorithms and *vice versa.*

The second algorithm is the discrete equivalent of the information filter, which updates beliefs in the classification of a discrete state. This algorithm is similar to that initially proposed by Rao [115] and described in Rao and Manyika [117]. However, in our probabilistic model this algorithm comes about as a result of considering the state in the probabilistic information update of Equation 2.50 as discrete. The continuous and discrete algorithms are therefore demonstrated to be mutually consistent as regards their derivation and development based on information.

3.1 Architectures from Information Update

We now demonstrate how the information update (Equation 2.50 - 2.52) is formulated for various architectures culminating in the update equations for a decentralized architecture.

3.1.1 Single Sensor Systems

The information update for a single sensor system written in terms of log-likelihoods is

$$ln\ p(\mathbf{x} \mid \mathbf{Z}^k) = ln\ p(\mathbf{x} \mid \mathbf{Z}^{k-1}) + ln\ [\alpha\ p(\mathbf{z}(k) \mid \mathbf{x})]\,. \tag{3.1}$$

This equation describes a single Bayesian information source as described in Chapter 2. Such a single sensor system is illustrated in Figure 3.1. Multi-sensor systems are formed by combining elements of this single sensor system.

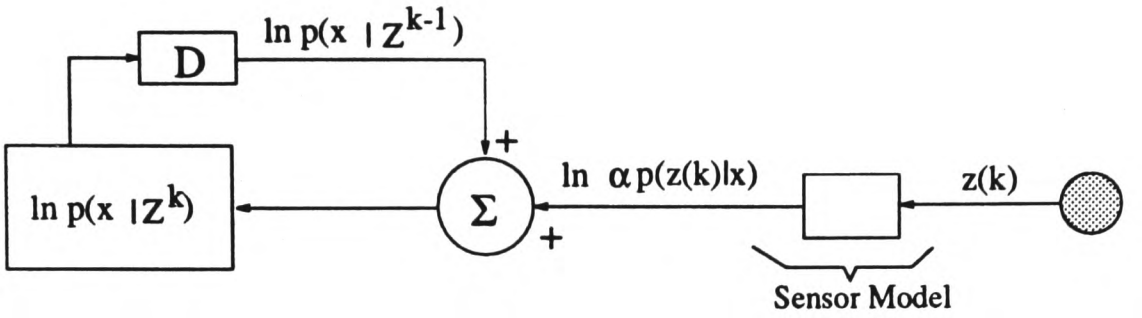


Figure 3.1: Single sensor information update.

3.1.2 Centralized and Hierarchical Systems

Consider a centralized multi-sensor system with a central fusion processor. Each sensor node [1] . j makes local observations $\mathbf{z}_j(k)$ at time k resulting in the set $\mathbf{Z}_j^k$ as defined in Equation 2.16 Each sensor communicates this information to the central processor which fuses it with information from other sensors to construct a global posterior PDF. The computation of the global result depends on the method employed to combine probabilistic information as discussed in Chapter 2.

1. **Independent Opinion Pool.** If, according to the discussion in Chapter 2, it is known that each sensor node has subjective prior information then the Independent Opinion Pool of Section 2.2.2 can be used resulting in the following: A complete local posterior distribution $p(\mathbf{x} \mid \mathbf{Z}_i^k)$ is computed at each sensor i and then communicated to the central processor. In this case the central processor then computes the global posterior as the summation

$$ln\ p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = \sum_j \underbrace{ln\ p(\mathbf{x} \mid \mathbf{Z}_j^k)}_{\text{communicated}}, \tag{3.2}$$

[1] We shall refer to each sensor in a multi-sensor system as a sensor node, node or simply sensor. Correspondingly, we shall use the terms local, nodal and partial interchangeably to mean the result obtained using only information available locally at each sensor or node.

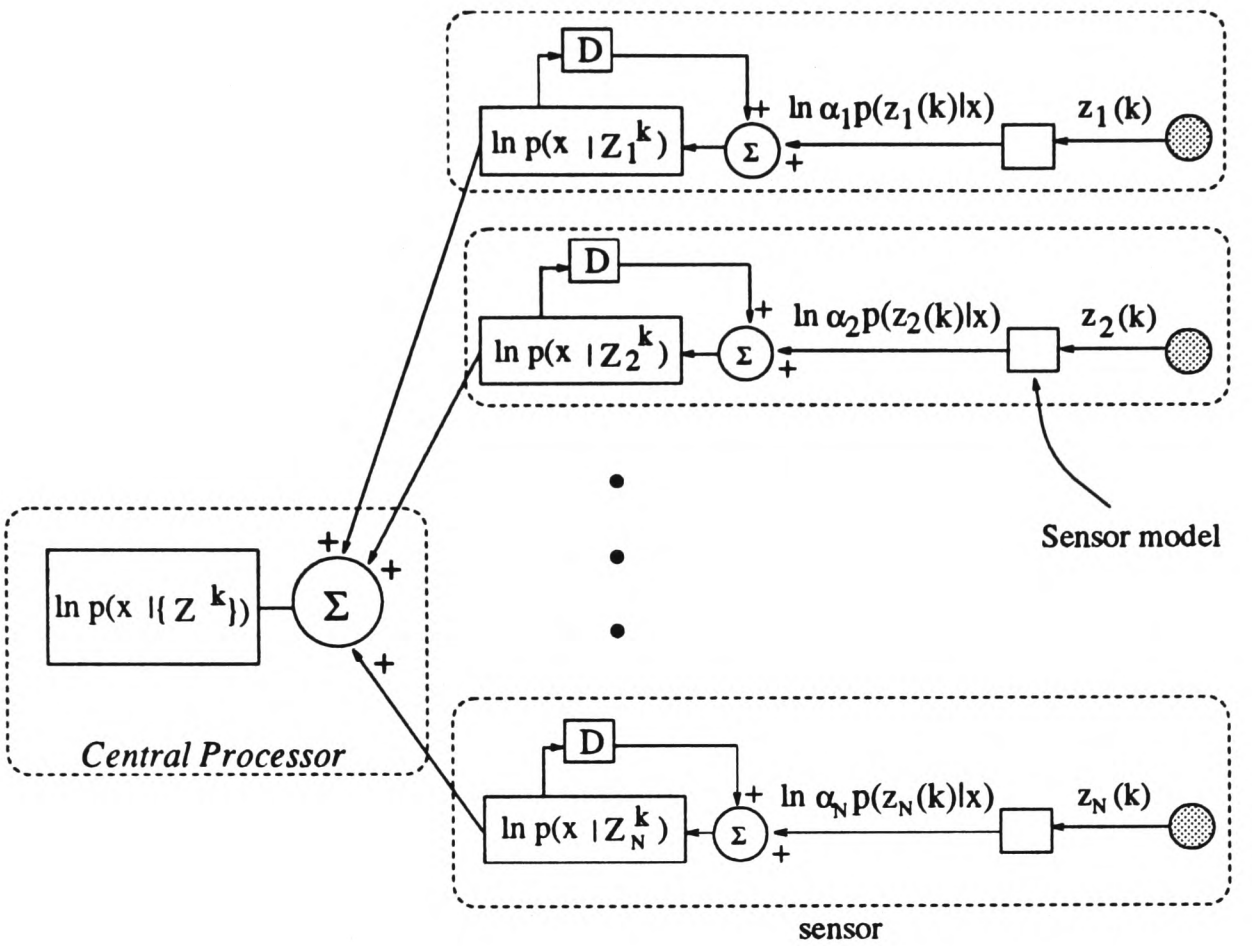


Figure 3.2: Centralized information update for architecture communicating local information to central processor.

where the local posterior is given by

$$ln\ p(\mathbf{x} \mid \mathbf{Z}_j^k) = \left[ln\ p(\mathbf{x} \mid \mathbf{Z}_j^{k-1}) + ln\ [\alpha_j\ p(\mathbf{z}_j(k) \mid \mathbf{x})]\right]. \tag{3.3}$$

Thus the global posterior is simply a summation of the local posteriors from each sensor. An important consideration is that the prior used in Equation 3.3 is a local one which is updated as $p(\mathbf{x} \mid \mathbf{Z}_i^k) \rightarrow p(\mathbf{x} \mid \mathbf{Z}_i^{k-1})$. This architecture is illustrated in Figure 3.2.

2. **Independent Likelihood Pool.** If no *local* subjective prior information is available, as is often the case, and Independent Likelihood Pool is assumed, the central processor

computes the global posterior using the following update rule

$$ln\ p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = ln\ p(\mathbf{x} \mid \{\mathbf{Z}^{k-1}\}) + \sum_j ln\ [\alpha_j\ p(\mathbf{z}_j(k) \mid \mathbf{x})]\,. \tag{3.4}$$

The nature of the hierarchy determines the information communicated to the central processor by sensor i; it may be in one of the following forms

- raw sensor data $\mathbf{z}(k)$ or
- likelihoods $[\alpha_i\ p(\mathbf{z}_i(k) \mid \mathbf{x})]$ or from considerations of sufficient statistics $p(T(\mathbf{z}_i(k)) \mid \mathbf{x})$ (see Chapter 2).

The communication of raw sensor data requires the central processor to have knowledge of the models for each sensor thus allowing it to compute the likelihoods $p(\mathbf{z}_i(k) \mid \mathbf{x})$ from the observation $\mathbf{z}_i(k)$ for all the sensors i. In this way, the global posterior $p(\mathbf{x} \mid \{\mathbf{Z}^k\})$ of Equation 3.4 is computed in its entirety at the central processor as illustrated in Figure 3.3. If each sensor has some computational capability the likelihood term may be computed locally at each sensor then communicated. This has the advantage that the sensor model can be kept locally at the each sensor. This architecture is illustrated in Figure 3.4. The global posterior is a summation of the prior global posterior and the likelihood information obtained at each sensor i.

Choosing between these hierarchies depends not only on the nature of the prior information, but also on the application. A system comprising simple sensors such as temperature transducers for example is ideally suited to the hierarchy implied by Figure 3.3. The communication of likelihoods or local posteriors, shown in Figure 3.4 and Figure 3.2 respectively, means that the central processor need not have any knowledge of the sensor models at each sensor node. This reduces the computational burden on the central processor. The hierar-

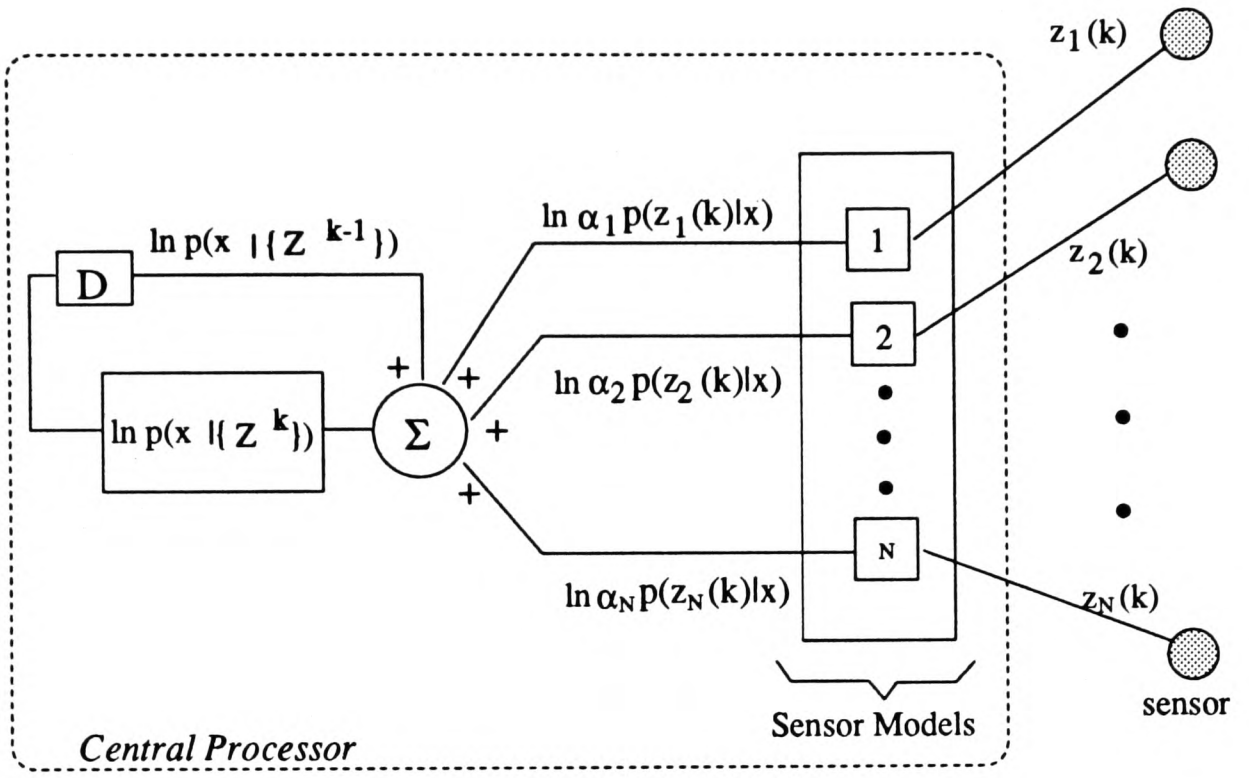


Figure 3.3: Centralized information update for architecture with sensor communicating actual observations.

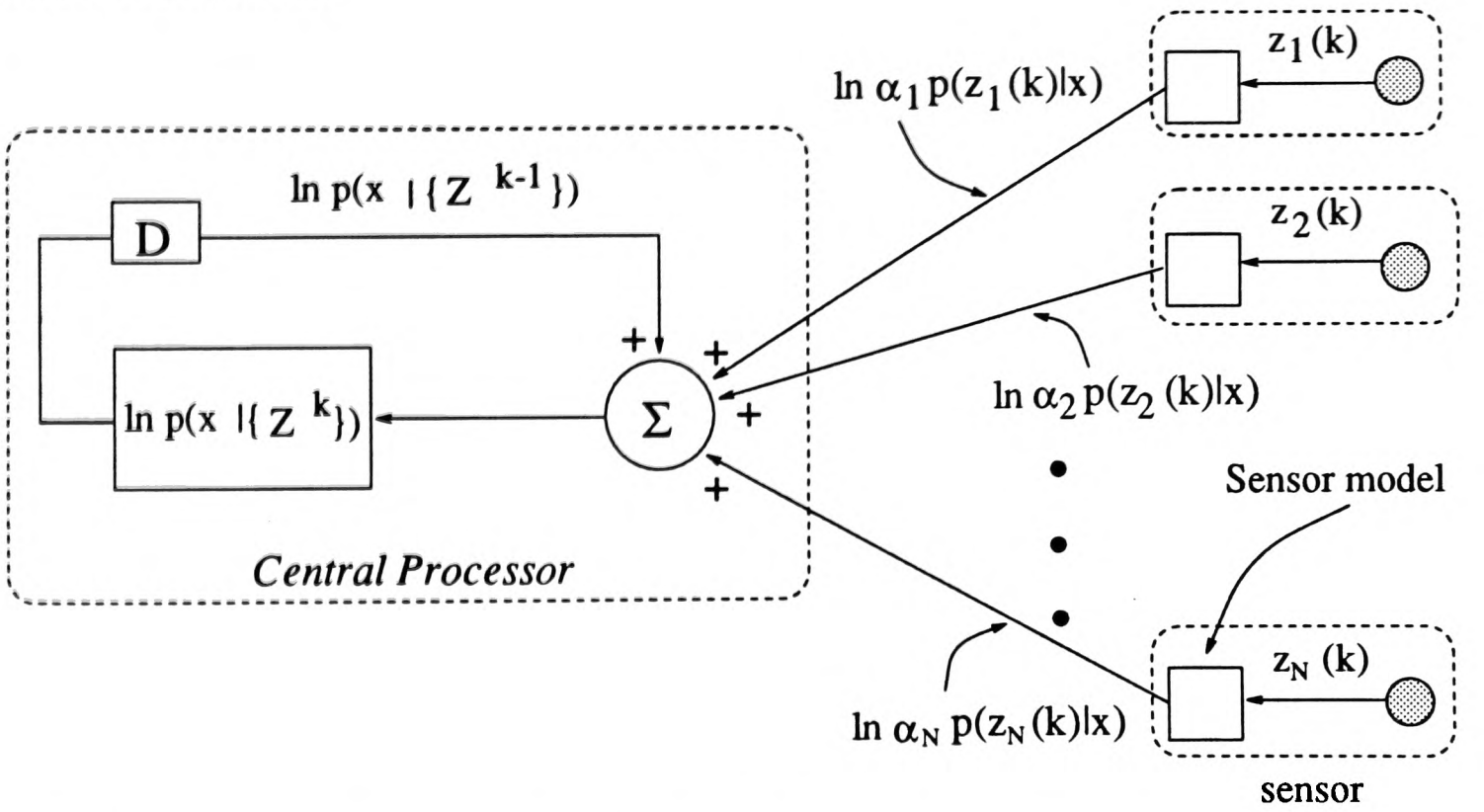


Figure 3.4: Centralized information update for architecture communicating likelihoods to central processor.

chies of Figure 3.4 and Figure 3.2 have the added advantage that a sensor can be changed or modified without the central processor needing to have any knowledge of it. This makes the system modular and facilitates the inclusion of a wide range of sensor complexity in the system because any complex signal processing necessary for a given sensor can be realized locally. In Figure 3.2 initialization is done locally due to the fact that the prior is subjective at each sensor node. (See discussion in Section 2.2, Chapter 2). The architectures of Figures 3.4 and 3.2 lead to the development of fully decentralized architectures.

3.1.3 Distributed and Decentralized Architectures

Each sensor in a decentralized system, in addition to making observations, is required to obtain the global posterior after communicating with all the other sensors. For a fully connected system, the global posterior is expected to be *identical* at each sensor. In order to formulate this architecture precisely, we define a local state vector

$$\mathbf{x}_i \in \mathcal{X}, \quad \text{associated with sensor } i. \tag{3.5}$$

The decentralized architecture is a result of distributing the computations performed by the hierarchical architectures we have just considered, so that each sensor node is able to perform these and obtain the global result. As before, we consider the decentralized architectures resulting from the Independent Opinion Pool and the Independent Likelihood Pool respectively:

1. **Independent Opinion Pool.** Consideration of the Independent Opinion Pool from Chapter 2, and the architecture suggested by the hierarchy of Equation 3.2 leads to a decentralized architecture. As before, the implicit assumption in the resulting architecture is that the observations conditioned on the state are independent *and*

each sensor node has subjective prior information. A decentralized architecture of this form can be realized by placing the central processor of Equation 3.2 (Figure 3.2) at each sensor node. In this way, each sensor node i communicates a local posterior $p(\mathbf{x}_i \mid \mathbf{Z}_i^k)$ and obtains the global posterior according to

$$ln\ p(\mathbf{x}_i \mid \{\mathbf{Z}^k\}) = \sum_j \underbrace{ln\ p(\mathbf{x}_j \mid \mathbf{Z}_j^k)}_{\text{communicated}}, \tag{3.6}$$

where the local posterior for each sensor node i is obtained as follows

$$ln\ p(\mathbf{x}_i \mid \mathbf{Z}_i^k) = ln\ p(\mathbf{x}_i \mid \mathbf{Z}_i^{k-1}) + ln\ [\alpha_i\ p(\mathbf{z}_i(k) \mid \mathbf{x}_i)]\,. \tag{3.7}$$

A disadvantage of this architecture is that each sensor node has access only to partial posterior information at each sensor node (which has been communicated) and does not have the observed likelihood information at each sensor node. Therefore no decision theoretic approach to management which requires this information can be implemented locally at each node.

2. **Independent Likelihood Pool.** In a decentralized system based on the Independent Likelihood Pool, communication is in the form of likelihood information [2]. At each sensor node all the likelihoods are fused to give the global posterior. This amounts to placing the central processor of Equation 3.4 at each sensor node. The global posterior at sensor node i is obtained using the following log update relationship

$$ln\ p(\mathbf{x}_i \mid \{\mathbf{Z}^k\}) = ln\ p(\mathbf{x}_i \mid \{\mathbf{Z}^{k-1}\}) + \sum_j \underbrace{ln\ [\alpha_j\ p(\mathbf{z}_j(k) \mid \mathbf{x}_j)]}_{\text{communicated}}. \tag{3.8}$$

This is illustrated for each sensor node in Figure 3.5.

[2] An important consequence of architectures where likelihoods are communicated is that a reduced order state can be used locally at each information source. This has been described in the work by Alouani [4] following on from the work described in [128][34][136].

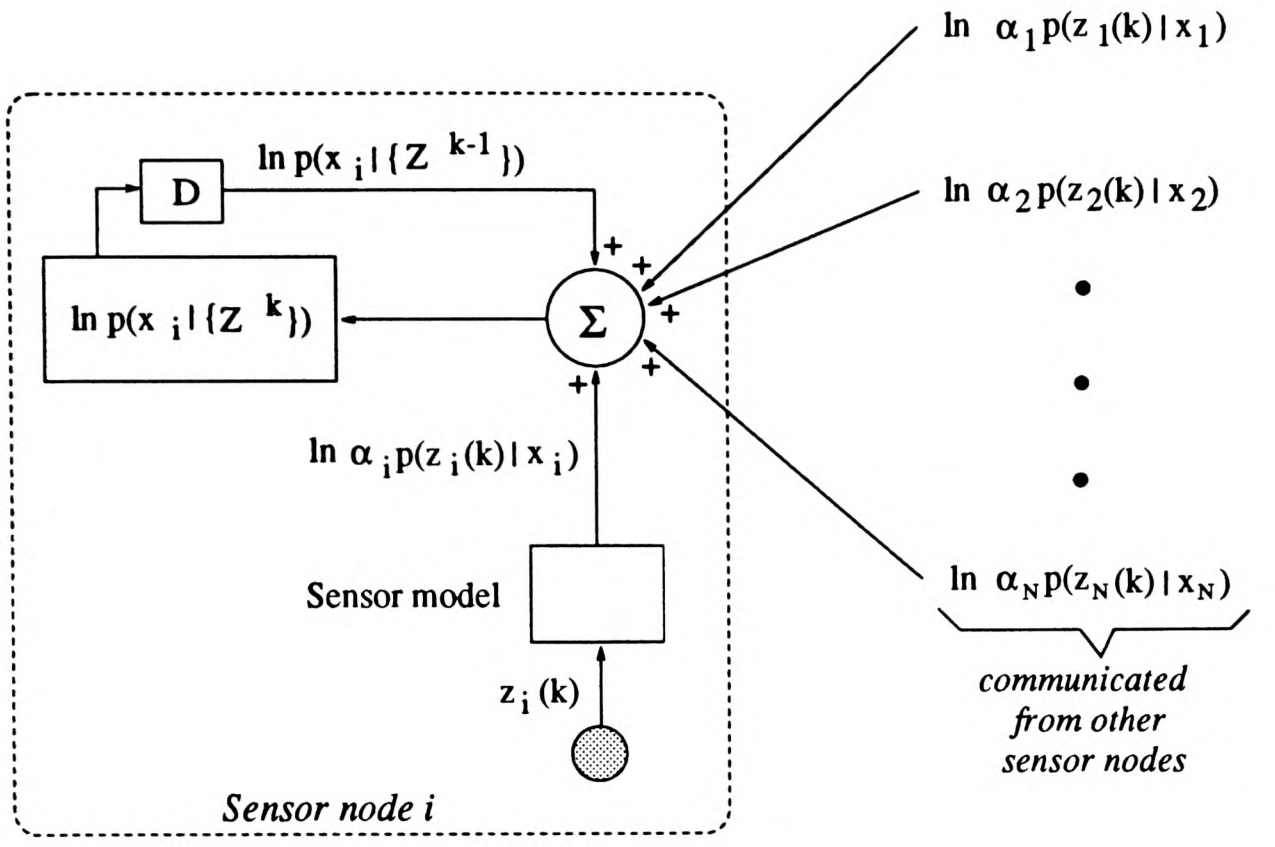


Figure 3.5: Decentralized information update at sensor node i. Sensor i communicates its likelihood to all the other sensors and in turn receives likelihoods from all the other sensors as shown.

In addition, each sensor node i can also compute a *local partial posterior* based only on local observation information and a global prior, that is

$$ln\ p(\mathbf{x}_i \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k)) = ln\ p(\mathbf{x}_i \mid \{\mathbf{Z}^{k-1}\}) + ln\ [\alpha_i\ p(\mathbf{z}_i(k) \mid \mathbf{x}_i)]\,. \tag{3.9}$$

This partial posterior has very important consequences because of what it represents, that is, a summation of information known globally before time k and the information that sensor i contributes at time k. This is invaluable if it is required to evaluate the significance of sensor i's observation, its impact and contribution to the global posterior *et cetera*. This is exploited in our development of sensor management strategies in Chapter 4.

3.1.4 Discussion

All the multi-sensor architectures above are mathematically equivalent for fully connected systems but differ in the way they combine probabilistic information as discussed in Section 2.2. The decentralized systems are more desirable in terms of robustness, performance and reliability [50] as discussed in Chapter 1.

Given that each sensor node has subjective prior information, an interesting point to note about architectures based on the Independent Opinion Pool (Equations 3.2 and 3.6) is that they allow the implementation of hybrid algorithms as follows; each sensor can locally compute its partial posterior using any method for generating a posterior locally and then the global result is obtained by summing up the partial posterior information from all the nodes.

3.2 Continuous State Estimation from Information Update

The Kalman filter is an algorithm for recursively estimating a state $\mathbf{x}(k)$ given a set of uncertain observations up to time k, denoted $\mathbf{Z}^k$. The standard Kalman filter (KF) formulation can be rewritten in terms of the inverse covariances as the so called information filter. What follows is a derivation of the information filter *directly* from the recursive probabilistic information update of Chapter 2.

3.2.1 Information Filter

We start by making the assumptions fundamental to the derivation of the information filter. We then outline the method that we shall use to derive the various architectural forms of the information filter.

Observation

We assume a linear observation model based on the general observation model of Equation 2.1, that is

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k), \tag{3.10}$$

where $\mathbf{H}(k)$ is an observation matrix relating observations $\mathbf{z}(k)$ in the observation space $\mathcal{Z}$, to the state $\mathbf{x}(k)$ in the state space $\mathcal{X}$. The observation noise is represented by $\mathbf{v}(k)$ which is zero mean Gaussian with covariance matrix $\mathbf{R}(k)$. The noise $\mathbf{v}(k)$ is a process which is uncorrelated in time, that is

$$E\{\mathbf{v}(k)\} = 0, \quad E\{\mathbf{v}(k)\mathbf{v}(j)^T\} = \mathbf{R}(k)\delta_{k,j}, \quad \forall k, j \tag{3.11}$$

where δ is the Kronecker delta operator. The covariance matrix $\mathbf{R}(k)$ is positive and semi-definite and the off-diagonal terms are cross-correlation terms.

State Transition

We assume a linear state transition of the form

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{w}(k), \tag{3.12}$$

where $\mathbf{F}(k)$ is the state transition matrix describing the transition of the state from time-step to time-step and $\mathbf{w}(k)$ is the state transition noise which is assumed to be zero-mean Gaussian with covariance matrix $\mathbf{Q}(k)$. The transition noise is uncorrelated in time and therefore satisfies

$$E\{\mathbf{w}(k)\} = 0, \quad E\{\mathbf{w}(k)\mathbf{w}(j)^T\} = \mathbf{Q}(k)\delta_{k,j}, \quad \forall k, j \tag{3.13}$$

where $\mathbf{Q}(k)$ is positive and semi-definite.

We also make the additional assumption that the observation noise $\mathbf{v}(k)$ and the state transition noise $\mathbf{w}(k)$ are uncorrelated, that is

$$E\{\mathbf{v}(k)\mathbf{w}(j)\} = 0, \quad \forall k, j. \tag{3.14}$$

Information update and Estimation

Given the probabilistic model of Chapter 2, the starting point is the information update relationship. We can write this in the equivalent form using log-likelihoods as follows

$$l \triangleq ln\ p(\mathbf{x}(k) \mid \mathbf{Z}^k) = ln\ p(\mathbf{x}(k) \mid \mathbf{Z}^{k-1}) + ln[\alpha\ p(\mathbf{z}(k) \mid \mathbf{x}(k))]. \tag{3.15}$$

We make the assumption that all the PDFs in Equation 3.15 are Gaussian such that for a distribution $p(\mathbf{x})$, we have that

$$p(\mathbf{x}) = N(\bar{\mathbf{x}}, \mathbf{P}), \quad \text{where } \bar{\mathbf{x}} = \text{mean and } \mathbf{P} = \text{covariance.} \tag{3.16}$$

We now require to recursively estimate the state, that is, obtain an MMSE estimate for $\mathbf{x}(k)$ given all the observations up to time-step k, which we shall denote $\hat{\mathbf{x}}(k \mid k)$. From the posterior of Equation 3.15, an MMSE estimate is obtained from Equations 2.12 and 2.13 as

$$\hat{\mathbf{x}}(k \mid k) = E\left\{\mathbf{x}(k) \mid \mathbf{Z}^k\right\}, \tag{3.17}$$

with covariance

$$\mathbf{P}(k \mid k) = E\left\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mid \mathbf{Z}^k\right\}. \tag{3.18}$$

Corresponding to this estimate, we define a transformed state vector which we call the *information state vector*. The information state vector at time step j given l is given by

$$\hat{\mathbf{y}}(j \mid l) \triangleq \mathbf{P}^{-1}(j \mid l)\hat{\mathbf{x}}(j \mid l). \tag{3.19}$$

Associated with the information state vector, is the *information matrix* given by the inverse covariance $\mathbf{P}^{-1}(j \mid l)$. It is important to note that $\mathbf{P}^{-1}(j \mid l)$ is actually the covariance of the information state vector $\mathbf{P}^{-1}(j \mid l)\hat{\mathbf{x}}(j \mid l)$.

These information estimates can be obtained based on the above assumptions by; (i) considering the probabilistic information update relationship l of Equation 3.15, and substituting in the appropriate Gaussian PDFs (ii) evaluating $\nabla_{\mathbf{x}} l$ and (iii) $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}{}^{T} l$. We now go through these steps and derive the information filter.

Derivation

Writing l in terms of Gaussian PDFs, we have that the score function $\nabla_{\mathbf{x}} l$ is evaluated from

$$\begin{aligned}
\nabla_{\mathbf{x}} l &\triangleq \nabla_{\mathbf{x}} \left[c_1 - \frac{1}{2}(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mathbf{P}^{-1}(k \mid k)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \right] \\
&= \nabla_{\mathbf{x}} \left[c_2 - \frac{1}{2}(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T \mathbf{P}^{-1}(k \mid k-1)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)) \right] \\
&\quad + \nabla_{\mathbf{x}} \left[c_3 - \frac{1}{2}(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k))^T \mathbf{R}^{-1}(k)(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k)) \right], \qquad (3.20)
\end{aligned}$$

where c_1, c_2 and c_3 are constants independent of $\mathbf{x}(k)$. Making use of the result $\nabla_{\mathbf{x}}(\mathbf{y}^T \mathbf{A} \mathbf{y}) = 2(\nabla_{\mathbf{x}} \mathbf{y}^T)\mathbf{A}\mathbf{y}$, where $\mathbf{A}$ is symmetric and independent of $\mathbf{x}$, we have that the left-hand side of Equation 3.20 corresponding to the posterior, simplifies to

$$\begin{aligned}
\nabla_{\mathbf{x}} & \left[c_1 - \tfrac{1}{2}(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mathbf{P}^{-1}(k \mid k)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \right] \\
&= -\left[\nabla_{\mathbf{x}} (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \mathbf{P}^{-1}(k \mid k)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \right] \\
&= -\mathbf{P}^{-1}(k \mid k)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)), \qquad (3.21)
\end{aligned}$$

and for the term corresponding to the prior

$$\begin{aligned}
\nabla_{\mathbf{x}} & \left[c_2 - \tfrac{1}{2}(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T \mathbf{P}^{-1}(k \mid k-1)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)) \right] \\
&= -\mathbf{P}^{-1}(k \mid k-1)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)), \qquad (3.22)
\end{aligned}$$

and similarly for that corresponding to the likelihood

$$\begin{aligned}\nabla_{\mathbf{x}} & \left[c_3 - \tfrac{1}{2}(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k))^T \mathbf{R}^{-1}(k)(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k))\right] \\ = & -\left[\nabla_{\mathbf{x}}(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k))^T\right] \mathbf{R}^{-1}(k)(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k)) \\ = & \quad \mathbf{H}(k)^T \mathbf{R}^{-1}(k)(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k)). \end{aligned} \tag{3.23}$$

Substituting these in Equation 3.20, we have

$$\begin{aligned}-\mathbf{P}^{-1}(k \mid k)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) = & -\mathbf{P}^{-1}(k \mid k-1)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \\ & +\mathbf{H}(k)^T \mathbf{R}^{-1}(k)(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k)). \end{aligned} \tag{3.24}$$

From the definition of Fisher information (Equations 2.47 and 2.48), $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}{}^T l$ gives the Fisher information matrix. Therefore applying $\nabla_{\mathbf{x}}$ again to Equation 3.24 evaluates to

$$\mathbf{P}^{-1}(k \mid k) = \mathbf{P}^{-1}(k \mid k-1) + \mathbf{H}(k)^T \mathbf{R}^{-1}(k)\mathbf{H}(k). \tag{3.25}$$

Equation 3.25 post-multiplied by $\hat{\mathbf{x}}(k \mid k)$ and added to Equation 3.24 gives

$$\mathbf{P}^{-1}(k \mid k)\hat{\mathbf{x}}(k \mid k) = \mathbf{P}^{-1}(k \mid k-1)\hat{\mathbf{x}}(k \mid k-1) + \mathbf{H}(k)^T \mathbf{R}^{-1}(k)\mathbf{z}(k). \tag{3.26}$$

Using the definition of the information state vector (Equation 3.19), we can write Equation 3.26 as

$$\hat{\mathbf{y}}(k \mid k) = \hat{\mathbf{y}}(k \mid k-1) + \mathbf{H}(k)^T \mathbf{R}^{-1}(k)\mathbf{z}(k), \tag{3.27}$$

which is the update equation for the information state vector. The predicted covariance $\mathbf{P}(k \mid k-1)$ is obtained from a consideration of its definition

$$\mathbf{P}(k \mid k-1) \triangleq E\left\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T \mid \mathbf{Z}^{k-1}\right\}. \tag{3.28}$$

Since

$$(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)) = \mathbf{F}(k)\left[(\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1))\right] + \mathbf{w}(k),$$

the predicted covariance becomes

$$\mathbf{P}(k \mid k-1) = \mathbf{F}(k)\mathbf{P}(k-1 \mid k-1)\mathbf{F}(k)^T + \mathbf{Q}(k). \tag{3.29}$$

And so the predicted information state vector $\hat{\mathbf{y}}(k \mid k-1)$, is obtained as

$$\begin{aligned}\hat{\mathbf{y}}(k \mid k-1) &= \mathbf{P}^{-1}(k \mid k-1)\mathbf{F}(k)\hat{\mathbf{x}}(k-1 \mid k-1) \\ &= \mathbf{P}^{-1}(k \mid k-1)\mathbf{F}(k)\mathbf{P}(k-1 \mid k-1)\hat{\mathbf{y}}(k-1 \mid k-1).\end{aligned} \tag{3.30}$$

It will be noted that in the above formulation, the vector $\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{z}(k)$ is a sufficient statistic of $\mathbf{x}(k)$. That $\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{z}(k)$ is a sufficient statistic for $\mathbf{x}(k)$ can be seen by writing the Gaussian $[\alpha\, p(\mathbf{z}(k) \mid \mathbf{x})]$ (see Equation 3.20) in the form

$$\begin{aligned}p(\mathbf{z}(k) \mid \mathbf{x}) \propto\ & c\exp\left\{-\frac{1}{2}\mathbf{x}(k)^T\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{H}(k)\mathbf{x}(k)\right\}\exp\left\{-\frac{1}{2}\mathbf{z}(k)^T\mathbf{R}^{-1}(k)\mathbf{z}(k)\right\} \\ & \times\exp\left\{\mathbf{x}(k)^T\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{z}(k)\right\},\end{aligned} \tag{3.31}$$

which by the factorization theorem [87][3] is the necessary condition for $\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{z}(k)$ to be sufficient for $\mathbf{x}(k)$. The statistic is normal with covariance $\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{H}(k)$. It has been shown in [121], that this sufficient statistic is unique (complete) and also minimal.

Non-linear information filter

Often the observation and transition equations are not linear, in this case we can derive a non-linear information filter. For a non-linear system the observation equation (based on Equation 2.1) is written

$$\mathbf{z}(k) = \mathbf{h}[k, \mathbf{x}(k)] + \mathbf{v}(k), \tag{3.32}$$

[3] A vector $\mathbf{t}$ is sufficient for the vector $\mathbf{x}$ iff $p(\mathbf{z} \mid \mathbf{x}) = p(\mathbf{z})p(\mathbf{t} \mid \mathbf{x})$, and the distribution $p(\mathbf{z} \mid \mathbf{x})$ is normal [121].

where $\mathbf{h}[\cdot]$ is the non-linear observation model transforming $\mathbf{x}(k)$ from the state space $\mathcal{X}$, to the observation space $\mathcal{Z}$. The observation noise is assumed to be additive and is of the same form as in Equation 3.11. Similarly, we can write the non-linear state transition equation

$$\mathbf{x}(k) = \mathbf{f}[k, \mathbf{x}(k-1)] + \mathbf{w}(k), \tag{3.33}$$

where $\mathbf{f}[\cdot]$ is the non-linear state transition model describing the transition of the state from time-step to time-step as a non-linear function of the state.

The derivation proceeds as outlined in the previous section by first substituting the appropriate Gaussians into Equation 3.15. However, the Gaussian substitution corresponding to the likelihood $ln[\alpha\ p(\mathbf{z}(k)) \mid \mathbf{x}(k)]$ is different from that in Equation 3.20 and reflects the non-linearities in Equation 3.32. It is given by

$$ln[\alpha\ p(\mathbf{z}(k) \mid \mathbf{x}(k))] = c - \frac{1}{2}(\mathbf{z}(k) - \mathbf{h}[k, \mathbf{x}(k)])^T \mathbf{R}^{-1}(k)(\mathbf{z}(k) - \mathbf{h}[k, \mathbf{x}(k)]). \tag{3.34}$$

Proceeding as before by taking (i) $\nabla_{\mathbf{x}} l$ and (ii) $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}{}^T l$ and making the appropriate substitutions we obtain the result that the information state vector is updated according to

$$\hat{\mathbf{y}}(k \mid k) = \hat{\mathbf{y}}(k \mid k-1) + \nabla_{\mathbf{x}} \mathbf{h}[k, \hat{\mathbf{x}}(k \mid k)]^T \mathbf{R}^{-1}(k)\mathbf{z}(k), \tag{3.35}$$

and the inverse covariance (information matrix) according to

$$\mathbf{P}^{-1}(k \mid k) = \mathbf{P}^{-1}(k \mid k-1) + \nabla_{\mathbf{x}} \mathbf{h}[k, \hat{\mathbf{x}}(k \mid k)]^T \mathbf{R}^{-1}(k) \nabla_{\mathbf{x}} \mathbf{h}[k, \hat{\mathbf{x}}(k \mid k)]. \tag{3.36}$$

It can be shown that

$$\begin{aligned} (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)) \;=\; & \nabla_{\mathbf{x}} \mathbf{f}[k, \hat{\mathbf{x}}(k-1 \mid k-1)](\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)) \\ & + \underbrace{\cdots}_{\text{higher order terms}} + \mathbf{w}(k), \end{aligned} \tag{3.37}$$

based on a linearization about the predicted state $\hat{\mathbf{x}}(k \mid k-1)$, where the higher order terms can be obtained from a Taylor series expansion [12]. Therefore, by ignoring the higher order terms, the predicted covariance is given by

$$\begin{aligned}\mathbf{P}(k \mid k-1) &= \nabla_{\mathbf{x}}\mathbf{f}[k, \hat{\mathbf{x}}(k-1 \mid k-1)]\mathbf{P}(k-1 \mid k-1)\, \nabla_{\mathbf{x}}\mathbf{f}[k, \hat{\mathbf{x}}(k-1 \mid k-1)]^T \\ &\quad + \mathbf{Q}(k). \end{aligned} \tag{3.38}$$

Consequently, the predicted information state vector is obtained as

$$\hat{\mathbf{y}}(k \mid k-1) = \mathbf{P}^{-1}(k \mid k-1)\mathbf{f}[k, \hat{\mathbf{x}}(k-1 \mid k-1)]. \tag{3.39}$$

It must be noted that to evaluate $\nabla_{\mathbf{x}}\mathbf{h}[k, \hat{\mathbf{x}}(k \mid k-1)]$, $\hat{\mathbf{x}}(k \mid k-1)$ is obtained from

$$\hat{\mathbf{x}}(k \mid k-1) = \mathbf{P}(k \mid k-1)\hat{\mathbf{y}}(k \mid k-1). \tag{3.40}$$

In the formulation of the non-linear information filter, the following equivalences with the linear information filter are apparent

$$\nabla_{\mathbf{x}}\mathbf{h}[k, \hat{\mathbf{x}}(k \mid k)] \equiv \mathbf{H}(k) \quad \text{and} \quad \nabla_{\mathbf{x}}\mathbf{f}[k, \hat{\mathbf{x}}(k \mid k)] \equiv \mathbf{F}(k). \tag{3.41}$$

Therefore substituting $\mathbf{H}(k)$ for $\nabla_{\mathbf{x}}\mathbf{h}[k, \hat{\mathbf{x}}(k \mid k)]$ and $\mathbf{F}(k)$ for $\nabla_{\mathbf{x}}\mathbf{f}[k, \hat{\mathbf{x}}(k \mid k)]$, makes the formulation of the first order non-linear information filter the same as for the linear information filter (with the exception of Equation 3.39).

The information filter just derived forms the basis of a state estimation algorithm which can be applied to any of the architectural paradigms described in Section 3.1. In the above form, the algorithm is implemented on the single sensor system of Figure 3.1. We can summarize the information filter as follows:

Summary

Information Filter Equations

The information state vector is updated according to

$$\hat{\mathbf{y}}(k \mid k) = \hat{\mathbf{y}}(k \mid k-1) + \mathbf{H}(k)^T \mathbf{R}^{-1}(k)\mathbf{z}(k), \tag{3.42}$$

and the inverse covariance (information matrix) according to

$$\mathbf{P}^{-1}(k \mid k) = \mathbf{P}^{-1}(k \mid k-1) + \mathbf{H}(k)^T \mathbf{R}^{-1}(k)\mathbf{H}(k). \tag{3.43}$$

The inverse covariance and the information state vector are predicted recursively as

$$\mathbf{P}(k \mid k-1) = \mathbf{f}(k)\mathbf{P}(k-1 \mid k-1)\mathbf{F}(k)^T + \mathbf{Q}(k), \tag{3.44}$$

(3.45)

$$\hat{\mathbf{y}}(k \mid k-1) = \mathbf{P}^{-1}(k \mid k-1)\mathbf{F}(k)\mathbf{P}(k-1 \mid k-1)\hat{\mathbf{y}}(k-1 \mid k-1), \tag{3.46}$$

respectively. The estimated state may be obtained from

$$\hat{\mathbf{x}}(k \mid k) = \mathbf{P}(k \mid k)\hat{\mathbf{y}}(k \mid k). \tag{3.47}$$

3.2.2 Hierarchical Information Filter

Hierarchical estimation has previously been discussed based on the usual formulation of the Kalman filter [34][63]. What we present here are methods based on deriving the appropriate form of the information filter from the information update. For the whole system we make the following definitions concerning the observation partitioning

$$\begin{aligned}
\text{observations } \mathbf{z}(k) &= [\mathbf{z}_1(k)^T, \mathbf{z}_2(k)^T, \ldots, \mathbf{z}_N(k)^T]^T \\
\text{observation matrix } \mathbf{H}(k) &= [\mathbf{H}_1(k)^T, \mathbf{H}_2(k)^T, \ldots, \mathbf{H}_N(k)^T]^T \\
\text{observation noise covariance } \mathbf{R}(k) &= diag[\mathbf{R}_1(k), \mathbf{R}_2(k), \ldots, \mathbf{R}_N(k)],
\end{aligned} \tag{3.48}$$

where the observation partitions are assumed to be uncorrelated. A centralized or hierarchical architecture is described by Equations 3.4 and 3.2 (see Figures 3.3 and 3.2). We now discuss each of these in turn (derivations here are only in outline):

1. For the hierarchy described by Equation 3.4, where each sensor node communicates likelihood information, the log-likelihood l, is given by

$$ln\ p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = ln\ p(\mathbf{x} \mid \{\mathbf{Z}^{k-1}\}) + \sum_j \underbrace{ln\ [\alpha_j\ p(\mathbf{z}_j(k) \mid \mathbf{x})]}_{\text{communicated}}.$$

Each sensor j makes an observation according to

$$\mathbf{z}_j(k) = \mathbf{H}_j(k)\mathbf{x}(k) + \mathbf{v}_j(k), \quad \forall j \in \mathcal{N}, \tag{3.49}$$

where the observation noise $\mathbf{v}_j(k)$ is under the same assumptions as in Equation 3.11.

In the information filter derived in Section 3.2.1, information from the observation is $\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{H}(k)$ which in this case is obtained from the summation

$$\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{z}(k) = \sum_j \mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)\mathbf{z}_j(k). \tag{3.50}$$

Similarly

$$\mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{H}(k) = \sum_j \mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)\mathbf{H}_j(k), \tag{3.51}$$

for the observation information matrix. Hence each sensor communicates to the central processor $\mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)\mathbf{z}_j(k)$ and $\mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)\mathbf{H}_j(k)$. And so the global information state vector is given by

$$\hat{\mathbf{y}}(k \mid k) = \hat{\mathbf{y}}(k \mid k-1) + \mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{z}(k), \tag{3.52}$$

and the inverse covariance (information matrix) by

$$\mathbf{P}^{-1}(k \mid k) = \mathbf{P}^{-1}(k \mid k-1) + \mathbf{H}(k)^T\mathbf{R}^{-1}(k)\mathbf{H}(k), \tag{3.53}$$

using Equations 3.50 and 3.51. The predictions $\mathbf{P}^{-1}(k \mid k-1)$ and $\hat{\mathbf{y}}(k \mid k-1)$ as before are given by Equations 3.46 and 3.45 respectively.

2. For an architecture defined by Equation 3.2, where each sensor i communicates a local posterior $p(\mathbf{x} \mid \mathbf{Z}_i^k)$, the global posterior is computed from

$$ln\ p(\mathbf{x} \mid \{\mathbf{Z}^k\}) = \sum_j \underbrace{ln\ p(\mathbf{x} \mid \mathbf{Z}_j^k)}_{\text{communicated}},$$

where for each sensor node i, the local posterior is given by

$$ln\ p(\mathbf{x} \mid \mathbf{Z}_i^k) = ln\ p(\mathbf{x} \mid \mathbf{Z}_i^{k-1}) + ln\ [\alpha_i\ p(\mathbf{z}_i(k) \mid \mathbf{x})], \tag{3.54}$$

provided that each sensor has available subjective prior information. By following the steps outlined in Section 3.2.1, the corresponding filter for this architecture can be derived. The global information state vector is obtained as

$$\hat{\mathbf{y}}(k \mid k) = \sum_j \hat{\mathbf{y}}_j(k \mid k), \tag{3.55}$$

and the information matrix

$$\mathbf{P}^{-1}(k \mid k) = \sum_j \mathbf{P}_j^{-1}(k \mid k), \tag{3.56}$$

where the local information state vectors $\hat{\mathbf{y}}_i(k \mid k)$, are given by

$$\hat{\mathbf{y}}_i(k \mid k) = \hat{\mathbf{y}}_i(k \mid k-1) + \mathbf{H}_i^T(k)\mathbf{R}_i^{-1}(k)\mathbf{z}_i(k), \tag{3.57}$$

and the local information matrices by

$$\mathbf{P}_i^{-1}(k \mid k) = \mathbf{P}_i^{-1}(k \mid k-1) + \mathbf{H}_i^T(k)\mathbf{R}_i^{-1}(k)\mathbf{H}_i(k). \tag{3.58}$$

The predictions are local versions of those in Section 3.2.1, that is

$$\mathbf{P}_i(k \mid k-1) = \mathbf{F}_i(k)\mathbf{P}_i(k-1 \mid k-1)\mathbf{F}_i(k)^T + \mathbf{Q}_i(k), \tag{3.59}$$

for the covariance and

$$\hat{\mathbf{y}}_i(k \mid k-1) = \mathbf{P}_i^{-1}(k \mid k-1)\mathbf{F}_i(k)\mathbf{P}_i(k-1 \mid k-1)\hat{\mathbf{y}}_i(k-1 \mid k-1), \qquad (3.60)$$

for the information state vector.

The above filters can be extended to non-linear estimation in the same way as was done in Section 3.2.1 for the single sensor information filter.

3.2.3 Decentralized Information Filter

Of particular interest in this thesis is the decentralized architecture. We can develop a decentralized information filter *directly* from probabilistic information update. The requirement is to implement a filter such that each sensor node is able to derive the global state as illustrated in Figure 3.5. We proceed as before by first stating the assumptions in the derivation.

Observation

Using the definition of a local state (Equation 3.5), we extend this to a state vector $\mathbf{x}_i(k)$ associated with each sensor node i. As in Equation 3.48, the complete observation vector $\mathbf{z}(k)$ is partitioned into sub-vectors corresponding to each nodes' observation; $\mathbf{z}(k) = [\mathbf{z}_1^T(k), \cdots, \mathbf{z}_N^T(k)]^T$ based on the assumption that observation partitions are uncorrelated. The observation matrix $\mathbf{H}(k)$, and the noise vector $\mathbf{v}(k)$, are partitioned in a similar manner. The nodal observation equation is thus written

$$\mathbf{z}_i(k) = \mathbf{H}_i(k)\mathbf{x}_i(k) + \mathbf{v}_i(k). \qquad (3.61)$$

We make the usual assumptions about the nodal observation noise, that is, that $\mathbf{v}_i(k)$ is uncorrelated

$$E\{\mathbf{v}_i(k)\} = 0, \quad E\left\{\mathbf{v}_i(k)\mathbf{v}_i(j)^T\right\} = \mathbf{R}_i(k)\delta_{k,j}, \ \forall k, j. \tag{3.62}$$

State Transition

The transition of the state $\mathbf{x}_i(k)$ is described by the equation

$$\mathbf{x}_i(k) = \mathbf{F}_i(k)\mathbf{x}_i(k-1) + \mathbf{w}_i(k), \tag{3.63}$$

where $\mathbf{F}_i(k)$ is the nodal state transition matrix and $\mathbf{w}_i(k)$ is the nodal state transition noise. The state transition noise is under the usual assumptions, that is, uncorrelated

$$E\{\mathbf{w}_i(k)\} = 0, \quad E\left\{\mathbf{w}_i(k)\mathbf{w}_i(j)^T\right\} = \mathbf{Q}_i(k)\delta_{k,j}, \ \forall k, j. \tag{3.64}$$

There is also the additional assumption that the observation noise and the state transition noise are uncorrelated

$$E\{\mathbf{v}_j(k)\mathbf{w}_j(k)\} = 0, \quad \forall j \in \mathcal{N}. \tag{3.65}$$

Information update and Estimation

We make use of the Bayes information update for an architecture which corresponds to the Independent Likelihood Pool of Chapter 2, that is

$$l_i \triangleq ln\ p(\mathbf{x}_i \mid \{\mathbf{Z}^k\}) = ln\ p(\mathbf{x}_i \mid \{\mathbf{Z}^{k-1}\}) + \sum_j \underbrace{ln\ [\alpha_j\ p(\mathbf{z}_j(k) \mid \mathbf{x}_j)]}_{\text{communicated}}, \tag{3.66}$$

where the observation set $\{\mathbf{Z}^k\}$ is as defined in Equation 2.16. From this posterior, an MMSE estimate is obtained at each sensor i as follows

$$\hat{\mathbf{x}}_i(k \mid k) = E\left\{\mathbf{x}_i(k) \mid \{\mathbf{Z}^k\}\right\}, \tag{3.67}$$

with covariance

$$\mathbf{P}_i(k \mid k) = E\left\{(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k))(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k))^T \mid \{\mathbf{Z}^T\}\right\}. \tag{3.68}$$

We define the information state vector for sensor i at time step j given l as

$$\hat{\mathbf{y}}_i(j \mid l) \triangleq \mathbf{P}_i^{-1}(j \mid l)\hat{\mathbf{x}}_i(j \mid l), \tag{3.69}$$

and sensor i's information matrix is given by the inverse covariance $\mathbf{P}_i^{-1}(j \mid l)$ which is the covariance of the information state vector $\mathbf{P}_i^{-1}(j \mid l)\hat{\mathbf{x}}_i(j \mid l)$. As before, all the PDFs are assumed to be Gaussian and the derivation proceeds by following the steps outlined in Section 3.2.1, but making use of the above assumptions and l_i.

Derivation

Substituting the appropriate Gaussians in l_i and evaluating $\nabla_{\mathbf{x}} l_i$ we have that

$$\begin{aligned}
\nabla_{\mathbf{x}_i} l_i &\triangleq \nabla_{\mathbf{x}_i}\left[c_1 - \frac{1}{2}(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k))^T \mathbf{P}_i^{-1}(k \mid k)(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k))\right] \\
&= \nabla_{\mathbf{x}_i}\left[c_2 - \frac{1}{2}(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k-1))^T \mathbf{P}_i^{-1}(k \mid k-1)(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k-1))\right] \\
&\quad + \sum_j \nabla_{\mathbf{x}_j}\left[c_3 - \frac{1}{2}(\mathbf{z}_j(k) - \mathbf{H}_j(k)\mathbf{x}_j(k))^T \mathbf{R}_j^{-1}(k)(\mathbf{z}_j(k) - \mathbf{H}_j(k)\mathbf{x}_j(k))\right].
\end{aligned} \tag{3.70}$$

Using results obtained before, this simplifies to

$$\begin{aligned}
-\mathbf{P}_i^{-1}(k \mid k)(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k)) &= -\mathbf{P}_i^{-1}(k \mid k-1)(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k)) \\
&\quad + \sum_j \mathbf{H}_j(k)^T \mathbf{R}_j^{-1}(k)(\mathbf{z}_j(k) - \mathbf{H}_j(k)\mathbf{x}_j(k)).
\end{aligned} \tag{3.71}$$

Taking $\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^T l_i$ gives the nodal Fisher information, that is, the information matrix. And so applying $\nabla_{\mathbf{x}_i}$ again we have

$$\nabla_{\mathbf{x}_i}\left[-\mathbf{P}_i^{-1}(k \mid k)(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k))\right] = \nabla_{\mathbf{x}_i}\left[-\mathbf{P}_i^{-1}(k \mid k-1)(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid k))\right]$$

$$+\sum_j \nabla_{\mathbf{x}_j}\left[\mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)(\mathbf{z}_j(k)-\mathbf{H}_j(k)\mathbf{x}_j(k))\right], \tag{3.72}$$

which simplifies to

$$\mathbf{P}_i^{-1}(k\mid k)=\mathbf{P}_i^{-1}(k\mid k-1)+\sum_j \mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)\mathbf{H}_j(k). \tag{3.73}$$

Post-multiplying Equation 3.73 by $\hat{\mathbf{x}}_i(k\mid k)$ and adding to Equation 3.71 gives

$$\mathbf{P}_i^{-1}(k\mid k)\hat{\mathbf{x}}_i(k\mid k)=\mathbf{P}_i^{-1}(k\mid k-1)\hat{\mathbf{x}}_i(k\mid k-1)+\sum_j \mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)\mathbf{z}_j(k). \tag{3.74}$$

Using the definition of the nodal information state vector, we can write Equation 3.74 as

$$\hat{\mathbf{y}}_i(k\mid k)=\hat{\mathbf{y}}_i(k\mid k-1)+\sum_j \mathbf{H}_j(k)^T\mathbf{R}_j^{-1}(k)\mathbf{z}_j(k). \tag{3.75}$$

The predicted covariance is given by

$$\begin{aligned}\mathbf{P}_i(k\mid k-1) &= E\left\{(\mathbf{x}_i(k)-\hat{\mathbf{x}}_i(k\mid k-1))(\mathbf{x}_i(k)-\hat{\mathbf{x}}_i(k\mid k-1))^T\mid\{\mathbf{Z}^{k-1}\}\right\}\\ &= \mathbf{F}_i(k)\mathbf{P}_i(k-1\mid k-1)\mathbf{F}_i(k)^T+\mathbf{Q}_i(k),\end{aligned} \tag{3.76}$$

based on a similar derivation as in Equation 3.30. And so the predicted information state is given by

$$\hat{\mathbf{y}}_i(k\mid k-1)=\mathbf{P}_i^{-1}(k\mid k-1)\mathbf{F}_i(k)\mathbf{P}_i(k-1\mid k-1)\hat{\mathbf{y}}_i(k-1\mid k-1). \tag{3.77}$$

Partial Estimates

Partial estimates are estimates of the state based on a global prior (global information before current local observation) and current local information *only*. The importance of these partial estimates has already been alluded to, and shall become apparent in the next

chapter. The partial estimate $\tilde{\mathbf{x}}_i(k \mid k)$, is obtained by considering information update given by Equation 3.9, that is

$$\tilde{l}_i \triangleq ln\ p(\mathbf{x}_i(k) \mid \{\mathbf{Z}^{k-1} \cup \mathbf{z}_i(k)\}) = ln\ p(\mathbf{x}_i(k) \mid \{\mathbf{Z}^{k-1}\}) + ln\left[\alpha_i\ p(\mathbf{z}_i(k) \mid \mathbf{x}_i(k))\right]. \quad (3.78)$$

From this posterior an MMSE partial estimate is obtained as

$$\tilde{\mathbf{x}}_i(k \mid k) = E\left\{\mathbf{x}_i(k) \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k)\right\}, \quad (3.79)$$

with covariance

$$\tilde{\mathbf{P}}_i(k \mid k) = E\left\{(\mathbf{x}_i(k) - \tilde{\mathbf{x}}_i(k \mid k))(\mathbf{x}_i(k) - \tilde{\mathbf{x}}_i(k \mid k))^T \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k)\right\}. \quad (3.80)$$

Corresponding to this partial estimate, we can define a partial information state vector at time j given l, that is

$$\tilde{\mathbf{y}}_i(j \mid l) = \tilde{\mathbf{P}}_i(j \mid l)\tilde{\mathbf{x}}_i(j \mid l), \quad (3.81)$$

and a partial information matrix given by the inverse covariance $\tilde{\mathbf{P}}_i(j \mid l)$. These estimates can be obtained by following the steps outlined in Section 3.2.1, that is, evaluating (i) $\nabla_{\mathbf{x}_i}\tilde{l}_i$ and (ii) $\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^T\tilde{l}_i$. This yields the partial information state vector as

$$\tilde{\mathbf{y}}_i(k \mid k) = \hat{\mathbf{y}}_i(k \mid k-1) + \mathbf{H}_i(k)^T\mathbf{R}_i^{-1}(k)\mathbf{z}_i(k), \quad (3.82)$$

and the partial information matrix as

$$\tilde{\mathbf{P}}_i(k \mid k)^{-1} = \mathbf{P}_i^{-1}(k \mid k-1) + \mathbf{H}_i(k)^T\mathbf{R}_i^{-1}(k)\mathbf{H}_i(k), \quad (3.83)$$

where $\mathbf{P}_i^{-1}(k \mid k-1)$ and $\hat{\mathbf{y}}_i(k \mid k-1)$ are the global predictions given by Equations 3.76 and 3.77.

Decentralized non-linear information filter

By considering non-linear systems we can also derive a decentralized non-linear filter. The nodal non-linear observation equation is given by

$$\mathbf{z}_i(k) = \mathbf{h}_i[k, \mathbf{x}_i(k)] + \mathbf{v}_i(k), \tag{3.84}$$

where $\mathbf{h}_i[\cdot]$ is a non-linear nodal observation model and $\mathbf{v}_i(k)$ is additive Gaussian noise. And similarly for the nodal state transition equation

$$\mathbf{x}_i(k) = \mathbf{f}_i[k, \mathbf{x}_i(k-1)] + \mathbf{w}_i(k), \tag{3.85}$$

where $\mathbf{f}_i[\cdot]$ is a non-linear state transition model and $\mathbf{w}_i(k)$ is additive state transition noise.

By repeating the above derivation, that is, evaluating $\nabla_{\mathbf{x}_i} l_i$, $\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^T l_i$, $\nabla_{\mathbf{x}_i}\tilde{l}_i$ and $\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^T \tilde{l}_i$ using Equations 3.84 and 3.85 and considering only first order terms, the following filter is obtained. The information state is obtained as

$$\hat{\mathbf{y}}_i(k \mid k) = \hat{\mathbf{y}}_i(k \mid k-1) + \sum_j \nabla_{\mathbf{x}} \mathbf{h}_j[k, \hat{\mathbf{x}}_j(k \mid k-1)]^T \mathbf{R}_j^{-1}(k)\mathbf{z}_j(k), \tag{3.86}$$

and the inverse covariance

$$\mathbf{P}_i^{-1}(k \mid k) = \mathbf{P}_i^{-1}(k \mid k-1) + \sum_j \nabla_{\mathbf{x}} \mathbf{h}_j[k, \hat{\mathbf{x}}_j(k \mid k-1)]^T \mathbf{R}_j^{-1}(k) \nabla_{\mathbf{x}} \mathbf{h}_j[k, \hat{\mathbf{x}}_j(k \mid k-1)], \tag{3.87}$$

where the predicted covariance is obtained from

$$\begin{aligned} \mathbf{P}_i(k \mid k-1) &= \nabla_{\mathbf{x}} \mathbf{f}_i[k, \hat{\mathbf{x}}_i(k-1 \mid k-1)] \mathbf{P}_i(k-1 \mid k-1) \nabla_{\mathbf{x}} \mathbf{f}_i[k, \hat{\mathbf{x}}_i(k-1 \mid k-1)]^T \\ &\quad + \mathbf{Q}_i(k), \end{aligned} \tag{3.88}$$

and the predicted information state vector is obtained from

$$\hat{\mathbf{y}}_i(k \mid k-1) = \mathbf{P}_i^{-1}(k \mid k-1) \mathbf{f}_i[k, \hat{\mathbf{x}}_i(k-1 \mid k-1)]. \tag{3.89}$$

The partial information state vector is given by

$$\tilde{\mathbf{y}}_i(k \mid k) = \hat{\mathbf{y}}_i(k \mid k-1) + \nabla_{\mathbf{x}} \mathbf{h}_i[k, \hat{\mathbf{x}}_i(k \mid k-1)]^T \mathbf{R}_i^{-1}(k) \mathbf{z}_i(k), \tag{3.90}$$

and the partial inverse covariance by

$$\tilde{\mathbf{P}}_i(k \mid k) = \mathbf{P}_i^{-1}(k \mid k-1) + \nabla_{\mathbf{x}} \mathbf{h}_i[k, \hat{\mathbf{x}}_i(k \mid k-1)]^T \mathbf{R}_i^{-1}(k) \nabla_{\mathbf{x}} \mathbf{h}_i[k, \hat{\mathbf{x}}_i(k \mid k-1)]. \tag{3.91}$$

In order to evaluate $\nabla_{\mathbf{x}} \mathbf{h}_i[k, \hat{\mathbf{x}}_i(k \mid k-1)]$, $\hat{\mathbf{x}}_i(k \mid k-1)$ is obtained from

$$\hat{\mathbf{x}}_i(k \mid k-1) = \mathbf{P}_i(k \mid k-1) \hat{\mathbf{y}}_i(k \mid k-1). \tag{3.92}$$

By comparing the linear decentralized information filter and its non-linear counterpart, the following equivalences can be seen

$$\nabla_{\mathbf{x}} \mathbf{h}_i[k, \hat{\mathbf{x}}_i(k \mid k)] \equiv \mathbf{H}_i(k) \quad \text{and} \quad \nabla_{\mathbf{x}} \mathbf{f}_i[k, \hat{\mathbf{x}}_i(k-1 \mid k-1)] \equiv \mathbf{F}_i(k). \tag{3.93}$$

Substituting these makes the the non-linear decentralized information filter the same as the linear form (with the exception of Equation 3.89). Therefore we can summarize the decentralized information filter as follows:

Summary

Decentralized Information Filter Equations

At each node i, the global information state vector is obtained as

$$\hat{\mathbf{y}}_i(k \mid k) = \hat{\mathbf{y}}_i(k \mid k-1) + \sum_j \mathbf{H}_j(k)^T \mathbf{R}_j^{-1}(k) \mathbf{z}_j(k), \tag{3.94}$$

and the information matrix is given by

$$\mathbf{P}_i^{-1}(k \mid k) = \mathbf{P}_i^{-1}(k \mid k-1) + \sum_j \mathbf{H}_j(k)^T \mathbf{R}_j^{-1}(k) \mathbf{H}_j(k). \tag{3.95}$$

The partial information state vector is given by

$$\tilde{\mathbf{y}}_i(k \mid k) = \hat{\mathbf{y}}_i(k \mid k-1) + \mathbf{H}_i(k)^T \mathbf{R}_i^{-1}(k)\mathbf{z}_i(k), \tag{3.96}$$

and the partial information matrix by

$$\tilde{\mathbf{P}}_i(k \mid k)^{-1} = \mathbf{P}_i^{-1}(k \mid k-1) + \mathbf{H}_i(k)^T \mathbf{R}_i^{-1}(k)\mathbf{H}_i(k). \tag{3.97}$$

The predictions are obtained from

$$\mathbf{P}_i(k \mid k-1) = \mathbf{F}_i(k)\mathbf{P}_i(k-1 \mid k-1)\mathbf{F}_i(k)^T + \mathbf{Q}_i(k), \tag{3.98}$$

and

$$\hat{\mathbf{y}}_i(k \mid k-1) = \mathbf{P}_i^{-1}(k \mid k-1)\mathbf{F}_i(k)\mathbf{P}_i(k-1 \mid k-1)\hat{\mathbf{y}}_i(k-1 \mid k-1). \tag{3.99}$$

The global state estimate is obtained from

$$\hat{\mathbf{x}}_i(k \mid k) = \mathbf{P}_i(k \mid k)\hat{\mathbf{y}}_i(k \mid k), \tag{3.100}$$

and the partial state estimate from

$$\tilde{\mathbf{x}}_i(k \mid k) = \tilde{\mathbf{P}}_i(k \mid k)\tilde{\mathbf{y}}_i(k \mid k). \tag{3.101}$$

An information filter corresponding to the architecture described by the Independent Opinion Pool (Equation 3.6 and Equation 3.7) can also be developed in a similar manner. The resulting filter is essentially a decentralization of the equivalent hierarchy for the Independent Opinion Pool given in Section 3.1.2. As before, the usefulness of such an information filter is limited to applications where the prior information at each sensor node is subjective.

3.2.4 Discussion

We have shown that given any architecture in which information update is a component of the data fusion process, where knowledge of the state transition model and the observation model is made available, we can develop from first principles (through the Information Update) an appropriate information filter to estimate the state. This is with the *proviso* that the PDFs in the information update are Gaussian.

The initialization of the information filter is a practical issue which can be approached in several ways. From considerations of maximum entropy, a least informative prior can be derived giving an initial covariance (see Appendix A.2). However from our discussion in Chapter 2, reasonable approximations to non-informative priors can also be used. A particular advantage of the information form is that at initialization, the elements of the inverse covariance (information matrix) can be set to be *zero* implying infinite variance or highest uncertainty. This agrees with our notion of the inverse covariance representing information which at initialization we set to zero.

It has already been shown by Durrant-Whyte *et al* [49] and by Willsky [136] that decentralizing or distributing the Kalman filter does not compromise optimality if the sensors are fully connected, and since the information filter is equivalent to a standard Kalman filter, the arguments for optimality also apply to the algorithms we have presented. Recent work by Grime and Durrant-Whyte [63] and Grime [62] has shown how decentralized Kalman filters can be developed for non-fully connected networks of sensor nodes. Non-linear filters however, compromise MMSE optimality as follows: As we have already seen, the conditional mean achieves MMSE optimality (see Chapter 2), however the estimate from the non-linear

filter is only an approximation of the true conditional mean, that is

$$\hat{\mathbf{x}}(k \mid k) \approx E\left\{\mathbf{x}(k) \mid \mathbf{Z}^k\right\}.$$

And so the corresponding $\mathbf{P}(k \mid k)$ is thus not an exact covariance but an approximate MSE. This arises from the use of the Taylor series expansion around the estimate for which only a finite number of terms are considered in practice.

3.3 Discrete Classification from Information Update

Discrete classification algorithms are a useful counterpart to continuous state estimation algorithms because they allow the attributes of tracked objects to be inferred. This is provided that the observed information contains attribute information. We now present such an algorithm.

3.3.1 Bayesian Classification Algorithm

In a classification or identification problem, the state vector $\mathbf{x}$ takes on discrete values, $\mathbf{x} = [X_1, X_2, \cdots, X_n]$, where each X_b signifies an attribute or distinct object type. Each attribute or object X_b is characterized by a set of observable parameters and so our observation model consists of a parameter set $\{M\} = \{M_1, M_2, \cdots, M_m\}$, which relates each observable $z_b \in \mathbf{z}$ to elements of $\mathbf{x}$. Using the observation model defined in Equation 2.1, the observations can be written

$$z_b = \{M\}(X_b, v_b), \tag{3.102}$$

where v_b is the observation noise associated with each observation z_b. We assume each observation z_b, of a parameter in $\{M\}$ is independent of all the other parameters in $\{M\}$. The likelihood for each distinct class $X_b \in \mathbf{x}$ is given by

$$\lambda_{\mathbf{z}(k)}(X_b) \triangleq p(\mathbf{z}(k) \mid X_b). \tag{3.103}$$

With the parameter model $\{M\}$, the likelihood vector for $\mathbf{x}$ is given by

$$\begin{aligned}\Lambda_{\mathbf{z}(k)}(\mathbf{x}) &= \left[\lambda_{\mathbf{z}(k)}(X_1),\cdots,\lambda_{\mathbf{z}(k)}(X_n)\right] = [p(\mathbf{z}(k)\mid X_1),\cdots,p(\mathbf{z}(k)\mid X_n)] \\ &= [p(\mathbf{z}(k)\mid M_1),\cdots,p(\mathbf{z}(k)\mid M_m)]\begin{bmatrix} p(M_1\mid X_1), & \cdots, p(M_1\mid X_n) \\ \vdots & \vdots \\ p(M_m\mid X_1), & \cdots, p(M_m\mid X_n)\end{bmatrix},\end{aligned}$$

where $p(\mathbf{z}(k)\mid\{M\})$ relates observations to the parameter model, that is, each $p(\mathbf{z}(k)\mid M_l)$ gives the probability of the observation $\mathbf{z}(k)$ given that the true parameter to be observed is M_l. The matrix $p(\{M\}\mid\mathbf{x})$ relates the parameter model to the vector $\mathbf{x}$, that is, each $p(M_l\mid X_b)$ gives the probability that given the classification is X_b, the observed parameter is M_l. We can then write Equation 2.5 for each X_b as follows

$$\forall X_b\in\mathbf{x},\quad p(X_b\mid\mathbf{Z}^k) = p(X_b\mid\mathbf{Z}^{k-1})\left[\alpha\, p(\mathbf{z}(k)\mid\{M\})\, p(\{M\}\mid X_b)\right], \tag{3.104}$$

where as before, α is a normalizing constant. From this, the inferred classification is given by the *maximum a posteriori* (MAP) estimate

$$\hat{X}_b = \arg\max_b\left[p(\mathbf{x}\mid\mathbf{Z}^k)\right] = \arg\max_b\left[p(X_1\mid\mathbf{Z}^k),\cdots,p(X_n\mid\mathbf{Z}^k)\right]. \tag{3.105}$$

From the discussion in Chapter 2, with a non-informative prior the MAP estimate is the same as the ML estimate. In general, for this algorithm an MMSE is not well defined since this is only appropriate for random parameters. However, if the posterior PDF is symmetric and the mean and mode coincide, the above MAP estimate becomes equivalent to an MMSE estimate. The algorithm of Equation 3.104 can be written in the now familiar log-likelihood form of the probabilistic information update for each $X_b\in\mathbf{x}$ as follows

$$\begin{aligned}\forall X_b\in\mathbf{x},\ ln\, p(X_b\mid\mathbf{Z}^k) &= ln\, p(X_b\mid\mathbf{Z}^{k-1}) + ln\left[\frac{p(\mathbf{z}(k)\mid\{M\})\, p(\{M\}\mid X_b)}{p(\mathbf{z}(k))}\right] \\ &= ln\, p(X_b\mid\mathbf{Z}^{k-1}) + ln\left[\alpha\lambda_{\mathbf{z}(k)}(X_b)\right].\end{aligned} \tag{3.106}$$

This classification algorithm corresponds to the single sensor architecture of Figure 3.1.

3.3.2 Hierarchical Classification Algorithm

We now consider the classification algorithm for centralized and hierarchical architectures:

1. In the hierarchy based on the Independent Opinion Pool and illustrated in Figure 3.2 where each sensor node communicates a complete local estimate, the fusion equation at the central processor is given by the summation

$$ln\ p(X_b \mid \{\mathbf{Z}^k\}) = \sum_j \underbrace{ln\ p(X_b \mid \mathbf{Z}_j^k)}_{\text{communicated}},\ \forall X_b \in \mathbf{x}, \tag{3.107}$$

where for each node i, the communicated local posterior $ln\ p(X_b \mid \mathbf{Z}_i^k)$, is given by

$$\begin{aligned} ln\ p(X_b \mid \mathbf{Z}_i^k) &= ln\ p(X_a \mid \mathbf{Z}_i^{k-1}) + ln\ \left[\frac{p(\mathbf{z}_j(k) \mid \{M_i\})\, p(\{M_i\} \mid X_a)}{p(\mathbf{z}_i(k))}\right] \\ &= ln\ p(X_a \mid \mathbf{Z}_i^{k-1}) + ln\ \left[\alpha_i \lambda_{\mathbf{z}_i(k)}(X_a)\right],\ \forall X_b \in \mathbf{x}. \end{aligned} \tag{3.108}$$

The central processor obtains a global classification simply by summing the local classification results from each sensor. As before, the usefulness of this is restricted to cases when local subjective information is available at each sensor.

2. In the hierarchy based on the Independent Likelihood Pool and illustrated in Figure 3.3, the fusion equation at the central processor is given by

$$\begin{aligned} ln\ p(X_a \mid \{\mathbf{Z}^k\}) &= ln\ p(X_a \mid \{\mathbf{Z}^{k-1}\}) + \sum_j ln\ \left[\frac{p(\mathbf{z}_j(k) \mid \{M_j\})\, p(\{M_j\} \mid X_{jb})}{p(\mathbf{z}_j(k))}\right] \\ &= ln\ p(X_a \mid \{\mathbf{Z}^{k-1}\}) + \sum_j \underbrace{ln\ \left[\alpha_j \lambda_{\mathbf{z}_j(k)}(X_{jb})\right]}_{\text{communicated}},\ \forall X_b \in \mathbf{x}. \end{aligned} \tag{3.109}$$

In this case, the central processor updates the global classification based on the global prior and the likelihood information from the sensor observations.

3.3.3 Decentralized Classification Algorithm

An equivalent decentralized algorithm can be derived in the same way as was done in Section 3.3.1. Using the definition of a local state in Equation 3.5, the state at each sensor i is denoted $\mathbf{x}_i = [X_{i1}, X_{i2}, \ldots, X_{in}]$. And so, at sensor node i we write the posterior as $p(\mathbf{x}_i \mid \{\mathbf{Z}^k\})$ and correspondingly for the other PDFs. The posterior $p(\mathbf{x}_i \mid \{\mathbf{Z}^k\})$ is comprised of probabilities $p(X_{ib} \mid \{\mathbf{Z}^k\})$, corresponding to each object type $X_{ib} \in \mathbf{x}_i$. Each sensor node i has its own parameter model $\{M_i\}$ corresponding to its sensing device. The nodal observations are described by

$$z_{i_b} = \{M_i\}(X_{ib}, v_{i_b}). \tag{3.110}$$

We assume that each sensor node's observation z_{i_b}, of a parameter in $\{M_i\}$ is independent of the particular values of all the other parameters in $\{M_i\}$. The partial likelihood vector at sensor node i is given by

$$\Lambda_{\mathbf{z}_i(k)}(\mathbf{x}_i) = p(\mathbf{z}_i(k) \mid \mathbf{x}_i). \tag{3.111}$$

The partial likelihood for each sensor node i can therefore be written as

$$\Lambda_{\mathbf{z}_i(k)}(\mathbf{x}_i) = p(\mathbf{z}_i(k) \mid \{M_i\})\ p(\{M_i\} \mid \mathbf{x}_i). \tag{3.112}$$

Each sensor node communicates its partial likelihood vectors to all connected nodes and in return receives their partial likelihoods vectors. Each sensor node is then able to fuse this likelihood information with a global prior to obtain a posterior, which is converted to a set of true probabilities by normalizing over the entire state vector $\mathbf{x}$. The fusion equation is given by

$$\forall X_{ib} \in \mathbf{x}_i, \quad p(X_{ib} \mid \{\mathbf{Z}^k\}) = p(X_{ib} \mid \{\mathbf{Z}^{k-1}\}) \prod_j [\alpha_j p(\mathbf{z}_j(k) \mid \{M_j\})\ p(\{M_j\} \mid X_{jb})]. \tag{3.113}$$

The most probable state is inferred using the MAP estimate which is given by

$$\hat{X}_i = \arg\max_b \left[p(\mathbf{x}_i \mid \{\mathbf{Z}^k\}) \right]. \tag{3.114}$$

The posterior information can be written in log-likelihood terms for each X_{ib} as follows

$$\begin{aligned} ln\ p(X_{ib} \mid \{\mathbf{Z}^k\}) &= ln\ p(X_{ib} \mid \{\mathbf{Z}^{k-1}\}) + \sum_j ln \left[\frac{p(\mathbf{z}_j(k) \mid \{M_j\})\ p(\{M_j\} \mid X_{jb})}{p(\mathbf{z}_j(k))} \right] \\ &= ln\ p(X_{ib} \mid \{\mathbf{Z}^{k-1}\}) + \sum_j ln \left[\alpha_j \lambda_{\mathbf{z}_j(k)}(X_{jb}) \right], \quad \forall X_{ib} \in \mathbf{x}_i. \end{aligned} \tag{3.115}$$

As we have shown in Equation 3.9, each sensor node can also compute partial information. From Equation 3.113 the partial posterior at each sensor node i is given by

$$\forall X_{ib} \in \mathbf{x}_i, \quad p(X_{ib} \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k)) = p(X_{ib} \mid \{\mathbf{Z}^{k-1}\}) \left[\alpha_i p(\mathbf{z}_i(k) \mid \{M_i\})\ p(\{M_i\} \mid X_{ib}) \right]. \tag{3.116}$$

The partial MAP estimate at sensor node i denoted $\tilde{X}_i$, is given by

$$\tilde{X}_i = \arg\max_b \left[p(\mathbf{x}_i \mid \{\mathbf{Z}^k\} \cup \mathbf{z}_i(k)) \right]. \tag{3.117}$$

In terms of log-likelihoods, the partial information at each sensor node i is given by

$$\begin{aligned} \forall X_{ib} \in \mathbf{x}_i, \quad ln\ p(X_{ib} \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k)) &= ln\ p(X_{ib} \mid \{\mathbf{Z}^{k-1}\}) + \\ & \quad ln \left[\alpha_i p(\mathbf{z}_i(k) \mid \{M_i\})\ p(\{M_i\} \mid X_{ib}) \right]. \end{aligned} \tag{3.118}$$

We have only presented the algorithm corresponding to a decentralized architecture based on the Independent Likelihood Pool. The corresponding algorithm for the Independent Opinion Pool follows from the hierarchal algorithm defined by Equations 3.107-Equation 3.108.

3.3.4 Discussion

We have assumed that we are classifying objects whose identity does not change with time. This could be made general to accommodate objects with time-variant state classification $\mathbf{x}(k)$, as long as the changes are within a known bounded set of alternatives $\mathbf{x}$. An additional requirement would be a transition model describing the way the true classification changes with time k, within the set of alternatives $\mathbf{x}$. Another implicit assumption in the classification algorithm, is that we are able to observe the target every processing step. This is because the algorithm on its own does not generate any information about the spatial positioning of the target. Thus this algorithm can conveniently be implemented in conjunction with the information filters of the target.

The definition of the state as $\mathbf{x}_i$ at sensor node i allows us to extend the algorithms to allow for different nodal frames of discernment. In practical terms this arises when sensor nodes have different sensor devices and are able to distinguish between different sets of objects. With the algorithms as presented here where $\mathbf{x}$ is the same at every sensor node, this can still be accounted for by assigning equi-probabilities to objects which are indistinguishable at a particular sensor node. However, this can become cumbersome in some applications if a particular sensor node can distinguish only a small subset of $\mathbf{x}$. Therefore by defining a global $\mathbf{x} = \cup_i\{\mathbf{x}_i\}$ where each $\mathbf{x}_i$ is restricted to only targets distinguishable at i, computations of the local partial likelihood at i are reduced to computations of only likelihoods of targets which can be distinguished. This makes the algorithm more modular as each sensor node (at the likelihood computation stage) needs only to know about targets that it can distinguish in its parameter model. This is similar to the use of reduced order states in distributed continuous estimation [4].

Chapter 4

Normative Data Fusion Management

This chapter addresses the issue of *how* to make decisions for sensor management. The decision-theoretic methods presented in Chapter 2 form the basis of a *normative* approach to making sensor management decisions. A normative approach is based on an axiomatic description of the decision making process making use of appropriate information or data relevant to the decision at hand. Such an approach can be described as being a combination of probabilistic reasoning, which establishes objectivity, and utility theory which incorporates subjectivity into the solution.

As with strict Bayesian analysis, the normative approach has traditionally been eschewed on the grounds that modelling the requisite *a priori* information on which the decisions are based, is difficult. This is because a normative approach requires that all information used should have a "proper" probabilistic interpretation. The situation is further exacerbated due to the fact that the solution is often hard to compute and difficult to implement practically. Notwithstanding these difficulties, we argue that a normative approach provides a common and consistent framework for formulating the management problem in decentralized systems, provided that an appropriate utility structure can be constructed that reflects the goals and preferences of the system.

We start by considering the elements making up a normative formulation of the sensor management problem. We then propose an information based utility and proceed to develop

a decentralized solution and discuss practical issues relating to implementation.

4.1 Elements of a Normative Formulation

4.1.1 Probabilistic Information

The work in Chapter 3 models the probabilistic components of the information update relationship of Equation 2.50 for various architectures implementing the two data fusion algorithms. For the information filter this is summarised by the following for the prior, likelihood and posterior:

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{Z}^{k-1}): &\quad \text{modelled by} \quad \underbrace{\hat{\mathbf{y}}(k \mid k-1)}_{\text{mean}} \text{ and } \underbrace{\mathbf{P}^{-1}(k \mid k-1)}_{\text{covariance}} \\ p(\mathbf{z}(k) \mid \mathbf{x}): &\quad \text{modelled by} \quad \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{z}(k) \text{ and } \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k) \\ p(\mathbf{x} \mid \mathbf{Z}^{k}): &\quad \text{modelled by} \quad \hat{\mathbf{y}}(k \mid k) \text{ and } \mathbf{P}^{-1}(k \mid k). \end{aligned} \tag{4.1}$$

In the classification algorithm the equivalent information is based on

$$\begin{aligned} \text{prior}: &\quad ln\left[p(X_a \mid \mathbf{Z}^{k-1})\right] \\ \text{likelihood}: &\quad ln\left[\alpha\lambda_{\mathbf{z}(k)}(X_a)\right] \\ \text{posterior}: &\quad ln\left[p(X_a \mid \mathbf{Z}^{k})\right], \quad \forall X_b \in \mathbf{x}. \end{aligned} \tag{4.2}$$

These models each have their decentralised equivalents as shown in Chapter 3.

4.1.2 Management Imperative, Actions and Outcomes

The management "imperative" can be defined as the fundamental motivating purpose of the sensing system, on which any sensing activities or choices can be based and is an abstraction of more specific sensor functions and task requirements. Sensor management has as a proximate goal the optimal management of sensor resources and capabilities for purposes of gaining information. In a given sensor system this may be refined to include

more specific requirements such as observation and track maintenance, and the effective use of limited resources. As a specific example, in [114], Popoli defines the management imperative in terms of sensor functional roles and tactical benefits for a combat aircraft application.

We define an action set which is a set of *distinct* alternatives for achieving the management imperative as follows

$$\mathcal{A} = \{a_1, a_2, \ldots, a_m\}. \tag{4.3}$$

The actions correspond to the different sensing configurations possible. Due to uncertainty, each action a_l produces a probabilistic outcome p_l defined on $\mathcal{P}$. Mapping an action to an outcome is sometimes termed *action-outcome association.* As a generalization of the probabilistic outcome, we define the outcome ρ_l which is a *set* of PDFs (over $\mathbf{x}$) given the action a_l such that

$$a_l \mapsto \rho_l, \quad \text{where} \quad \rho_l \triangleq \{p_1, p_2, \ldots p_q\}_l = \{p_1, p_2, \cdots, p_q \mid a_l\}. \tag{4.4}$$

Associating an action a_l with a set of distributions ρ_l over the state $\mathbf{x}$, makes intuitive sense from the point of view that the sensing strategy adopted leads to the probability distributions of $\mathbf{x}$ obtained. These distributions are then subsequently used to make inferences concerning the state $\mathbf{x}$. And so the action taken has a direct bearing on the inferred state. In the inference problems that we considered in Chapter 3, we can say that a probabilistic outcome ρ_l maps through inference (estimation) to an estimate $\hat{\mathbf{x}}$ of the state of nature as shown in Figure 4.1. The modalities of this mapping were discussed in Chapter 2, where sets of probabilities are aggregated to give a global distribution from which we can make inferences.

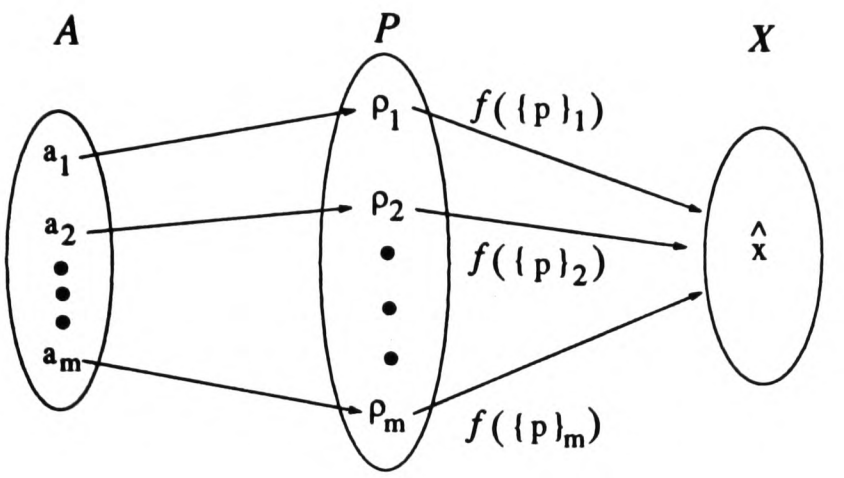


Figure 4.1: Mapping from action space through probabilistic results set to inferred state $\hat{\mathbf{x}}$.

4.1.3 Sensor Preferences and Decision Structure

In order to encode the preferential structure of a sensor faced with several choices of sensing actions, we make use of Utility theory as introduced in Chapter 2. A sensing system will be required to state a preference from among the outcomes thus forming an ordered set

$$\langle \rho_1, \rho_2, \ldots, \rho_m \rangle, \quad \text{such that} \quad \rho_1 \succeq \rho_2 \succeq \cdots \succeq \rho_m, \tag{4.5}$$

where ρ_1 is the most preferred outcome. Without such a preferential structure, the management problem in terms of decision theory is ill-defined as there is no basis for evaluation of actions. This preference structure is guaranteed by the rationality axioms of Section 2.3.3. A utility function must therefore be constructed which reflects this preference ordering through its expected value.

Decisions are then made by a maximization of the expected utilities for the various actions in $\mathcal{A}$ as shown in Chapter 2. We now write this in terms of ρ_l, that is,

$$\hat{a} \;=\; \arg\max_a \beta(\rho_l) = \arg\max_a E^{\rho_l}\{U(x, a_l)\}. \tag{4.6}$$

This is the sensor management structure which is illustrated in Figure 4.2. Assuming that the sensor has adequate computing and communication resources, and all the information

concerning **x** is made available, the whole management structure can be implemented locally at each sensor.

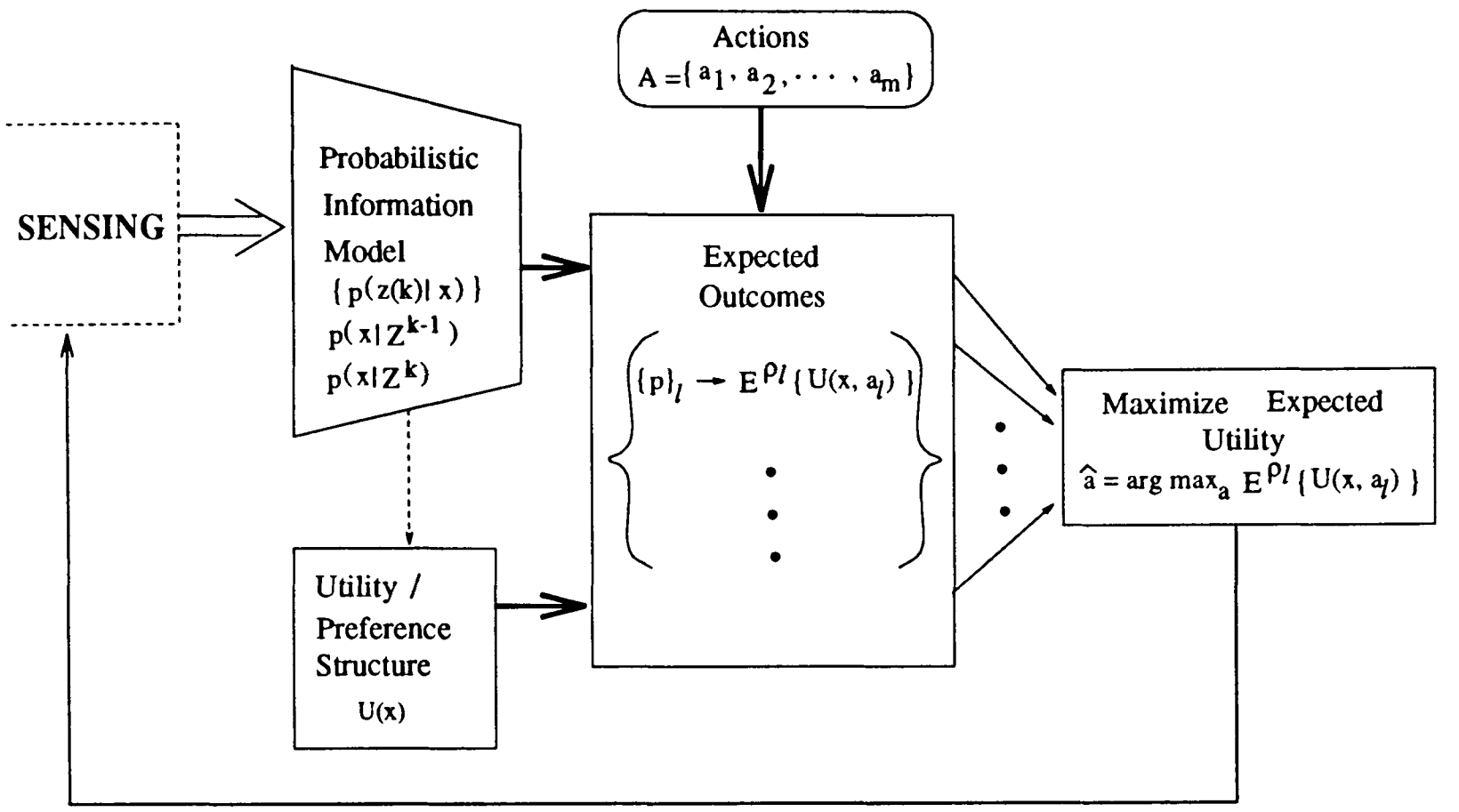


Figure 4.2: Elements of a normative approach to sensor management.

4.2 The Utility of Information

The purpose of the observation process described in Chapter 2 is the gain of information and the reduction of uncertainty. Therefore, given a choice of actions and corresponding outcomes, a rational sensor (in this sense) prefers outcomes which result in a larger gain of information. It would thus seem logical to develop a preferential utility structure based on information and its gain.

4.2.1 Information as Expected Utility

We propose the use of information as the expected utility in making management decisions and taking actions. For a *single* outcome PDF p_l corresponding to the action a_l, we can write that

$$E^{p_l}\{U(\mathbf{x}, a_l)\} \propto \mathcal{I}(p_l), \tag{4.7}$$

where $\mathcal{I}(\cdot)$ is a function which quantifies the information in its argument. In general, the outcome corresponding to a_l is the *set* of distributions $\rho_l = \{p_1, p_2, \ldots, p_q\}_l$. In this case the total expected utility of action a_l is given by

$$\begin{aligned} E^{\rho_l}\{U(\mathbf{x}, a_l)\} &= E^{p_1}\{U(\mathbf{x}, a_l)\} + E^{p_2}\{U(\mathbf{x}, a_l)\} + \cdots + E^{p_q}\{U(\mathbf{x}, a_l)\} \\ &\propto [\mathcal{I}(p_1) + \mathcal{I}(p_2) + \cdots + \mathcal{I}(p_q)] \\ &\propto \mathcal{I}(\rho_l) \\ &= \kappa\mathcal{I}(\rho_l), \end{aligned} \tag{4.8}$$

where $\mathcal{I}(\rho_l)$ is read as the information contained in the outcome set ρ_l. The summation of expected utilities in this way is justifiable given the axioms of utility and the properties of the expectation operator. To justify the proposition in Equation 4.7, the function $\mathcal{I}(\cdot)$ must have the following properties:

1. **Preference structure.** From utility theory, given two actions a_1 and a_2,

$$E^{\rho_1}\{U(\mathbf{x}, a_1)\} \leq E^{\rho_2}\{U(\mathbf{x}, a_2)\} \Rightarrow \rho_1 \preceq \rho_2. \tag{4.9}$$

   We require that given two such actions a_1 and a_2, the function $\mathcal{I}(\cdot)$ maintains this preference structure that is

$$\rho_1 \preceq \rho_2 \Rightarrow \mathcal{I}(\rho_1) \leq \mathcal{I}(\rho_2), \tag{4.10}$$

and in terms of the actual resulting PDFs

$$\rho_1 \preceq \rho_2 \Rightarrow \mathcal{I}(\{p\}_1) \leq \mathcal{I}(\{p\}_2). \tag{4.11}$$

2. **Transitivity.** Given actions a_1, a_2 and a_3, we expect the following utility inequalities

$$E^{\rho_1}\{U(\mathbf{x}, a_1)\} \leq E^{\rho_2}\{U(\mathbf{x}, a_2)\} \quad \text{and} \quad E^{\rho_2}\{U(\mathbf{x}, a_2)\} \leq E^{\rho_3}\{U(\mathbf{x}, a_3)\},$$

$$\Rightarrow \rho_1 \preceq \rho_2 \preceq \rho_3. \tag{4.12}$$

Therefore the function $\mathcal{I}(\cdot)$ should give the same ordering that is

$$\rho_1 \preceq \rho_2 \preceq \rho_3 \Rightarrow \mathcal{I}(\{p\}_1) \leq \mathcal{I}(\{p\}_2) \leq \mathcal{I}(\{p\}_3). \tag{4.13}$$

3. **Conditionality.** Given actions a_1, a_2 and a_3 such that in the corresponding result space $\rho_1 \preceq \rho_2$ and $\rho_2 \preceq \rho_3$, conditionality requires that

$$\alpha\, E^{\rho_1}\{U(\mathbf{x}, a_1)\} + (1-\alpha)\, E^{\rho_3}\{U(\mathbf{x}, a_3)\} \leq \alpha\, E^{\rho_2}\{U(\mathbf{x}, a_2)\} + (1-\alpha)\, E^{\rho_3}\{U(\mathbf{x}, a_3)\}, \tag{4.14}$$

and so we require that

$$\alpha\, \mathcal{I}(\{p\}_1) + (1-\alpha)\, \mathcal{I}(\{p\}_3) \leq \alpha\, \mathcal{I}(\{p\}_2) + (1-\alpha)\, \mathcal{I}(\{p\}_3), \tag{4.15}$$

where $0 \leq \alpha \leq 1$.

Given these axiomatic requirements on the function $\mathcal{I}(\cdot)$, the proportionality constant in Equation 4.7 is evidently of no significance and so can be assigned $\kappa = 1$ and ignored. In the context of the requirements, $\mathcal{I}(\cdot)$ must be a mapping from the probability space $\mathcal{P}$ to the real line

$$\mathcal{I}(\cdot) : \mathcal{P} \mapsto \Re, \tag{4.16}$$

which amounts to an information valuation metric. What remains is to find such an information metric satisfying the axiomatic requirements.

Motivated by the fact that "..the value of a probability distribution .. would be given by the expected utility.." [18] and the role of entropy as we described it in Chapter 2, we propose using entropy to realize the function $\mathcal{I}(\cdot)$. Thus from Equation 4.8

$$\begin{aligned} E^{\rho_l}\{U(\mathbf{x}, a_l)\} &= \mathcal{I}(\rho_l) = [\mathcal{I}(p_1) + \mathcal{I}(p_2) \cdots + \mathcal{I}(p_q)] \\ &\triangleq [-h(p_1) - h(p_2) - \cdots - h(p_q)] \\ &= -h(\rho_l). \end{aligned} \tag{4.17}$$

The summation of entropies in this way is justified as shown in [112]. By its definition, entropy satisfies the mapping of Equation 4.16 where, for an n-dimensional discrete state vector , the space $\mathcal{P}$ becomes the standard simplex $n-1$ in $\Re^n$ [29]. And for an n-dimensional continuous state vector, $\mathcal{P}$ is the support of p such that $supp(p) = \{x \in \Re^n \mid p(x) \neq 0\}$. With this premise, entropy satisfies all the above requirements. The use of such a metric is illustrated in the following example:

Example 1. At any one moment, a sensor can observe 3 targets which provide information concerning the same state $\mathbf{x}$. The management decision is defined by the action set $\mathcal{A} = \{a_1, a_2\}$, where a_1 implies observation of target set $\{t\}_1$ and a_2 implies observation of target set $\{t\}_2$. The outcomes are $\rho_1 = \{p_1, p_2, p_3 \mid a_1\}$ and $\rho_2 = \{p_1, p_2, p_3 \mid a_2\}$. These PDFs are as illustrated in Figure 4.3. The action resulting in higher information content as quantified by $\mathcal{I}(\rho_l)$ is chosen where

$$\mathcal{I}(\rho_l) = \mathcal{I}(p_1, a_l) + \mathcal{I}(p_2, a_l) + \mathcal{I}(p_3, a_l).$$

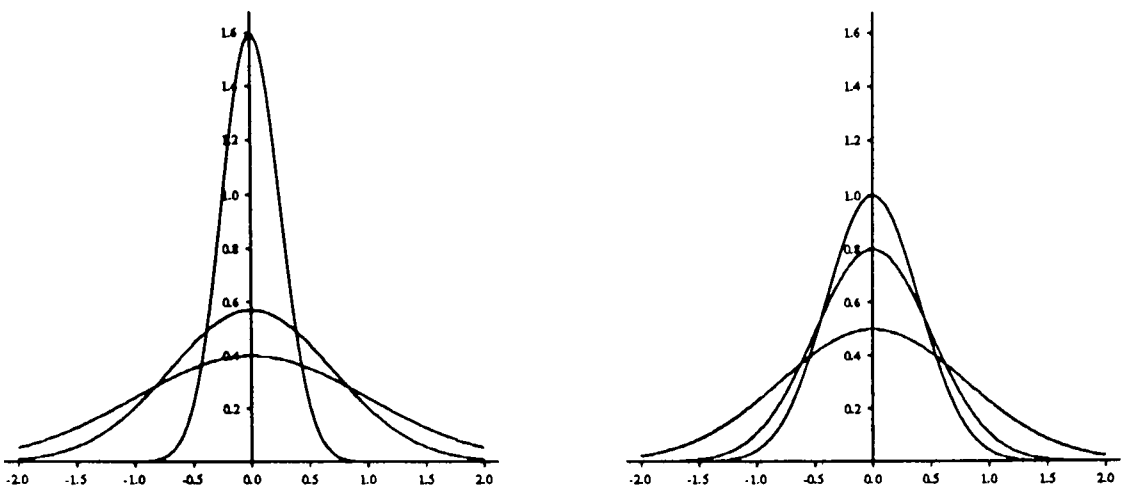


Figure 4.3: Probability distributions arising from actions a_1 and a_2 in **Example 1.**

4.2.2 Resulting Preference Profiles

As discussed in Chapter 2, it is necessary to characterise the behaviour of the decision-maker given the proposed information-based expected utility. This can be analysed from a consideration of Equation 4.17 as follows; for an outcome which is a single PDF p_l we have that

$$E^{p_l}\{U(\mathbf{x}, a_l)\} = E^{p_l}\{\ ln\ p_l\}, \tag{4.18}$$

from which it follows that the utility function is given by

$$U(\mathbf{x}, a_l) = \ ln\ p_l. \tag{4.19}$$

From the definition of strict convexity [39], we can show that the function $U(\mathbf{x}, a_l)$ defined as above is concave over the distribution. This can be easily verified by showing that the second derivative is *not* non-negative everywhere[1]. Ignoring the a_l's and considering a single outcome distribution p, we can write for a concave function

$$\begin{aligned} ln\ p(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) &\geq \alpha ln\ p(\mathbf{x}_1) + (1-\alpha) ln\ p(\mathbf{x}_2) \\ U[\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2] &\geq \alpha U(\mathbf{x}_1) + (1-\alpha)U(\mathbf{x}_2), \end{aligned} \tag{4.20}$$

[1] A plot of $ln\ \rho$ wrt ρ demonstrates this graphically.

for $0 \leq \alpha \leq 1$, which can be validated by the log-inequality. This implies that

$$U(E\{\mathbf{x}\}, a_l) \geq E\{U(\mathbf{x}, a_l)\}. \tag{4.21}$$

For the outcome $\rho_l = \{p\}_l$, we have that (using Equation 4.8)

$$U(\mathbf{x}, a_l) = [ln\ p_1 + ln\ p_2 + \cdots + ln\ p_q]. \tag{4.22}$$

This utility exhibits the same behaviour as for the single PDF outcome considered above. A utility function satisfying this concave inequality results in risk-averse behaviour over the eventual inferred states $\mathbf{x}$. This assertion is based on a proof by Ferguson [54] which demonstrates that a scalar functional can be used to order outcomes according to the decision-maker's preferences, where the precise nature of the preference structure depends on the convexity of the the scalar functional. Risk-averse behaviour takes into account the probabilities associated with acquiring information, whereas risk-prone behaviour only considers the value of the information which could be gained irrespective of the probability of acquiring it. Hence, our chosen utility function results in risk-aversion which is desirable in a system where robustness and reliability are requisite characteristics.

Alternatively, the information in Equation 4.8 may be quantified using Fisher information. From the definition of Fisher information (Equation 2.47), we can write Equation 4.7, for the case where ρ_l is the single distribution $p_l = p(\mathbf{Z}^k, \mathbf{x})$, as

$$E^{p_l}\{U(\mathbf{x}, a_l)\} = E\{-\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}{}^T\ ln\ p_l(\mathbf{Z}^k, \mathbf{x})\}. \tag{4.23}$$

To demonstrate the decision structure which results, consider the following for scalar variables

$$\begin{aligned} E^p\{U(x, a_l)\} &= E\left\{\frac{\partial^2}{\partial x^2} ln\ p(z, x)\right\} \\ \text{Thus}\quad U(x, a_l) &= -\frac{\partial^2}{\partial x^2} ln\ p(z, x), \end{aligned} \tag{4.24}$$

which is a convex function of the distribution resulting in risk-prone behaviour.

4.3 Algorithms and Information Metrics

4.3.1 Information Filter Metrics

The logical measure of information in the information filter is Fisher information $\mathbf{J}$, which is simply given by the inverse covariance. In a Fisher sense, information in the posterior $p(\mathbf{x} \mid \mathbf{Z}^k)$, is given by $\mathbf{P}^{-1}(k \mid k)$ and that in the prior $p(\mathbf{x} \mid \mathbf{Z}^{k-1})$, by $\mathbf{P}^{-1}(k \mid k-1)$. The information due to the observation is given by $\mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k)$. (See Chapter 3.). The discussion in Section 4.1 highlights the need for a metric which is a scalar functional of its argument and which realizes the mapping in Equation 4.16. Several techniques are available for obtaining a scalar functional from the Fisher information matrices [78][2]. However, from its definition in Chapter 2, Fisher information is not generally applicable, being limited to cases when the distribution is continually differentiable everywhere. Given this limitation and the risk-prone decision profiles resulting from using Fisher information as expected utility, we advocate the use of information defined using entropy as the expected utility.

In the ensuing discussion we quantify the information in the components of the information update for the information filter and its decentralized version using entropy.

Considering a Gaussian PDF and using the result in Appendix A.1, we obtain the following:

1. **Posterior Entropy and Information.** For an n-dimensional state vector $\mathbf{x}(k)$, the entropy is given by

$$h(k) \triangleq h(p(\mathbf{x}(k) \mid \mathbf{Z}^k)) = \frac{1}{2} E\left\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mathbf{P}^{-1}(k \mid k)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) + ln\ [(2\pi)^n \mid \mathbf{P}(k \mid k) \mid]\right\}$$

[2]The determinant $|\ |$, or a norm such as the Frobenius norm $\|\ \|$ could be used. Work described by Jeffreys [78] on non-informative priors makes use of the determinant to quantify Fisher information.

$$= \frac{1}{2} ln\ [(2\pi e)^n \mid \mathbf{P}(k \mid k) \mid]. \tag{4.25}$$

From this, we define an information metric $\mathbf{I}(k)$ which quantifies all information concerning the state $\mathbf{x}$ available up to and including time step k as follows

$$\mathbf{I}(k) \triangleq \mathcal{I}(p(\mathbf{x}(k) \mid \mathbf{Z}^k)) = -\frac{1}{2} ln\,[(2\pi e)^n \mid \mathbf{P}(k \mid k) \mid]. \tag{4.26}$$

It must be noted that if a non-informative prior is used at initialization, then $\mathbf{I}(k)$ quantifies all observation information up to and including that at time step k.

2. **Prior Entropy and Information.** In a similar manner, we can show that

$$h(k-1) = \frac{1}{2} ln\ [(2\pi e)^n \mid \mathbf{P}(k \mid k-1) \mid]. \tag{4.27}$$

Corresponding to this, we can define the metric $\mathbf{I}(k-1)$, for information at k given only information up to the $(k-1)$th time-step as

$$\mathbf{I}(k-1) \triangleq \mathcal{I}(p(\mathbf{x}(k) \mid \mathbf{Z}^{k-1})) = -\frac{1}{2} ln[(2\pi e)^n \mathbf{P}(k \mid k-1)]. \tag{4.28}$$

3. **Likelihood information.** From a consideration of the information contained in the observation, we can show that

$$\begin{aligned} i\left(\frac{p(\mathbf{z}(k) \mid \mathbf{x}(k))}{p(\mathbf{z}(k) \mid \mathbf{Z}^{k-1})}\right) &= -\frac{1}{2} E\left\{(\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k))^T \mathbf{R}^{-1}(k) (\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}(k))\right. \\ &\quad \left. + ln\ [(2\pi)^n \mid \mathbf{R}(k) \mid]\right\} \\ &= -\frac{1}{2} ln\ [(2\pi e)^n \mid \mathbf{R}(k) \mid]. \end{aligned} \tag{4.29}$$

And so we define $\mathbf{i}(k)$, the information metric for information contained in the observation at time k *only* as

$$\mathbf{i}(k) \triangleq -\frac{1}{2} ln\ [(2\pi e)^m \mid \mathbf{R}(k) \mid]. \tag{4.30}$$

If the observation noise matrix $\mathbf{R}(k)$ is not dependent on the actual observations, then this quantity can be pre-computed.

We define similar metrics for the decentralized information filter: At each sensor node i, a metric for all the *global* information up to time k is given by

$$\mathbf{I}_i(k) \triangleq \mathcal{I}(p(\mathbf{x}_i(k) \mid \{\mathbf{Z}^k\})) = -\frac{1}{2} ln\left[(2\pi e)^n \mid \mathbf{P}_i(k \mid k) \mid\right], \tag{4.31}$$

and the global information metric at the kth step given only information up to the $(k-1)$th time step is

$$\mathbf{I}_i(k-1) \triangleq \mathcal{I}(p(\mathbf{x}_i(k) \mid \{\mathbf{Z}^{k-1}\})) = -\frac{1}{2} ln\left[(2\pi e)^n \mid \mathbf{P}_i(k \mid k-1) \mid\right]. \tag{4.32}$$

We also define a partial (local) information metric for information up to time k as

$$\tilde{\mathbf{I}}_i(k) \triangleq \mathcal{I}(p(\mathbf{x}_i(k) \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k))) = -\frac{1}{2} ln\left[(2\pi e)^n \mid \tilde{\mathbf{P}}_i(k \mid k) \mid\right], \tag{4.33}$$

and define a metric for sensor i's observation information at time step k as

$$\mathbf{i}_i(k) \triangleq \mathcal{I}(\alpha_i p(\mathbf{z}_i(k) \mid \mathbf{x}_i)) = -\frac{1}{2} ln\left[(2\pi e)^n \mid \mathbf{R}_i(k) \mid\right]. \tag{4.34}$$

4.3.2 Discrete Classification Metrics

For the classification algorithm, Fisher information cannot be used to quantify information. This is because when $\mathbf{x}$ is discrete, the log-likelihood is not continually differentiable in $\mathbf{x}$ which is a necessary condition in the definition of Fisher information. Hence, we turn again to entropy as defined for discrete states in Equation 2.42.

1. **Posterior entropy and Information.** The posterior entropy is given by

$$h(k) \triangleq E\{-ln\, p(\mathbf{x} \mid \mathbf{Z}^k)\} = -\sum_b p(X_b \mid \mathbf{Z}^k)\, ln\, \left[p(X_b \mid \mathbf{Z}^k)\right], \tag{4.35}$$

and so the information metric for all the information concerning the state $\mathbf{x}$ available up to and including k is given by

$$\mathbf{I}(k) = -h(k). \tag{4.36}$$

As with the information filter, it must be noted that if a non-informative prior is used at initialization, then $\mathbf{I}(k)$ represents *all* the observation information up to and including time-step k.

2. **Prior entropy and Information.** Similarly, the prior entropy is given by

$$h(k-1) \triangleq E\{-ln\, p(\mathbf{x} \mid \mathbf{Z}^{k-1})\} = -\sum_b p(X_b \mid \mathbf{Z}^{k-1})\, ln\, \left[p(X_b \mid \mathbf{Z}^{k-1})\right], \tag{4.37}$$

and the corresponding information metric is $\mathbf{I}(k-1) = -h(k-1)$.

3. **Likelihood information.** The information metric for the observation information at time k is defined by

$$\begin{aligned} \mathbf{i}(k) &= E\left\{ln\, \left[\frac{p(\mathbf{z}(k) \mid \mathbf{x})}{p(\mathbf{z}(k))}\right]\right\} = \sum_b \frac{p(\mathbf{z}(k) \mid X_b)}{p(\mathbf{z}(k))}\, ln\, \left[\frac{p(\mathbf{z}(k) \mid X_b)}{p(\mathbf{z}(k))}\right] \\ &= \sum_b \alpha \Lambda_{\mathbf{z}(k)}(X_b)\, ln\, \left[\alpha \Lambda_{\mathbf{z}(k)}(X_b)\right]. \end{aligned} \tag{4.38}$$

In the same manner, we can define metrics for the decentralized classification algorithm: The metric for global information at sensor node i up to time k is given by

$$\mathbf{I}_i(k) = \sum_b p(X_{ib} \mid \{\mathbf{Z}^k\})\, ln\, \left[p(X_{ib} \mid \{\mathbf{Z}^k\})\right], \tag{4.39}$$

and for the partial classification information at time k by

$$\tilde{\mathbf{I}}_i(k) = \sum_b p(X_{ib} \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k))\, ln\, \left[p(X_{ib} \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k))\right], \tag{4.40}$$

and finally, for the observed information at time k it is given by

$$\mathbf{i}_i(k) = \sum_b \alpha_i \Lambda_{\mathbf{z}_i(k)}(X_{ib})\, ln\, \left[\alpha_i \Lambda_{\mathbf{z}_i(k)}(X_{ib})\right]. \tag{4.41}$$

4.4 Towards Decentralized Sensor Management

In the work that follows, we make presentations in terms of results and components of the information filter, on the understanding that the same can be done for the classification algorithm unless otherwise stated. The work presented is in terms of the decentralized architecture and algorithms which result from considering the Independent Likelihood Pool (See Chapter 2 and 3). In setting out our solution to the decentralized sensor management problem, we make the following assumption:

Assumption 2 All sensors make conditionally independent observations in order to make inferences about the *same* state $\mathbf{x}$.

This is a simplification which allows direct comparisons of the expected utilities by ensuring that all the distributions pertain to the same state $\mathbf{x}$, thus pointing clearly to the optimal action. Implicit in this is an assumption that all the data is correctly associated.

4.4.1 Available Nodal (Local) Information

In this section we outline the information available for decision making at each decentralized sensor node. Subsequent to data fusion communication and assimilation, each sensor node i has the following probabilistic information:

1. Information directly concerning sensor i, that is

$$\mathbf{I}_i(k), \quad \mathbf{I}_i(k-1), \quad \tilde{\mathbf{I}}_i(k) \text{ and } \mathbf{i}_i(k), \tag{4.42}$$

based on the global posterior and prior, the partial posterior and the likelihood respectively.

2. As a direct consequence of data fusion communication, each sensor has likelihood information from *all* the sensors in the system. This information is in the form of the log-likelihoods

$$ln\,[\alpha_j\, p(\mathbf{z}_j(k) \mid \mathbf{x}_j(k))], \quad \forall j \in \mathcal{N}$$

and in terms of the algorithms this information is modelled by

$$\mathbf{H}_j^T(k)\mathbf{R}_j^{-1}(k)\mathbf{H}_j(k) \quad \text{and} \quad ln\,\left[\alpha_j \lambda_{\mathbf{z}_j(k)}(X_{jb})\right], \quad \forall j \in \mathcal{N} \tag{4.43}$$

for the decentralized information filter and classification algorithms respectively. Therefore, each sensor i can compute all the other sensors' observation information metrics

$$\mathbf{i}_j(k), \quad \forall j \in \mathcal{N}. \tag{4.44}$$

In addition to this information, each sensor i is able to reconstruct the partial posteriors for all the nodes $j \in \mathcal{N}$. This is made possible by the assertion that for a fully-connected system, the global posteriors and priors are identical at every node. The reconstruction is as follows; each node j computes its partial posterior according to

$$ln\, p(\mathbf{x}_j \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_j(k)) = ln\, p(\mathbf{x}_j \mid \{\mathbf{Z}^{k-1}\}) + ln\, [\alpha_j\, p(\mathbf{z}_j(k) \mid \mathbf{x}_j)]. \tag{4.45}$$

The last term on the right hand side is the likelihood which is communicated by j to all the other sensors for data fusion purposes. The first term on the right hand side is the global prior which is the same at every node. Hence, sensor i after communicating with j (that is, having received j's likelihood), has all the information necessary to compute j's partial posterior as in Equation 4.45. This is illustrated for the information filter as follows; from Equation 3.83, sensor j computes its partial information matrix according to

$$\tilde{\mathbf{P}}_j^{-1}(k \mid k) = \mathbf{P}_j^{-1}(k \mid k-1) + \mathbf{H}_j^T(k)\mathbf{R}_j^{-1}(k)\mathbf{H}_j(k).$$

Sensor node i can reconstruct j's partial information matrix as follows

$$(\text{at } i): \quad \tilde{\mathbf{P}}_j^{-1}(k \mid k) = \mathbf{P}_j^{-1}(k \mid k-1) + \underbrace{\mathbf{H}_j^T(k)\mathbf{R}_j^{-1}(k)\mathbf{H}_j(k)}_{\text{communicated}}.$$

The term unavailable at sensor node i in this equation is $\mathbf{P}_j^{-1}(k \mid k-1)$, but by our assertion for a fully connected system $\mathbf{P}_j^{-1}(k \mid k-1) = \mathbf{P}_i^{-1}(k \mid k-1)$. Thus sensor i can reconstruct sensor j's partial information as

$$(\text{at } i): \quad \tilde{\mathbf{P}}_j^{-1}(k \mid k) = \underbrace{\mathbf{P}_i^{-1}(k \mid k-1)}_{\text{own prior}} + \underbrace{\mathbf{H}_j^T(k)\mathbf{R}_j^{-1}(k)\mathbf{H}_j(k)}_{\text{communicated}}. \tag{4.46}$$

Therefore, in addition to being able to quantify likelihood information from every sensor j, sensor i can, if required, quantify the partial posterior information for every sensor j. The same result can also be shown for the classification algorithm. In this way, each sensor i is also able to obtain the partial information metric

$$\tilde{\mathbf{I}}_j(k), \quad \forall j \in \mathcal{N}. \tag{4.47}$$

4.4.2 Action-Outcome Association

In order to relate all of this to a decentralized sensing scenario, we develop a representative example. In the example we also introduce and develop several new concepts.

Example 2. Consider a system of 3 decentralized sensors, each of which is able to observe a single entity that we shall call a "target" with a view to estimating a state $\mathbf{x}(k)$, which in this case is the location of the platform on which the sensors are mounted. We assume that there are 3 targets, that is, the target set is $\mathcal{T} = \{t_1, t_2, t_3\}$ as illustrated in Figure 4.4.

For a sensor system, each action $a_l \in \mathcal{A}$ gives rise to a particular sensing configuration. To develop this, consider the sensing scenario, as depicted by Figure 4.4, in which each sensor can observe only a single target. Thus, all possible actions are constrained by a

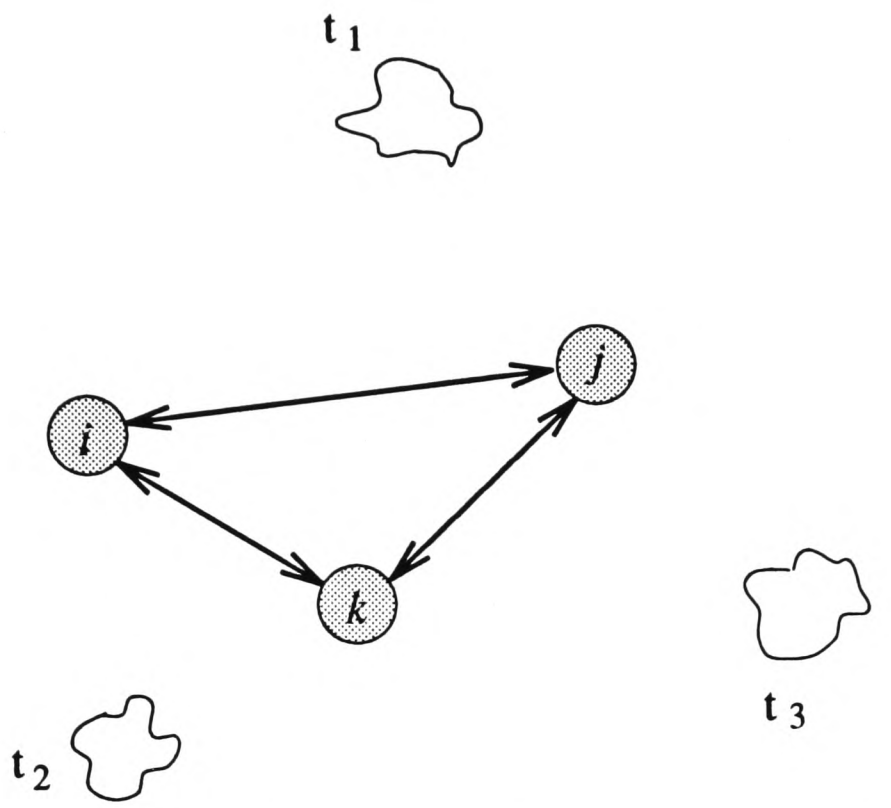


Figure 4.4: The decentralized sensor system of Example 2.

1-1 sensor-target assignment. In this case, the action set is defined by the various possible target sensor assignments. These could, for example, be

$$\mathcal{A} = \left\{ \begin{array}{l} a_1 = (t_1 \longrightarrow i;\ \ t_2 \longrightarrow j;\ \ t_3 \longrightarrow k), \\ a_2 = (t_1 \longrightarrow i;\ \ t_2 \longrightarrow k;\ \ t_3 \longrightarrow j), \\ a_3 = (t_1 \longrightarrow k;\ \ t_2 \longrightarrow j;\ \ t_3 \longrightarrow i), \\ \qquad\qquad \vdots \end{array} \right\}. \tag{4.48}$$

In maximizing the expected utilities in order to obtain the optimal action, we consider posterior information or equivalently likelihood information by the Likelihood Principle (see Equations 2.28 and 2.29 and discussion in Section 2.3.1). But first we extend our notation to the decentralized case. For each sensor i, the set of outcome PDFs resulting from action a_l is written ρ_{li} and given by

$$\rho_{li} \triangleq \{p_1, p_2, \ldots, p_q \mid a_l\}_i, \tag{4.49}$$

The set of all outcome PDFs for *all* the sensors arising from action a_l is written

$$\{\rho\}_l \triangleq \{\rho_{li}\} = \{\rho_{l1}, \rho_{l2}, \ldots, \rho_{lN}\}, \tag{4.50}$$

where each ρ_{li} is given by Equation 4.49. And so we have that:

1. Considering partial posterior information, the set $\{\rho\}_l$ when the outcome at each sensor is a single PDF, is given by

$$\begin{aligned} \{\rho\}_l &= \{p_{l1}, p_{l2}, \ldots, p_{lN}\} \\ &= \left\{p(\mathbf{x}_j \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_j(k)), a_l\right\}, \quad \forall j \in \mathcal{N}. \end{aligned} \tag{4.51}$$

2. Considering likelihood information, the set $\{\rho\}_l$ when the outcomes at each sensor are single PDFs is written

$$\{\rho\}_l = \{p(\mathbf{z}_j(k) \mid \mathbf{x}_j), a_l\}, \quad \forall j \in \mathcal{N}. \tag{4.52}$$

Although the equivalence of these two approaches has been demonstrated in Chapter 2, there is a difference in the interpretation of the information they provide. The use of Equation 4.52 means that the only significant information in the management problem is current information from observations at time step k, that is, $\mathbf{z}(j)$k. In contrast, Equation 4.51 considers a sensor's observation at time step k, that is, $\mathbf{z}_j(k)$ added to all that is currently known about the state $\mathbf{x}$ at time $k-1$, which is based on *all* the observation information up to time $k-1$, that is $\{\mathbf{Z}^{k-1}\}$. Equation 4.52 has the advantage that *no additional* computations are necessary since all the requisite information is communicated in the data fusion process. In contrast, Equation 4.51 has the disadvantage of incurring the computational overheads implicit in Equation 4.46, where the partial posteriors are re-computed at each sensor node. However, this overhead can be overcome through additional communication as we shall show later.

4.5 Determining Decentralized Solutions

4.5.1 Comparable and Non-Comparable Utility Solutions

The solution to the question of how to choose an optimal action from the acceptable class of actions given in Equation 2.33 depends on the admissibility of utility comparisons. The admissibility or inadmissibility of utility comparisons leads to different decisions and, consequently, different optimal actions. This arises from the fact that the solutions adopt differing optimizing philosophies.

The case for comparable information-based utilities.

While conceding to Arrow's impossibility theorem as regards the general validity of utility comparisons, we argue the following:

1. Our formulation of expected utilities in terms of information quantified using entropy effectively places the expected utilities on the same relative scale. This facilitates comparison of utilities from different sensors. Furthermore, based on Assumption 2. expected utilities are in the form of information contained in the same type of PDFs for each of the algorithms. In the case of the information filter for example, the distributions are Gaussian (from Chapters 2 and 3). PDFs of the same type can be compared in terms of the information that they contain about a common state. We are, thus, able to state preference unambiguously, through direct comparison of expected utilities constructed using a metric based on these PDFs which satisfies the axioms in Section 4.2.1.

2. From the discussion in Chapter 2, a decentralized sensor network is unlike a human[3] group decision making problem in that, from Equation 2.25, differences in sensors

[3]Parallels are drawn with human situations because this is were most objections to utility comparisons are raised.

are a result of different likelihood functions (observations) given the same prior (by the Likelihood Principle). This realizes precisely the hypothetical human situation described by Savage [120] in his justification for comparable utilities; that comparisons would be possible in the case of a jury whose members "... are supposed to have common value judgements in connection with legal matters..."[134]. Objectivity, which is implied by the likelihood function, is in this sense a pre-requisite for direct comparison of opinions [4] . Berger [18] assures us that the likelihood function is the only basis for objectivity as it is based *only* on observed data rather than subjective opinions.

3. Being system designers, we find the concept of subjective utility has a somewhat different meaning because we can analyse and describe it by developing sensor models. This allows us to evaluate sensor subjectivity quantitatively on a common scale[5] . Thus rather than merely describing decision-makers based on their outer exhibited preferences and behaviour we are in the unique position of being able to "open them up" and gain an understanding of how they work. Russell sums this up (in a slightly different context) by saying "...we are interested in designing agents, rather than describing agents or being agents."[119].

More importantly, a quantitative argument further strengthening the case for comparable expected utilities, can be appreciated by considering the solutions obtained in data fusion management when comparable utilities are assumed and when they are not as we now

[4]Objectivity, in this case, being simply defined as an opinion based only on observed information (data) thus avoiding the potential philosophical minefield of defining objectivity.

[5]The equivalent of this in a human situation would be to analyse quantitatively individual value judgements and how they come about, which may entail a complete psycho-analysis of an individual's whole life experiences on which judgements may be based. Objectivity in this case is impossible, given that the investigator is another such individual and not the designer!

demonstrate:

From Chapter 2 and assuming comparable utilities, the optimal action is given by maximizing Equation 2.35 which we write in the following form; the group expected utility for each action a_l is given by the summation

$$\mathcal{B}_c(a_l) = \left\{ \sum_j w_j \left[\beta_j(a_l) - c(j)\right]^\gamma \right\}^{1/\gamma}, \tag{4.53}$$

where $\beta_j(a_l)$ is sensor j's expected utility given by Equation 4.7. Setting $\gamma = 1$ and $w_j = 1/N$ a constant, and also ignoring individual security levels we have that

$$\mathcal{B}_c(a_l) = \sum_j \beta_j(a_l) = \sum_j \mathcal{I}(\rho_{lj}), \tag{4.54}$$

where $\mathcal{I}(\rho_{lj})$ is read as the information contained in sensor j's set of distributions ρ_j, which result from action a_l. Assuming non-comparable utilities, the Nash solution gives the group expected utility (Equation 2.36) for each action a_l, as

$$\mathcal{B}_{nc}(a_l) = \prod_j \left[\beta_j(a_l) - c(j)\right] = \prod_j \left[\mathcal{I}(\rho_{lj}) - c(j)\right]. \tag{4.55}$$

The security level $c(j)$ is a feature of the Nash solution and hence is retained. While the value of $c(j)$ is arbitrary, it is usually set to be agent j's current expected utility level. From this two cases arise:

1. If decision-making occurs at the start of the observation process and a non-informative prior is used, $c(j)$ can be set to zero.

2. If sensor j is already making observations at time of decision-making, $c(j)$ is set to j's current utility level.

We now illustrate the different solutions that the comparable and non-comparable utility approaches give under various conditions. We extend the example introduced in Figure 4.4.

Example 2 (*cont*). In the action-outcome association suggested by Equation 4.48, we assign values to each of the utilities for each of the sensors i, j and k. We use scaled values (for ease of illustration) of typical expected utilities and write the following:

		$\mathcal{B}_{c1}(Eqn.4.54)$	$\mathcal{B}_{c2}$	$\mathcal{B}_{nc1}(Eqn.4.55)$	$\mathcal{B}_{nc2}$
a_1 :	$t_1 \xrightarrow{20} i;\ t_2 \xrightarrow{16} j;\ t_3 \xrightarrow{18} k;$	54	4	5760	0
a_2 :	$t_1 \xrightarrow{20} i;\ t_2 \xrightarrow{14} k;\ t_3 \xrightarrow{15} j;$	49	-3	4200	60
a_3 :	$t_1 \xrightarrow{10} k;\ t_2 \xrightarrow{16} j;\ t_3 \xrightarrow{33} i;$	59	+9	5280	-528
a_4 :	$t_1 \xrightarrow{10} k;\ t_2 \xrightarrow{22} i;\ t_3 \xrightarrow{15} j;$	47	-3	3300	0
a_5 :	$t_1 \xrightarrow{10} j;\ t_2 \xrightarrow{22} i;\ t_3 \xrightarrow{18} k;$	50	0	3960	0
a_6 :	$t_1 \xrightarrow{10} j;\ t_2 \xrightarrow{14} k;\ t_3 \xrightarrow{33} i;$	57	7	4620	0

(4.56)

Notice that due to the 1-1 assignment constraint, the outcomes ρ_{li} for each sensor i for a given action a_l, are single PDFs where the PDF corresponds to the assignment of sensor i according to action a_l. We assume the sensor target assignment at time of decision making is that given by action a_5. The expected group utilities $\mathcal{B}_{c1}$ and $\mathcal{B}_{c2}$, are the solutions using comparable utilities without a security level (that is $c(j) = 0$) and with $c(j)$ set to the current expected utility respectively. The equivalent non-comparable utility solutions are given by $\mathcal{B}_{nc1}$ and $\mathcal{B}_{nc2}$ respectively. From 4.56 we can observe the following:

1. Using the solution for comparable utilities $\mathcal{B}_{c1}$, we obtain the following action preference ordering $\langle a_3, a_6, a_1, a_5, a_2, a_4 \rangle$, that is, $\hat{a}_{c1} = a_3$. With security levels $\mathcal{B}_{c2}$, we obtain the ordering $\langle a_3, a_6, a_1, a_5 \rangle$, that is, $\hat{a}_{c2} = a_3$ where the actions (a_2, a_3) are deemed inadmissible with respect to the sensor security levels.

2. Using the non-comparable utility solution with the security level set to $c(j) = 0$, we obtain the following preference ordering $\langle a_1, a_3, a_6, a_2, a_5, a_4 \rangle$, that is, $\hat{a}_{nc1} = a_1$. Increasing the security level for every sensor to the value of the least individual utility at the time of decision making, in this case 10, gives the ordering $\langle a_1, a_2, (a_3, a_4, a_5, a_6) \rangle$ and again the optimal action is a_1, where (a_3, a_4, a_5, a_6) all result in zero group expected utilities. With security levels set to the current values of each individual sensor at the time of decision making, the ordering is $\langle a_2, (a_1, a_4, a_5, a_6) \rangle$. Therefore the optimal action is a_2 and the actions (a_1, a_4, a_5, a_6) result in a zero group utility. In this case a_3 is an unacceptable action.

The results of Example 2 serve to illustrate the diversity of solutions obtained from adopting either the comparable utility solution (Equation 4.54) or the non-comparable utility solution (Equation 4.55). The solutions based on comparable utilities directly maximize group expected utility without regard for individual sensor utilities. The degree of disregard depends on the security levels set, which renders some actions inadmissible. The solutions obtained by considering non-comparable utilities optimize the decision process for the individual sensor as evidenced above. This is desirable in a social context when making decisions concerning human individuals. However, these solutions do not result in as high group utilities as the solutions assuming comparable utilities. To see this we can write that the total information in the system is given by

$$\mathcal{I}(\{\rho\}_{\hat{a}}) = \sum_j \mathcal{I}(\rho_{\hat{a}j}), \tag{4.57}$$

where $\rho_{\hat{a}j}$ is the set of outcomes at sensor j given the optimal action $\hat{a}$ as defined in Equation 4.49. By applying Equation 4.57 to the optimal solution obtained with the two approaches,

it can be seen that

$$\mathcal{I}(\{\rho\}_{\hat{a}_c}) > \mathcal{I}(\{\rho\}_{\hat{a}_{nc}}),$$

where $\hat{a}_c$ and $\hat{a}_{nc}$ are the comparable and non-comparable solutions respectively. This fact is also demonstrated by the following argument: For the comparable utility solution the optimal action is obtained by maximizing

$$\begin{aligned}\sum_j \beta_j(a_l) &= E^{p_1}\{U_1(\mathbf{x}, a_l)\} + E^{p_2}\{U_2(\mathbf{x}, a_l)\} + \cdots + E^{p_N}\{U_N(\mathbf{x}, a_l)\} \\ &= \int_{\mathbf{x}} U_1(\mathbf{x}, a_l) p(\mathbf{Z}_1^k \mid \mathbf{x}) d\mathbf{x} + \int_{\mathbf{x}} U_2(\mathbf{x}, a_l) p(\mathbf{Z}_2^k \mid \mathbf{x}) d\mathbf{x} + \cdots \\ &\quad + \int_{\mathbf{x}} U_N(\mathbf{x}, a_l) p(\mathbf{Z}_N^k \mid \mathbf{x}) d\mathbf{x}, \end{aligned} \tag{4.58}$$

which from our definition of expected utility in Equation 4.8 evaluates to

$$\begin{aligned}\sum_j \beta_j(a_l) &= \mathcal{I}(p(\mathbf{Z}_1^k \mid \mathbf{x}), a_l) + \mathcal{I}(p(\mathbf{Z}_2^k \mid \mathbf{x}), a_l) + \cdots + \mathcal{I}(p(\mathbf{Z}_N^k \mid \mathbf{x}), a_l) \\ &= \sum_j \mathcal{I}(p(\mathbf{Z}_j^k \mid \mathbf{x}), a_l), \end{aligned} \tag{4.59}$$

thus

$$\hat{a}_c = \arg\max_{a_l} \sum_j \mathcal{I}(p(\mathbf{Z}_j^k \mid \mathbf{x}), a_l), \tag{4.60}$$

in which we have assumed the decisions to be based on likelihood information up to time k. This can be generalized to when the set ρ_{lj} contains more than one element and becomes $\{p(\mathbf{Z}_j^k \mid \mathbf{x})\}$. In which case the action $\hat{a}_c$ is given by

$$\hat{a}_c = \arg\max_a \sum_j \beta_j(a_l) = \sum_j \mathcal{I}\left(\left\{p(\mathbf{Z}_j^k \mid \mathbf{x})\right\}, a_l\right). \tag{4.61}$$

From Equation 4.57, clearly $\hat{a}_c$ directly maximizes the *total* information in the system.

On the other hand, with the solution assuming non-comparable utilities, we have the following (ignoring $c(j)$)

$$\prod_j \beta_j(a_l) = E^{p_1}\{U_1(\mathbf{x}, a_l)\} \times E^{p_2}\{U_2(\mathbf{x}, a_l)\} \times \cdots \times E^{p_N}\{U_N(\mathbf{x}, a_l)\}$$

$$= \int_{\mathbf{x}} U_1(\mathbf{x}, a_l) p(\mathbf{Z}_1^k \mid \mathbf{x}) d\mathbf{x} \times \int_{\mathbf{x}} U_2(\mathbf{x}, a_l) p(\mathbf{Z}_2^k \mid \mathbf{x}) d\mathbf{x} \times \cdots$$
$$\times \int_{\mathbf{x}} U_N(\mathbf{x}, a_l) p(\mathbf{Z}_N^k \mid \mathbf{x}) d\mathbf{x}. \tag{4.62}$$

Again from our definition in Equation 4.8, substitutions yield

$$\hat{a}_{nc} = \arg\max_a \prod_j \beta_j(a_l) \quad = \quad \arg\max_a \prod_j \mathcal{I}\left(\left\{p(\mathbf{Z}_j^k \mid \mathbf{x})\right\}, a_l\right). \tag{4.63}$$

In an information sense, the result in Equation 4.63 is clearly absurd since from information theory information is additive and not multiplicative [39]. This calls into question an approach which maximizes a product of information quantities. Consequently, the total information resulting from maximization of a product of information terms is clearly not optimal given system information as the criteria for optimality.

On the basis of the preceding discussion where the system is estimating the *same* state $\mathbf{x}$ (Assumption 2), it is evident that the solution assuming comparable utilities gives the most "optimal" solution in terms of maximizing system information gain.

4.5.2 Formulation In Terms of Fusion Algorithms

For a sensor i in a decentralized system, the expected utility of tracking a target t corresponding to action a_l is given by

$$\beta_i(a_l) = \tilde{\mathbf{I}}_{it}(k), \tag{4.64}$$

where $\tilde{\mathbf{I}}_{it}(k)$ is as given in Equation 4.33. If we only consider observation information at time k we have

$$\beta_i(a_l) = \mathbf{i}_{it}(k), \tag{4.65}$$

where $\mathbf{i}_{it}(k)$ is as given in Equation 4.34. In general, if each sensor i is capable of observing *concurrently* several targets then an action a_l which entails the tracking of several targets

has as an outcome a set of PDFs, that is, ρ_{li}. The elements of ρ_{li} are PDFs corresponding to each target tracked by i as prescribed by action a_l. Given that sensor i observes targets in the set $\{t_1, t_2, \cdots, t_q\}$, the total expected utility at i is now given by the summation of the individual target utilities that is

$$\begin{aligned}
\beta_i(a_l) &= \tilde{\mathbf{I}}_{it_1}(k) + \tilde{\mathbf{I}}_{it_2}(k) + \cdots + \tilde{\mathbf{I}}_{it_q}(k) \\
&= -\frac{1}{2} ln \left[(2\pi e)^n \mid \tilde{\mathbf{P}}_{it_1}(k \mid k) \mid \right] - \frac{1}{2} ln \left[(2\pi e)^n \mid \tilde{\mathbf{P}}_{it_2}(k \mid k) \mid \right] - \\
&\quad \cdots - \frac{1}{2} ln \left[(2\pi e)^n \mid \tilde{\mathbf{P}}_{it_q}(k \mid k) \mid \right] \\
&= -\frac{1}{2} ln \left[(2\pi e)^{nq} \mid \tilde{\mathbf{P}}_{it_1}(k \mid k) \mid \mid \tilde{\mathbf{P}}_{it_2}(k \mid k) \mid \cdots \mid \tilde{\mathbf{P}}_{it_q}(k \mid k) \mid \right], \qquad (4.66)
\end{aligned}$$

where $\tilde{\mathbf{P}}_{it_q}(k \mid k)$ is the partial covariance when sensor i is tracking target t_q. By considering only observation information at time k, we have that the total expected utility for sensor i is given by

$$\begin{aligned}
\beta_i(a_l) &= \mathbf{i}_{it_1}(k) + \mathbf{i}_{it_2}(k) + \cdots + \mathbf{i}_{it_q}(k) \\
&= -\frac{1}{2} ln \left[(2\pi e)^{nq} \mid \mathbf{R}_{it_1}(k) \mid \mid \mathbf{R}_{it_2}(k) \mid \cdots \mid \mathbf{R}_{it_q}(k) \mid \right]. \qquad (4.67)
\end{aligned}$$

Using these methods, each sensor is able to compute its own expected utility β_i for each action a_l. From the discussion in Section 4.5.1, the solution to the decentralized management problem involves the maximization of $\sum_j \beta_j(a_l)$ for each a_l. Therefore, for each sensor i to compute the most optimal group action $\hat{a}$, it requires the expected utilities $\beta_j(a_l), \ \forall j \in \mathcal{N}$. How the expected utilities of other sensors are obtained is the subject of the next section.

Example 2 (*cont*). In the light of the preceding discussion, each of the expected utilities for this example given in Equation 4.56 were obtained as follows: In Figure 4.4 the sensors estimate a state $\mathbf{x}(k) = [x, y]^T$ and each sensor makes observations $\mathbf{z}_{it}(k) = [z_{i1}, z_{i2}]^T$ where

these observations are made with noise covariance

$$\mathbf{R}_{it}(k) = \begin{bmatrix} \sigma^2_{z_{i1}} & 0 \\ 0 & \sigma^2_{z_{i2}} \end{bmatrix}. \tag{4.68}$$

Therefore, as an example

$$\beta_i(a_1) = -const\ \frac{1}{2} ln \left[(2\pi e)^2 \mid \mathbf{R}_{it_1}(k) \mid \right] = 20.0, \tag{4.69}$$

where *const* is simply a scalar constant.

4.6 Realizing Decentralized Management

4.6.1 Computation, Communication and Bargaining

Information is required about every sensor's utilities for the various actions in order for an optimal decision to be made according to Equation 4.61. Sensor i can obtain other sensors' expected utilities either by *computing* them or having them *communicated* to it by the other sensors themselves. However, in both cases sensor i, in addition to evaluating its expected utility for its current tasks, has to compute its expected utility for tasks currently being performed by sensor j, $\forall j \in \mathcal{N}$. This means that if sensor j is currently observing a target t, sensor i has to evaluate its utility were it to observe that target t. In order to facilitate this, information is required which gives i a basis for evaluating its utility for j's tasks. The most basic information on which such evaluations may, in general, be based are the observations themselves. This can be improved upon for specific applications, for example evaluation of i's expected utility for j's tasks can be based on sector or grid information whereby i evaluates its ability to make observations in the particular sector where j is currently making observations. However, such approaches are in general related to the observations themselves we shall for purposes of this discussion consider communication of the observations themselves.

We consider firstly, the approach in which each sensor computes every sensor's expected utility for each action and secondly, the approach where each sensor only computes its own expected utility for each action and communicates it. We outline these approaches assuming that management decisions are to be based only on observation information at time k (Equation 4.52). The same methods can be used when decisions are based on observation information up to k (Equation 4.51).

Computing *all* expected utilities

In order for sensor i to compute $\beta_j(a_l)$, $\forall j \in \mathcal{N}$, the required information is $\mathbf{R}_{jt}(k)$ for each target t being tracked by each sensor j. This information is available in the form $\mathbf{H}_j(k)^T\mathbf{R}_{jt}^{-1}(k)\mathbf{H}_j(k)$, from the data fusion communications of Equation 4.43.

The maximization in Equation 4.61 over the entire action set $\mathcal{A}$ involves considering the expected utilities resulting from all the possible actions. This means that each sensor i needs to evaluate its own expected utility of tracking targets currently being observed by each sensor j, $\forall j \in \mathcal{N}$. In the simple case when each sensor i's observation noise covariance is simply $\mathbf{R}_i(k)$, these expected utilities can be pre-computed. However, in a real application the observation noise covariance may be observation dependent (and by implication also state dependent) and written as $\mathbf{R}_i[k, \mathbf{z}_i(k)]$. It is thus necessary to communicate to i the observation vector for the target for which an expected utility is to be computed. In addition, for the sensor to compute the expected utilities of the other sensors, the observation models of all the sensors must be communicated! Having computed the expected utilities of each sensor, sensor i is then able to compute the most optimal action $\hat{a}_c$. This approach is illustrated in Figure 4.5. The severity of the computation required with this computation approach can be appreciated as follows; for a system with the target set $\mathcal{T} = \{t_1, \ldots, t_q\}$,

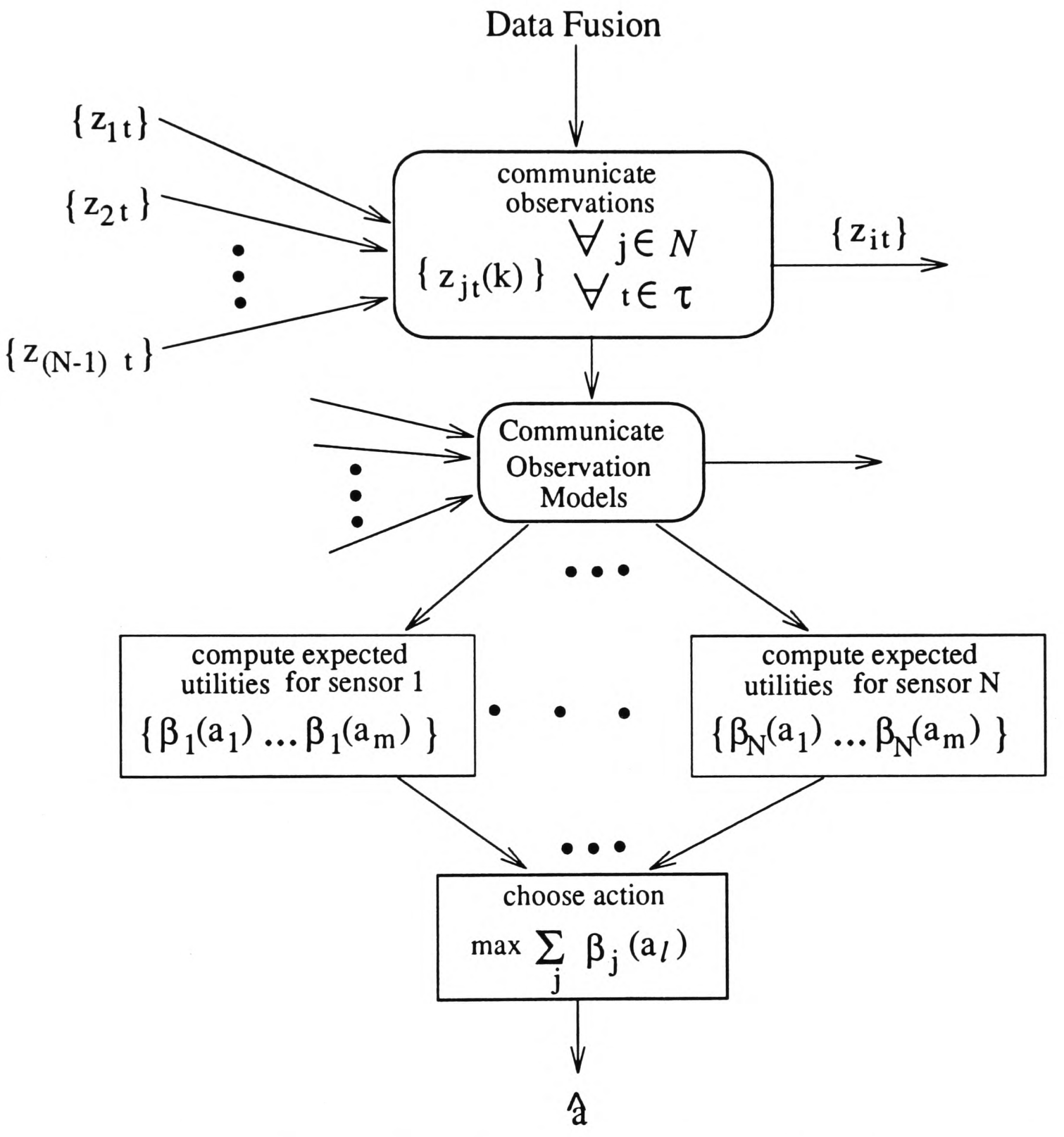


Figure 4.5: Computing the expected utilities for every sensor in the system at sensor i.

each sensor i performs the following computations:

1. In computing its own expected utilities for the targets, sensor i evaluates the metric $\mathbf{i}_i(k)$ q times.

2. In computing the expected utilities of the other sensors for the target set $\mathcal{T}$, sensor i computes the metric $\mathbf{i}_j(k)$ q times for each target and hence $q(N-1)$ times for all the sensors.

Therefore, each sensor repeats the same computations Nq times. It is evident that there is a lot of replication in the computations performed at each sensor node. In addition the communication of observation models violates the modular and autonomous nature of the decentralized system. The situation is exacerbated when decisions are to be based on information up to time k, that is, $\tilde{\mathbf{I}}_{jt}(k)$. This is because additional computations are required at each sensor as discussed in Section 4.4.1 to re-compute the partial posterior. These realizations lead to the next approach.

Communicating *all* expected utilities

With this approach, each sensor i communicates its *own* expected utilities $\beta_i(a_l), \forall a_l \in \mathcal{A}$. In order to be able to compute its expected utilities for targets currently being observed by other sensors the sensor i requires knowledge of their observations. Thus the observations for all the targets are communicated by the sensor currently observing them. However, there is no need with this approach to communicate observation models. The method proceeds as shown in Figure 4.6. Each sensor computes its own expected utilities for the different actions a_l and communicates these to every other sensor in the system. On receiving the expected utilities of other nodes each sensor is thus able to compute the maximization of

Equation 4.61 from which the optimal action $\hat{a}_c$ can be obtained.

With this approach, the only expected utilities that sensor i has to compute are its own. This avoids the need to communicate observation models as in the previous method. The overhead incurred with this approach is the communication of expected utilities by every sensor. Since the expected utility for a given action is a numeric value, the information to be communicated by each sensor is simply a set of numbers whose cardinality depends on the number of distinct elements in $\mathcal{A}$. This, depending on complexity, may be a smaller overhead than that incurred in the previous method where observation models are communicated. This method also maintains the modularity of the decentralized system by keeping observation models local to each sensor. The replication in the computations of the previous method are eliminated. Whereas in Figure 4.5 for q targets each node computes expected utilities Nq times, in Figure 4.6 each node only computes this q times.

Towards a bargaining solution

The communication method suggests a further improvement as follows; suppose we have that each sensor computes the following ordered set by considering each $a_l \in \mathcal{A}$

$$\left\langle \beta_i(a^{i1}), \beta_i(a^{i2}), \ldots, \beta_i(a^{ir}) \right\rangle, \quad \text{where } \beta_i(a^{i1}) \succeq \beta_i(a^{i2}) \succeq \cdots \succeq \beta_i(a^{ir}), \tag{4.70}$$

such that a^{i1} is sensor i's first preference action and a^{ir} is i's least preferred action. The communication required in Figure 4.6 could be reduced if each sensor i only communicates a subset of the expected utility set corresponding to preferred actions. Such a subset could be chosen by considering only those actions which result in expected utilities greater than or equal to i's current expected utility that is

$$\{\beta_i(a_l) : \beta_i(a_l) \geq c(i)\}, \tag{4.71}$$

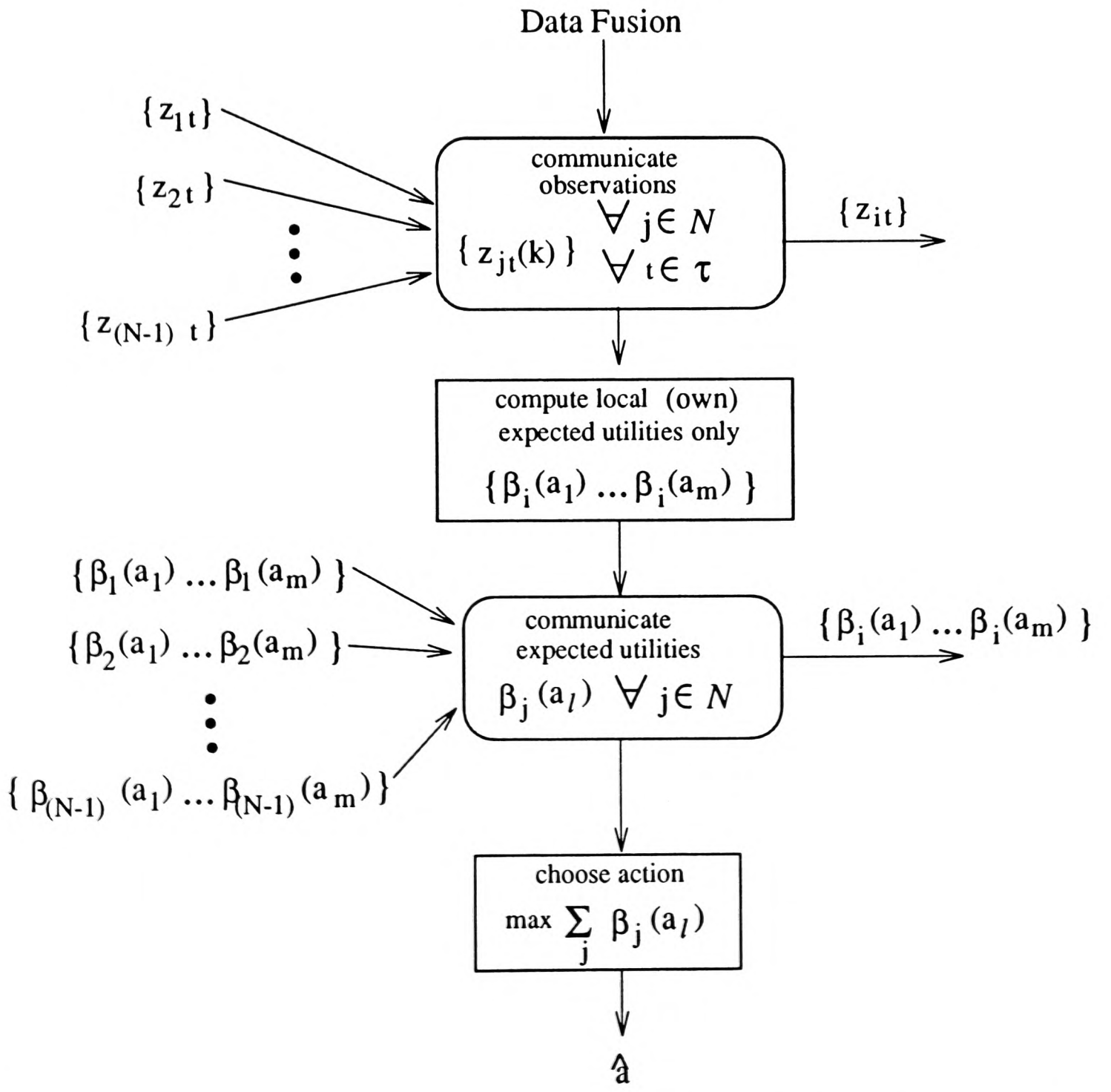


Figure 4.6: Communicating expected utilities for sensor i locally and then communicating to other sensors.

where $c(i)$ is sensor i's current utility at time of decision making. Another possibility is to only consider expected utilities above a given threshold value. The effect of this is to reduce the size of the expected utility set considered and hence the number of possible actions to consider.

Example 2 (*cont*). For this example, the sensor action preferences in order are

$$i : \langle (a_3, a_6), (a_4, a_5), (a_1, a_2) \rangle$$

$$j : \langle (a_3, a_1), (a_4, a_2), (a_5, a_6) \rangle$$

$$k : \langle (a_5, a_1), (a_6, a_2), (a_3, a_4) \rangle ,$$

where each tuple represents a pair for which the preferences, that is, the expected utilities are the same. It can be seen (Figure 4.7) that by communicating only the expected utilities

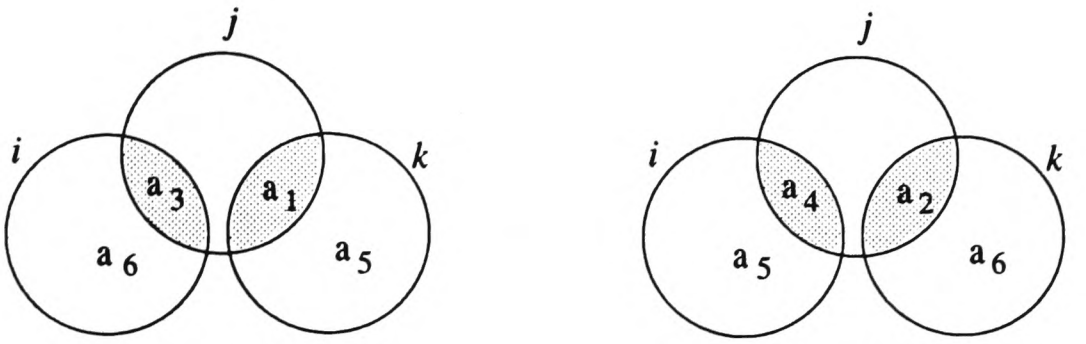


Figure 4.7: 1st and 2nd action preferences for **Example 2**.

corresponding to their first preference actions, the solution obtained from maximizing on these, that is a_3, is the same as the one obtained by maximizing over all possible actions. This is a fortuitous result for this particular example. However, the idea of considering preferences in order, leads to the development of an iterative algorithm in the next section.

4.6.2 An Iterative Bargaining Algorithm

We can develop an iterative algorithm akin to bargaining, which updates the globally preferred action each cycle as follows:

1. **Order preferences.** Each sensor i computes its ordered expected utility set and the corresponding ordered preferred action set that is

$$\left\langle \beta_i(a^{i1}), \ldots, \beta_i(a^{ir}) \right\rangle \text{ and } \left\langle a^{i1}, \ldots, a^{ir} \right\rangle . \tag{4.72}$$

2. **Communicate first preferences.** Each sensor i communicates its most preferred action a^{i1} and the corresponding expected utility $\beta_i(a^{i1})$ to other sensors. In the event that there are two actions with an identical expected utility (As in Example 2.) then both the actions are communicated.

3. **Compare first preferences.** Each sensor i then compares its own first preference a^{i1} with other received first preferences a^{j1}, $\forall j \in \mathcal{N}$, if all refer to the same action then $\hat{a} = a^{i1}$. In practice this will be unlikely except for very simple management problems. If the communicated preferences do not refer to the same action then each sensor i must communicate its expected utility corresponding to each received preferred action a^{j1}, $\forall j \in \mathcal{N}$.

4. **Maximize on first preferences.** Each sensor is then able to compute the group expected utility

$$\mathcal{B}_c(a^{j1}) = \beta_1(a^{j1}) + \beta_2(a^{j1}) + \cdots + \beta_N(a^{j1}), \tag{4.73}$$

corresponding to the preferred action for each sensor, that is, for each a^{j1}, $\forall j \in \mathcal{N}$. From this each sensor i can find the most optimal solution based on all the sensors'

first preferences as

$$\arg\max_{a^{j1}} \mathcal{B}_c(a^{j1}).$$

This completes the first iteration of the bargaining process.

5. **Repeat and maximize on subsequent preferences...**

Each sensor is thus able to compute $\mathcal{B}_c(a^{j2})$, $\forall j \in \mathcal{N}$ and so on. The bargaining process continues until there are no more actions to consider. Iterated to completion, that is, when all the actions in $\mathcal{A}$ have been considered, the bargaining algorithm is *exactly* equivalent to the method communicating all expected utilities. This algorithm is illustrated in Figure 4.8.

Example 2 (*cont*). In this example, the first iteration of the bargaining would yield; considering i's most preferred actions a^{i1}

$$\begin{aligned}
\mathcal{B}_c(a^{i1}) &= \mathcal{B}_c(a_3) = \beta_i(a_3) + \beta_j(a_3) + \beta_k(a_3) = 59 \\
\mathcal{B}_c(a^{i1}) &= \mathcal{B}_c(a_6) = \beta_i(a_6) + \beta_j(a_6) + \beta_k(a_6) = 57,
\end{aligned}$$

considering j's most preferred actions a^{j1}

$$\begin{aligned}
\mathcal{B}_c(a^{j1}) &= \mathcal{B}_c(a_3) \quad \text{(already computed)} \\
\mathcal{B}_c(a^{j1}) &= \mathcal{B}_c(a_1) = \beta_i(a_1) + \beta_j(a_1) + \beta_k(a_1) = 54,
\end{aligned}$$

considering k's most preferred actions a^{k1}

$$\begin{aligned}
\mathcal{B}_c(a^{k1}) &= \mathcal{B}_c(a_1) \quad \text{(already computed)} \\
\mathcal{B}_c(a^{k1}) &= \mathcal{B}_c(a_5) = \beta_i(a_5) + \beta_j(a_5) + \beta_k(a_5) = 50.
\end{aligned}$$

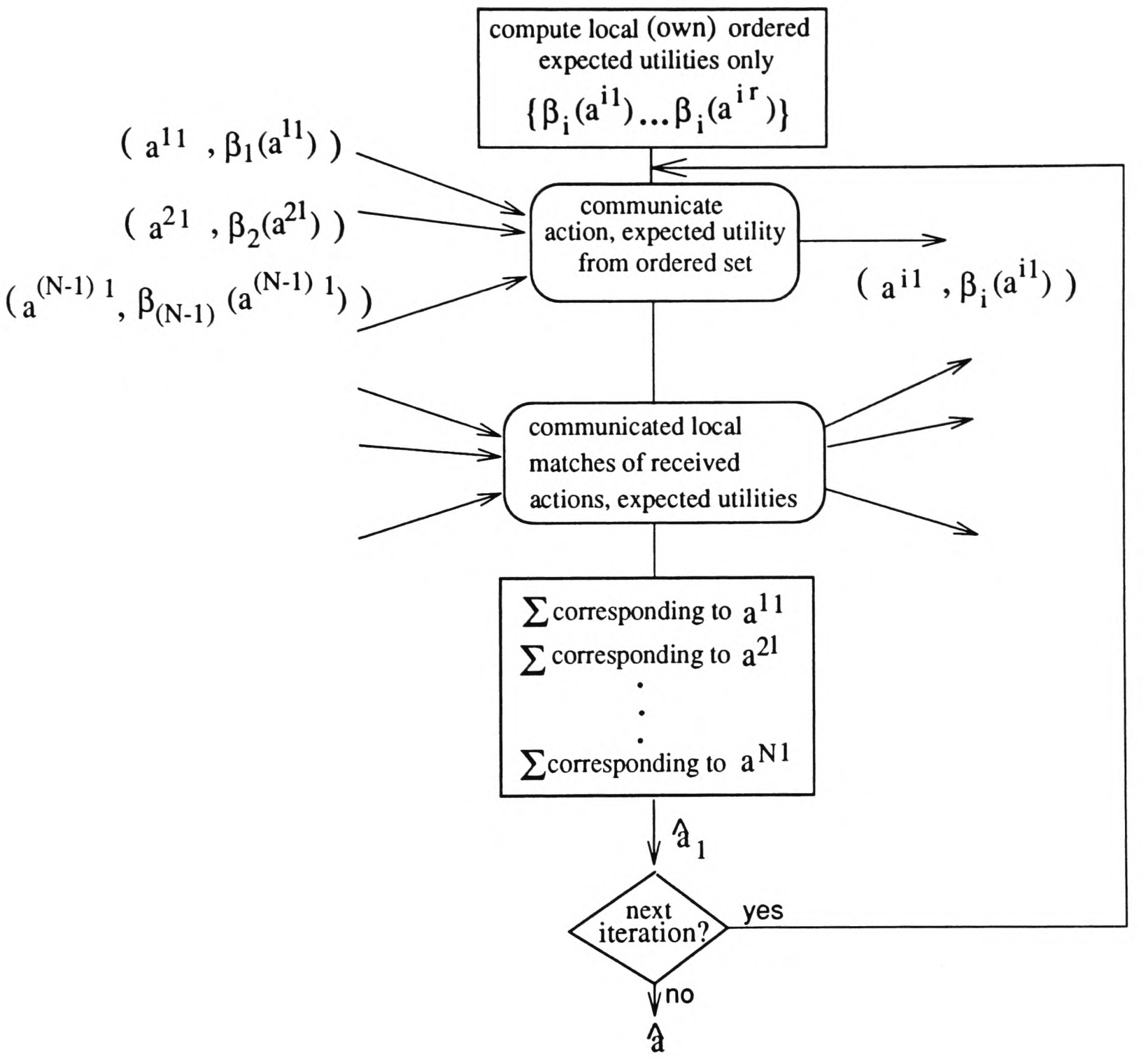


Figure 4.8: A bargaining algorithm. Bargaining towards the rational decision.

A second bargaining iteration yields

$$\begin{aligned} \mathcal{B}_c(a^{i2}) &= \mathcal{B}_c(a_5) \text{ (already computed)} \\ \mathcal{B}_c(a^{i2}) &= \mathcal{B}_c(a_4) = \beta_i(a_4) + \beta_j(a_4) + \beta_k(a_4) = 47, \end{aligned}$$

considering j's 2nd most preferred actions a^{j2}

$$\begin{aligned} \mathcal{B}_c(a^{j2}) &= \mathcal{B}_c(a_4) \text{ (already computed)} \\ \mathcal{B}_c(a^{j2}) &= \mathcal{B}_c(a_2) = \beta_i(a_2) + \beta_j(a_2) + \beta_k(a_2) = 49, \end{aligned}$$

considering k's 2nd most preferred actions a^{k2}

$$\begin{aligned} \mathcal{B}_c(a^{k2}) &= \mathcal{B}_c(a_2) \text{ (already computed)} \\ \mathcal{B}_c(a^{k2}) &= \mathcal{B}_c(a_6) \text{ (already computed).} \end{aligned}$$

Thus a maximization based only on the first iteration would in this example yield the solution $\hat{a}_c = a_3$ as before. This highlights the fact that for relatively simple management problems, there is rarely a need to have more than a single iteration of the bargaining process. However, for management problems where there are a large number of sensors and a large action set, more than one iteration may be required as the following example illustrates:

Example 3. Consider a system of 4 sensor nodes making observations of targets such that there are 8 distinct sensing configurations possible, that is, $\mathcal{A} = \{a_1, \ldots, a_8\}$. The following

are the ordered sensor expected utilities for each of the actions:

sensor i	sensor j	sensor k	sensor l
$a_1 - 20$	$a_5 - 40$	$a_3 - 20$	$a_7 - 35$
$a_2 - 15$	$a_6 - 35$	$a_4 - 18$	$a_2 - 30$
$a_3 - 10$	$a_1 - 20$	$a_5 - 16$	$a_3 - 10$
$a_4 - 5$	$a_2 - 15$	$a_6 - 14$	$a_5 - 5$
$a_5 - 4$	$a_8 - 10$	$a_7 - 12$	$a_6 - 4$
$a_6 - 3$	$a_4 - 5$	$a_8 - 10$	$a_4 - 3$
$a_7 - 2$	$a_3 - 2$	$a_1 - 8$	$a_8 - 2$
$a_8 - 1$	$a_7 - 1$	$a_2 - 6$	$a_1 - 1.$

(4.74)

The 1st bargaining iteration yields.

$$\begin{aligned}
\mathcal{B}_c(a^{i1}) &= \mathcal{B}_c(a_1) = 49 \\
\mathcal{B}_c(a^{j1}) &= \mathcal{B}_c(a_5) = 65 \longleftarrow \hat{a} \\
\mathcal{B}_c(a^{k1}) &= \mathcal{B}_c(a_3) = 42 \\
\mathcal{B}_c(a^{l1}) &= \mathcal{B}_c(a_7) = 50.
\end{aligned}$$

Thus according to the first iteration the optimal action is a_5. However, proceeding to the 2nd iteration yields

$$\begin{aligned}
\mathcal{B}_c(a^{i2}) &= \mathcal{B}_c(a_2) = 66 \longleftarrow \hat{a} \\
\mathcal{B}_c(a^{j2}) &= \mathcal{B}_c(a_6) = 56 \\
\mathcal{B}_c(a^{k2}) &= \mathcal{B}_c(a_4) = 31 \\
\mathcal{B}_c(a^{l2}) &= \mathcal{B}_c(a_2) \text{ (already computed)},
\end{aligned}$$

showing that the optimal action is updated to a_2. A third bargaining iteration yields actions that have already been considered and similarly for the 4th iteration. In the 5th iteration all the actions have been considered except a_8 which yields

$$\mathcal{B}_c(a^{j5}) = \mathcal{B}_c(a_8) = 30.$$

The number of iterations required in the bargaining depends on the level of optimality required in the actions taken. This leads to the next discussion.

4.6.3 Rationality and Optimality

In the bargaining algorithm just described, the question arises as to how far to take the bargaining. Clearly, in the blind pursuit of "rationality" in the purely decision-theoretic sense, the bargaining should proceed until there are no more actions to consider. This makes the bargaining algorithm equivalent to the algorithm communicating all the utilities (Figure 4.6). Given the ordered nature of the bargaining process, the chances of finding an optimal solution by considering 2nd, 3rd iterations and so on diminish but cannot be completely ruled out. An appropriate example is that in Example 4; here the first iteration yields a group expected utility of 65 and a second iteration improves this with an action yielding 66. Two important questions arise:

- Was it worth the additional effort in computation and communication to obtain this optimality? To answer this in specific terms requires knowledge of the specific system; matrix and vector sizes *et cetera.*

- How optimal is the decision-theoretic solution $\hat{a}$? The optimal action $\hat{a}$ is really only optimal at time step k and so the longer we take to compute it the less optimal it may become depending on how the expected utilities vary with time.

The two questions raised come about because rationality in the decision-theoretic sense associates optimality only with the action which is finally taken rather than also taking cognizance of the actual process of arriving at the optimal solution and the possibly dynamic nature of the decision problem. Up to now we have chosen to ignore these factors and assumed that the expected utilities of Section 4.5 do not change in the time it takes to

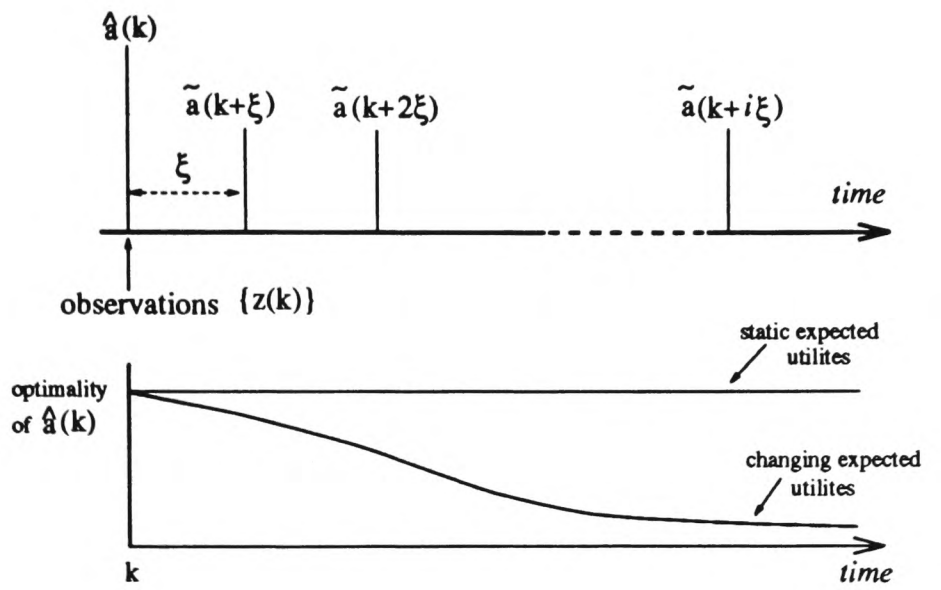


Figure 4.9: Depicting the time dependency of computing more iterations of the bargaining algorithm. Also shown is the change in the optimality of $\hat{a}$ computed at different times after the instant k for non-changing expected utilities and for changing expected utilities.

maximize Equation 4.61 to compute $\hat{a}$. For the ensuing discussion we depict the utility-time dependency as shown in Figure 4.9 where $\hat{a}(k)$ is the decision-theoretic solution at time k, $\tilde{a}(k+i\xi)$ is the solution from the ith iteration of the bargaining algorithm based on probabilistic information obtained at time k and assuming that each iteration takes a fixed time ξ to compute.

For a system with a large action set, the maximization to obtain $\hat{a}$ may be computationally significant thus making ξ significant. In a dynamic system, the actual expected utilities may be significantly different at time $k+i\xi$ corresponding to the ith iteration of the bargaining algorithm, compared to what they were at time k. Under such real constraints, it would thus appear reasonable to reconsider the notion of optimality and rationality. Such considerations form the basis for the concept of *Bounded Rationality* [124] which asserts that if an action takes infinite resources and time to compute then it is not a feasible action thus calling into question its optimality. In the same vein, Good [61] suggests maximizing expected utilities taking into account computational and temporal costs.

And so the action obtained by truncating the bargaining algorithm at a suitable point may actually be closer to the "truly" rational decision than an "optimal action" which is arrived at too late. Therefore, in deciding when to stop bargaining several factors should be taken into account; (i) the time dependency of the expected utilities, (ii) the frequency of management decisions, (iii) the computational capabilities of the decision makers, and the size of the action set and the number of sensors in the system.

4.7 Discussion

4.7.1 Is it worth the bother?

Having made decisions concerning the most optimal strategy at a given time k, the action decided upon must be put into effect. An action may require considerable effort to implement depending on the application. The question arises as to whether the additional information to be gained is worth the bother. In answer to this question, metrics for evaluating the gain in information can be developed as follows: We can compare the information being gained by maintaining the *status quo* with the information to be gained after implementing the optimal action. Therefore, if $a_{current}$ is the sensing strategy being carried out at the time of decision making, then the "value" of implementing the action $\hat{a}$ is given by

$$E^{\rho_{\hat{a}}}\left\{U(\mathbf{x},\hat{a})\right\} - E^{\rho_{a_{current}}}\left\{U(\mathbf{x},a_{current})\right\}. \tag{4.75}$$

This in fact represents the gain in information, which can be written

$$\mathcal{I}_{gain} = \mathcal{I}(\{p(\mathbf{x} \mid \{\mathbf{Z}^k\}),\hat{a}\}) - \mathcal{I}(\{p(\mathbf{x} \mid \{\mathbf{Z}^k\}),a_{current}\}), \tag{4.76}$$

or more generally

$$\mathcal{I}_{gain} = \mathcal{B}(\hat{a}) - \mathcal{B}(a_{current}), \tag{4.77}$$

using the notation of Section 4.5. The methods of Section 4.3 can be applied to quantify this information gain, based on which a decision can be made as to whether to implement the optimal action or not. In a simple scheme, the information gain $\mathcal{I}_{gain}$ can be compared to a threshold value to determine whether action $\hat{a}$ is worth being implemented.

4.7.2 Coupled Management of Data Fusion Algorithms

The methods developed are applicable to *both* the information filter and the classification algorithm even though the presentation has largely been in terms of the information filter. It is often desirable to implement both the information filter and the classification algorithm on the same sensor system for particular applications. In such systems it is equally desirable to *couple* (or combine) in some way the management of the two algorithms in a single management strategy. The ease with which this can be achieved depends largely on a consideration of Assumption 2 and is described in the following two cases.

Case 1: Whenever Assumption 2 can be guaranteed for both the classification and the information filter, the management of the two algorithms can be coupled based on developing a single information utility as follows: The information up to time step k now becomes a linear combination of that from the information filter and that from the classification algorithm as follows

$$\tilde{\mathbf{I}}_i(k) = \kappa_1 \, \tilde{\mathbf{I}}_i(k)_{est} + \kappa_2 \, \tilde{\mathbf{I}}_i(k)_{class}, \tag{4.78}$$

where $\mathbf{I}_i(k)_{est}$ and $\mathbf{I}_i(k)_{class}$ is the information from the information filter and the classification respectively. The constants κ_1 and κ_2 reflect the relative contributions of the information from the two algorithms. And similarly, for observation information at time k only

$$\mathbf{i}_i(k) = \kappa_1 \, \mathbf{i}_i(k)_{est} + \kappa_2 \, \mathbf{i}_i(k)_{class}. \tag{4.79}$$

As an example, consider a system that estimates the spatial positions of targets and at the same time attempts to classify them: In the information filter, the state may be defined by $\mathbf{x} = [x_t, y_t]^T$, where (x_t, y_t) are the 2-dimensional spatial coordinates of the target. The state for the classification algorithm may be defined as $\mathbf{x}_t = [X_1, X_2]$ where X_1 and X_2 may represent target types such as $[car, human]$ for example. Because the information filter and the classification algorithm are making inferences concerning states of the same target t, the information components may be coupled as in Equations 4.78 and 4.79. Therefore, information of Equations 4.78 and 4.79 can be interpreted as representing *all* the information known about target t. The role of sensor management now becomes one of optimizing gain of both location and identity information of the target.

Case 2: Difficulties arise in attempting to couple the management of these two algorithms when the state of nature in the one algorithm is not related to the same entity as in the other algorithm. An illustrative example of this is a sensor making observations of a target t whose position is known, in order to estimate the sensor platform's (x_s, y_s) location in space and at the same time trying to classify attributes of the target under observation. In this case, the state of nature according to the information filter may be $\mathbf{x}_s = [x_s, y_s]^T$ for example. The state of nature in the classification algorithm relates to the different possible attributes of t such as $\mathbf{x}_t = [plane, corner]$ for example. Thus the information metrics $\mathbf{i}_i(k)_{est}$ and $\mathbf{i}_i(k)_{class}$ are not directly related and so optimizing decisions based on their linear combination is meaningless! Under such circumstances it is not clear how to proceed with a coupled solution. One approach is to manage only one algorithm the choice of which is dependent on which type information is deemed as being of primary importance.

Throughout the remainder of this thesis we shall keep the management problem *decou-*

pled (in the above sense) whenever the sensing situation is as described in Case 2. We shall only use the coupled approach of Equations 4.78 and 4.79 only when the sensing situation is as described in Case 1.

4.7.3 Summary

To summarize; the methods presented in this chapter result in the following

- Sensors in a decentralized system are able to make non-conflicting decisions concerning sensing actions **locally** at each sensor.
- The actions chosen are guaranteed to be **consistent** throughout the system and result in sensor synergy.
- The action chosen is **optimal** in that it maximizes the information obtained by the system with regard to the state of nature.

The work presented in this chapter is applicable to a wide variety of systems including our application as discussed in Chapter 1. We now have the requisite tools to implement decentralized data fusion and manage the sensors in such a system. Thus far we have been working on the assumptions that; firstly, we have sensors capable of offering several sensing alternatives, and secondly, that the information from such sensors is well understood and objectively modelled in terms of probabilities, this being a pre-requisite for employing normative techniques. The next chapter is aimed at precisely these issues and develops a model for a sensor which can be managed and describes the information from it in probabilistic terms. This then leads to our implementation of data fusion and the sensor management demonstrations.

Chapter 5

An Autonomous Sonar Sensor for Data Fusion

Sensors make observations or measurements of physical quantities such as temperature, range, angle *et cetera*, thus providing information which may be used for inference and perception. In the context of the model presented in Chapter 2, we require to express sensor information in probabilistic terms, as the likelihood function with respect to the state of nature in which we are interested in making inferences. A sensor model entails developing an understanding of the sensed environment, the nature of the measurements provided by the sensor, the limitations of the sensor and most importantly a probabilistic understanding of the sensor in terms of uncertainty and accuracy. Moreover, the sensor model is also concerned with optimizing the information gathering activities of the sensor and reducing the uncertainty. All this is with the express aim of providing the most informative likelihood function given the sensor and its capabilities.

As discussed in Chapter 1, an application of the theoretical work in this thesis is in the area of mobile robotics. In current research on autonomous robot vehicles, a variety of sensor technologies are used, ranging from infra-red devices, laser range-finders, sonar, CCD cameras and gyroscopes. In this work, we make use of the standard Polaroid device [38]. While sonar, in the form of the standard Polaroid device, is the most widely used sensor in mobile robotics, its model is not understood well enough and its data is plagued with uncertainty thus making it an ideal candidate for further research. While research has

produced sophisticated sonar sensors capable of complete imaging, the standard Polaroid due to its extremely low cost and ease of use, has remained a popular choice in mobile robotics research. In this thesis we chose to model sonar[1] based on the Polaroid device because it allows us to explore and implement the theoretical work presented in this thesis while at the same time permitting us to present some new developments in sonar modelling and data processing, in the form of a *Tracking Sonar* which is ideal for managed data fusion.

Starting from a physical description of the device, we explain the nature of the uncertainty associated with in-air sonar. We then develop a model aimed at reducing sonar data uncertainty. Most important of all we analyse in probabilistic terms the uncertainty associated with data from the novel sonar model which we present.

5.1 Physical Model

Sonar sensing has been widely used in mobile robotics for tasks such as obstacle detection and avoidance [75][22], navigation and map-building [41][90]. The application of sonar to these tasks poses a number of problems. In particular, it is well known that range returns from sonar sensors are subject to distorting effects caused by long wave-lengths and wide beam-widths, making it difficult to apply any "ray-tracing" type algorithms to the interpretation of sonar data. Furthermore, because the rate at which range information can be obtained is physically limited by the speed of sound in air, a vehicle using sonar sensors to obtain map or guidance information is typically forced to adopt a "stop-look-move " motion strategy during task execution.

[1] Hereafter, references to sonar shall be taken to mean the standard laboratory grade Polaroid device and its associated electronics as described in [38].

5.1.1 RCD Observation Model Revisited

Sonar works by transmitting an acoustic signal which is reflected back on encountering an obstacle (feature) as illustrated in Figure 5.1(a). The receiving system detects only the first echo which exceeds a threshold setting of the receiving circuit and ignores subsequent echoes. A detailed physical model of the principle is given in Appendix B.1. The standard Polaroid device has a measuring range spanning $0.30 - 10m$, a nominal range accuracy of approximately 1% and a beam-width of approximately 10° at the -3dB level. In most indoor applications, the environmental features normally encountered are planes, corners, edges and cylinders (or arcs). For features such as planes and corners, the received echo is largely due to reflection (*reflective targets*). For edges and cylinders whose radius is very small compared to the range the received echo is due to diffraction (*diffractive targets*).

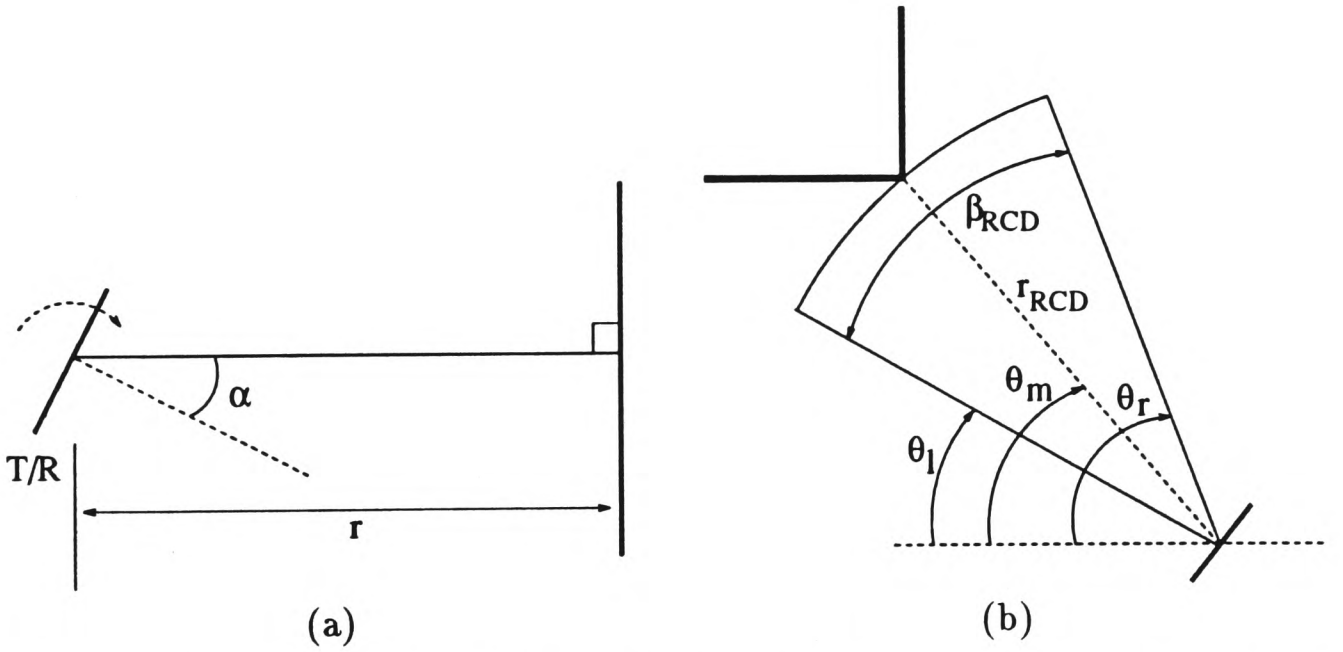


Figure 5.1: (a) Illustrates angle of inclination to the direction of propagation α (b) Parameters in the RCD model.

In developing our model we exploit the observation made by Kuc and Siegel [85] that at the wave-lengths typical of in-air sonar, common indoor environments consist predom-

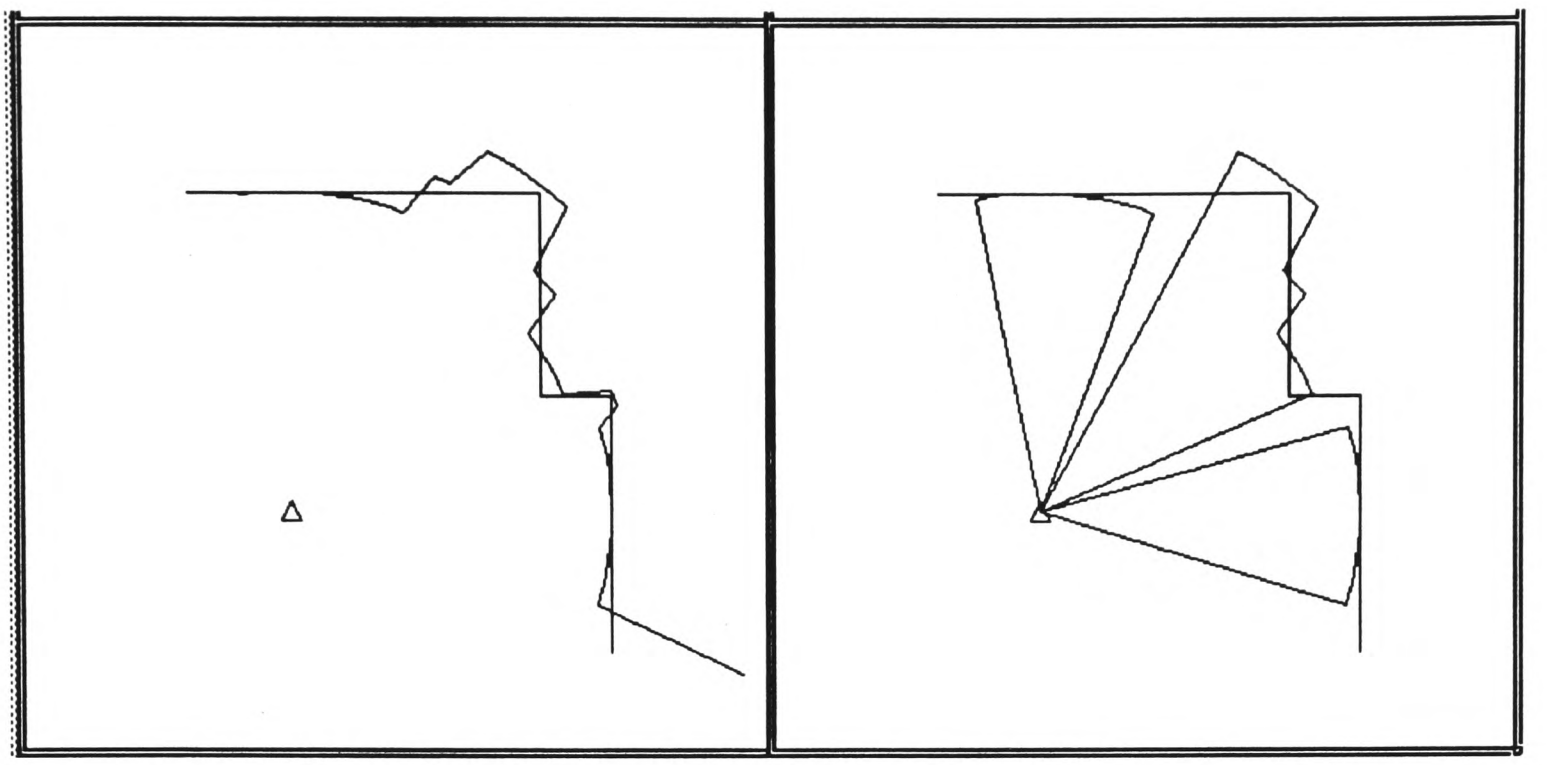


Figure 5.2: Sector scan showing (a) range information (b) RCDs extracted.

inantly of "mirror-like" reflectors which correspond to features in the environment[2]. This results in range-bearing scans composed of regions in which the measured range is constant. It has already been well-documented that given knowledge of these *Regions of Constant Depth* (RCD), accurate (sub-centimeter) and reliable maps of indoor environments may be constructed from sonar data [84][90]. An RCD is "generated" in the following way; when a scanning sonar device is aimed perpendicular to an environment feature, the range returned is indeed the true range to that feature. However, as the sonar is rotated away from the perpendicular, the center of the main lobe of the sonar beam is reflected off the feature in such a way that it is not returned to the receiver. Instead, a reflection from that part of the main lobe which is now perpendicular to the feature is returned to the receiver. Thus, the same range measurement is recorded as the sonar is rotated through an angle equal

[2]These "mirror-like" reflectors have been called specularities in the literature [22][88]. However, strictly speaking, specularity results from the highly differing acoustic impedances of air and solids [104] whereas the mirror-like reflectors are due to the wavelengths of in-air sonar and the wide beam-widths coupled with the thresholding nature of the Polaroid circuit.

to the main-lobe beam-width. The result of this is that a complete range-bearing scan is composed of a set of RCDs, one associated with each environment feature. Figure 5.2 shows a typical sonar scan together with a map of the room from which the scan was taken. A semi-circular RCD associated with each main environment feature, wall, corner or edge, can be seen.

By analysing the impulse response of sonar as done in Appendix B.1, it can be shown that the amplitude of the response decreases and its duration increases with the angle of inclination to the direction of propagation α. This means that for a given threshold setting in the receiving circuit, a feature will continue to be visible as the angle α is increased from zero to α_{max}. This is made possible by the fact that the beam-width on the Polaroid device is of the order of 10^o at the -3dB point. Hence, the correct range to a feature will be measured for angles α such that $-\alpha_{max} \leq \alpha \leq \alpha_{max}$ where α is measured from the direction of propagation to the target. The precise width of the RCD is determined by α_{max} which depends on various physical parameters of the propagated beam:

1. **Threshold of receiving circuit.** The results of Appendix B.1 show that that the received echo can be detected for increased values of α if the detection threshold is decreased and *vice versa.* Thus RCDs will appear wider with lower thresholds.

2. **Frequency.** The effect of frequency on the propagated acoustic beam and consequently α_{max} is discussed in Appendix B.1. Frequency has the effect that decreasing it will increase the RCD width for a given threshold setting in the receiving circuit.

In practice, successive range returns constitute an RCD if the absolute range difference between the minimum and the maximum of the connected set of ranges is less than some ϵ_r. This is typically of the order of the precision with which range can be measured. The

range of the RCD r_{RCD}, is given by the mode of the distribution of the ranges making up the RCD. We can define β_{RCD} the width of the RCD and β_{min} the minimum acceptable width for what constitutes an RCD. This RCD model is illustrated in Figure 5.1(b) for a symmetrical beam [3]. RCDs may be extracted from a range-bearing scan by simply segmenting range returns sequentially according to the criteria ϵ_r and rejecting those whose width is less than β_{min}.

The Limitations of Sonar

Despite the better understanding provided by the RCD model, we can identify the following shortcomings of sonar:

1. **Imprecise target bearing.** Since the location of the actual target is given by the centre of the RCD, the exact location of the target can only be determined from a scan which encompasses the bounds of the RCD. Hence with a single range return, the criticism "Sonar has poor angular resolution ..." [57] still holds in the absence of a scan establishing the RCD edges. Scanning to establish RCD edges can be a significant handicap if precise orientation information is required quickly. In addition, RCD edges tend to be unstable both as a result of range jumps induced by the thresholding nature of the receiving circuit [85] and because of partial feature obscuration. RCD instability increases when there is relative motion (such as on a vehicle). As a consequence, RCD edges are typically not a good indication of precise RCD location.

2. **Characterisation of targets.** The range and edge-angles defining an RCD $(\theta_l, \theta_r, r_{RCD})$, give no indication as to the geometric nature of the target. The geometric classification of target features is particularly important in map-building and

[3] In practise the Polaroid beam is not symmetric and varies from device to device.

navigation tasks, and so it would be desirable to obtain information from sonar to allow us to achieve this.

3. **Sonar data processing.** Since an RCD may be characterized by a single range value together with the bearing of its center (See Figure 5.1(b)), the vast majority of sonar range readings do not provide any useful additional information about the feature causing a reflection. This means that the bulk of all sonar data acquisition and processing does not actually provide any useful environment information and is simply not necessary. If, instead of taking a complete sonar scan of an environment, it were possible to take single range and bearing measurements of individual RCD reflectors, the number of range measurements required to generate a complete map of an environment could be reduced from several hundred (typical of scanning sonar systems) to about ten (the typical number of reflectors in a room). This would result in a considerable speed-up in data acquisition and consequent reduction in data-processing.

In addressing these limitations, we are motivated by Leonard's orienteering analogy "...attention should be concentrated only on information that is directly relevant to the task in hand..." [89]. Precisely, the sensor that we present is designed to detect and track RCDs as quickly as possible.

5.1.2 RCD-Monopulse Sonar

The sonar device that we develop consists of two standard Polaroid transducers mounted on a common baseline on a high-speed servo. The device is shown in Figure 5.3. One of the transducers (T/R) both transmits and receives acoustic signals, measuring time-of-flight in a conventional manner. The second transducer (R) only acts as a receiver, and measures

the time of flight from the other transmitter to itself. Therefore, the following is available;

- a conventional time-of-flight range measurement and
- a measurement of the difference in time of arrival between the two receivers.

When the target feature is centered between the receiver pair, the difference in signal arrival times is zero. As the target moves away from center, the difference signal grows linearly with off-axis bearing. The exact bearing to the target feature can be determined by rotating the sonar pair so as to null the time difference signal and simply reading off the resulting aim direction. If the feature moves, or more commonly in mobile robotics, the platform on which the sonar is mounted moves, the same signal can be used to maintain track on the target feature as we shall show.

The use of two antennae measuring the same returned signal from two slightly different locations in this way, is known as *monopulse*. This principle is used extensively in tracking radar systems [126][19] and the difference signal measured is commonly called the monopulse signal. In radar the difference signal is either constructed from a comparison of the returned signal amplitudes (called amplitude-mono-pulsing) or from a comparison of the returned signal phase (called phase-mono-pulsing) [19]. A similar approach can be used with sonar assuming that we have devices which are able to measure amplitude and phase. This is not possible with the standard Polaroid device but can be implemented using modified sonar devices as shown by Barshan and Kuc [15].

While not true phase-monopulse in the strict sense, the device that we describe is closely related to the phase-monopulse technique in providing a measurement of the difference in time of arrival of a common signal at two spaced receivers. For this reason we refer to the sonar device as an RCD-monopulse sensor or more accurately *Differential Sonar*. Figure

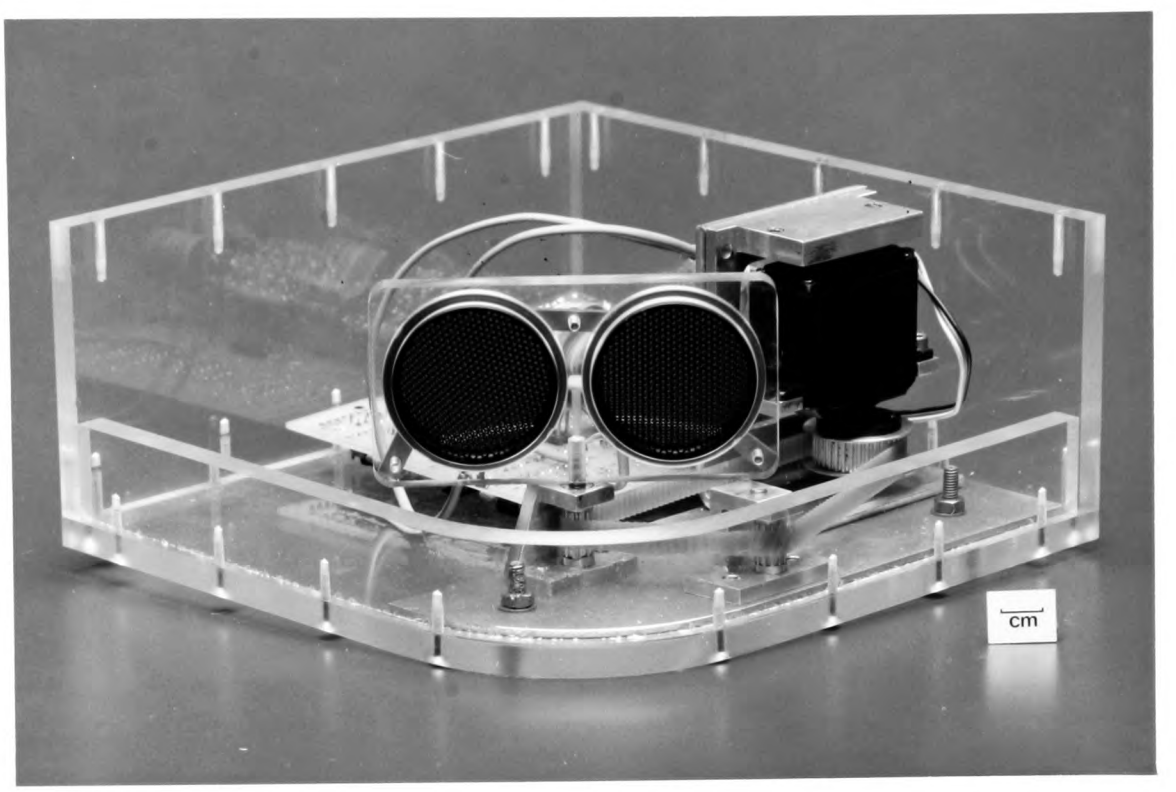
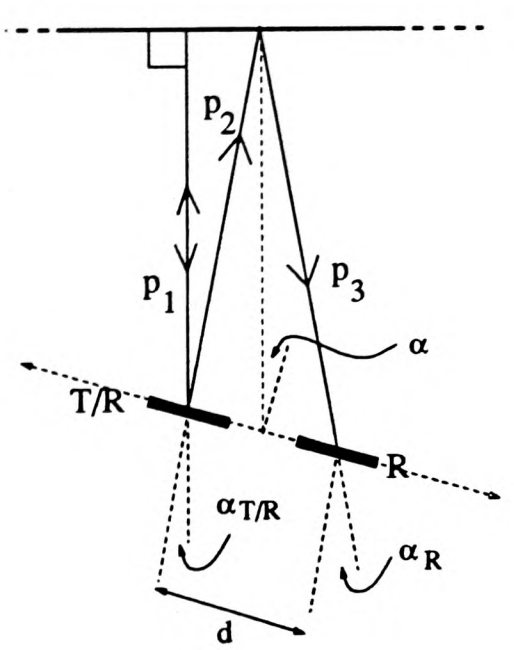


Figure 5.3: Prototype RCD-Monopulse unit.



path length for T/R = $2p_1$

path length for R = (p_2+p_3)

from geometrical considerations

$$2p_1 < (p_2+p_3)$$

since TOF = f(path length)

$$\text{TOF}\ 2p_1 < \text{TOF}\ (p_2+p_3)$$

$$\text{differential} = \text{TOF}\ 2p_1 - \text{TOF}\ (p_2+p_3)$$

Figure 5.4: Illustrating the differential principle for a positive value of α.

5.4 illustrates the principle. As the angle α is varied, detection of the echo is possible as long as the angles $\alpha_{T/R}$ and α_R do not exceed α_{max} given the threshold and frequency settings of the T/R and R device. The normal *time of flight* (TOF) reading for the range is obtained from the T/R device. A differential Δ, is obtained by measuring the time between the reception on the first device to receive, *primary reception* and the time when the second device receives, *secondary reception.* Logic on the driver board notes the order of reception and the occurrence or non-occurrence of secondary reception. Appendix C describes the hardware.

5.1.3 Differential Feature Model

Given that r is the range to the feature, d the base-line of the T/R and R devices and α the angle of inclination from the perpendicular to the plane of the feature, we can define the differential time of flight in general as follows;

$$\Delta \triangleq \frac{1}{c}\delta(r, d, \alpha), \tag{5.1}$$

where δ is the differential path length for primary and secondary reception for angle α such that $-\alpha_{max} \leq \alpha \leq \alpha_{max}$. We now present the differentials for the commonly encountered features in indoor environments that is planes, corners, edges and cylinders. We only consider edges and corners that are orthogonal, as shown in Figure 5.5. Detailed geometrical derivations can be found in Appendix B.2. From Figure 5.5, we can show the following:

1. **Plane.** The differential TOF for the plane is given by

$$\Delta_p = \frac{1}{c}\left[\sqrt{r^2 + \frac{d^2}{4} - rd\,\sin(\alpha)} + \sqrt{r^2 + \frac{d^2}{4} + rd\,\sin(\alpha)} - 2\left(r - \frac{d}{2}\,\sin(\alpha)\right)\right]. \tag{5.2}$$

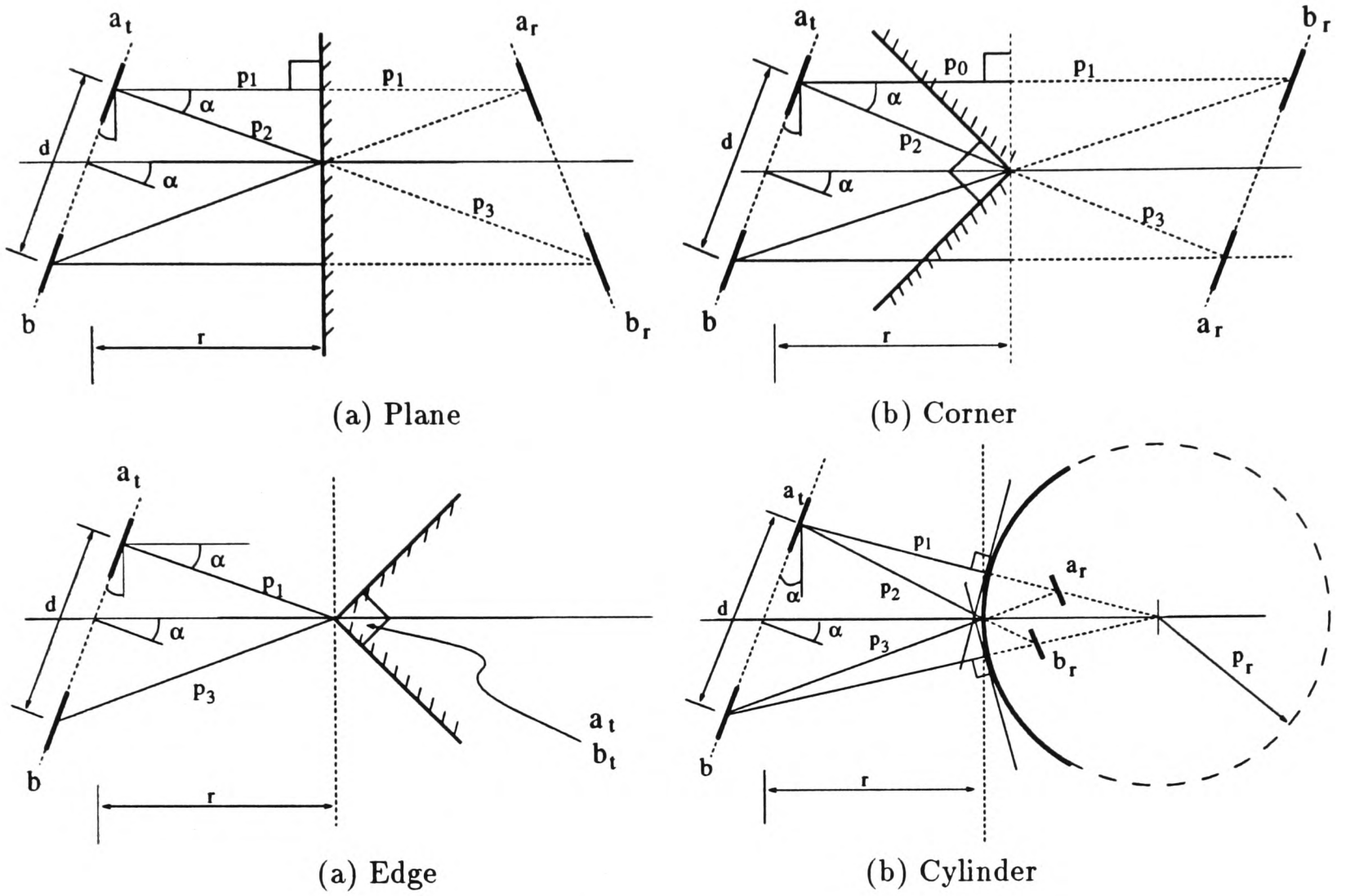


Figure 5.5: Differential sonar model for various features.

2. **Corner.** For a corner the differential TOF is given by

$$\Delta_c = \frac{2}{c}\left[r - \sqrt{r^2 + \frac{d^2}{4} - rd\,\sin(\alpha)}\right]. \tag{5.3}$$

3. **Edge.** The differential TOF for an edge target is given by

$$\Delta_e = \frac{1}{c}\left[\sqrt{r^2 + \frac{d^2}{4} - rd\,\sin(\alpha)} - \sqrt{r^2 + \frac{d^2}{4} + rd\,\sin(\alpha)}\right]. \tag{5.4}$$

4. **Cylinder.** For a cylinder the differential is given by

$$\begin{aligned}\Delta_{cyl} = \frac{1}{c}\Bigg[&\sqrt{r^2 + \frac{d^2}{4} - r\,d\,\sin(\alpha)} + \sqrt{r^2 + \frac{d^2}{4} + r\,d\,\sin(\alpha)} \\ &-2\sqrt{\left(\frac{d}{2}\cos(\alpha)\right)^2 + \left(r - \frac{d}{2}\,sin(\alpha) + p_r\right)^2} - 2p_r\Bigg],\end{aligned} \tag{5.5}$$

where p_r is the radius of the cylinder. The differential of the cylinder can be shown to approximate to that for the plane as the the radius of the cylinder goes to infinity and to that for an edge as the radius goes to zero (See Appendix B.2).

5.1.4 Measured Differentials and Practical Issues

Our present prototype hardware provides a precision of $3.2\mu s/count$ for the measurement of the differential TOF. And we have chosen a base-line of $0.05m$ for the operational range of $0.45 - 3m$. Appendix B.2 discusses the choice of base-line d.

In practice the actual differential obtained is very similar to the one predicted by the theory. Figure 5.6 shows a predicted differential for a plane at a range of $3m$. In Figure 5.7, an actual differential for a plane at a range of $3m$ in a controlled environment, is measured and plotted. Figure 5.8 shows several differentials measured for the four feature types. These are measured in a typical cluttered indoor environment. The deviation from the expected linear characteristic of the differentials can be explained in terms of the irregularities on the surface which cause irregular reflections, the quantization effects from the differential measurement and the imprecision in the positioning by the servo. Differentials for planes and corners are more susceptible to surface irregularities and this is because the received echo at T/R and R is virtually all due to reflection and that due to diffraction is negligible. On the other hand, cylinders (with a small radius compared to the range) and edges generally have better behaved signals because reception is largely due to a diffracted wave and so irregularities on the surface are of less consequence.

In the differentials of Figure 5.8 it would seem as if the null point is offset to the left always. This can be explained by considering the asymmetry in the differential arrangement as follows. The width of the RCD is determined by considering the range returns r which

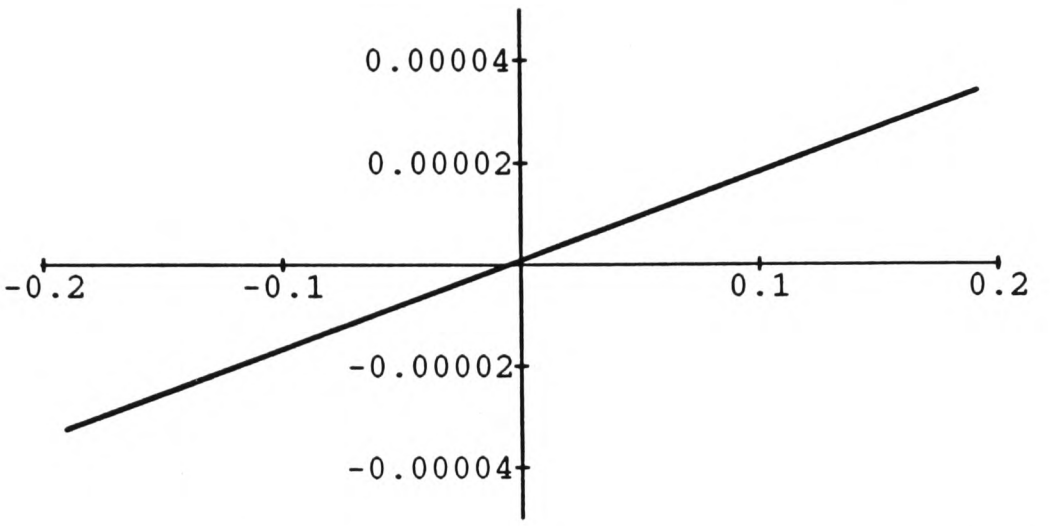


Figure 5.6: Plot showing the predicted differential TOF (in seconds)for a plane RCD. The range, r of the RCD is 3m and base-line, d is 0.06m.

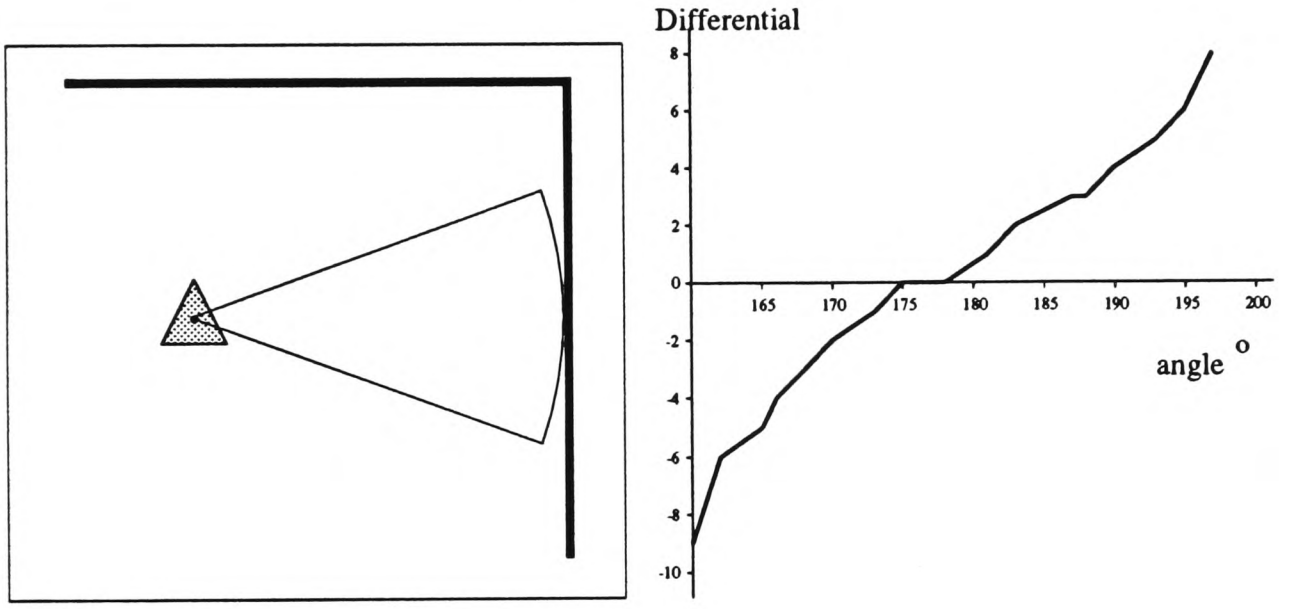
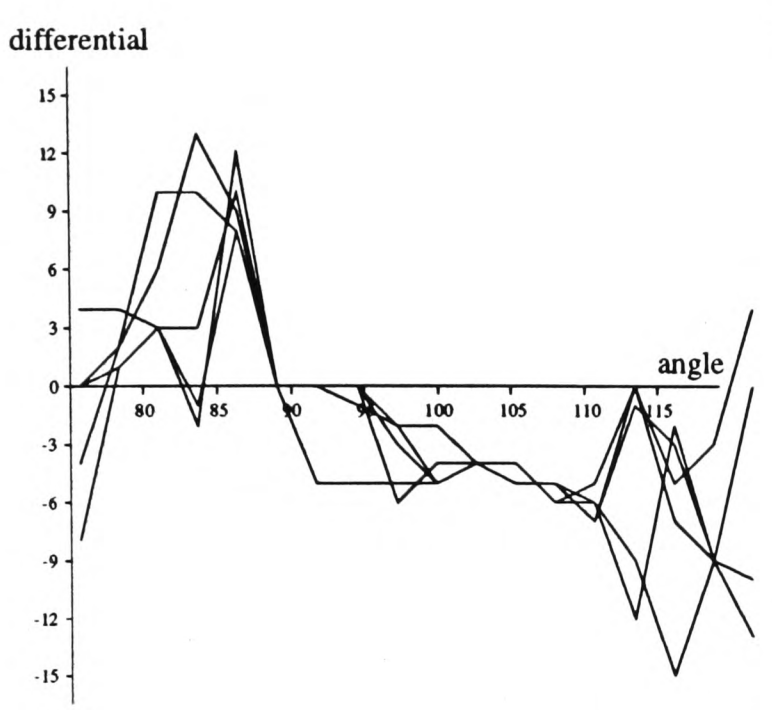
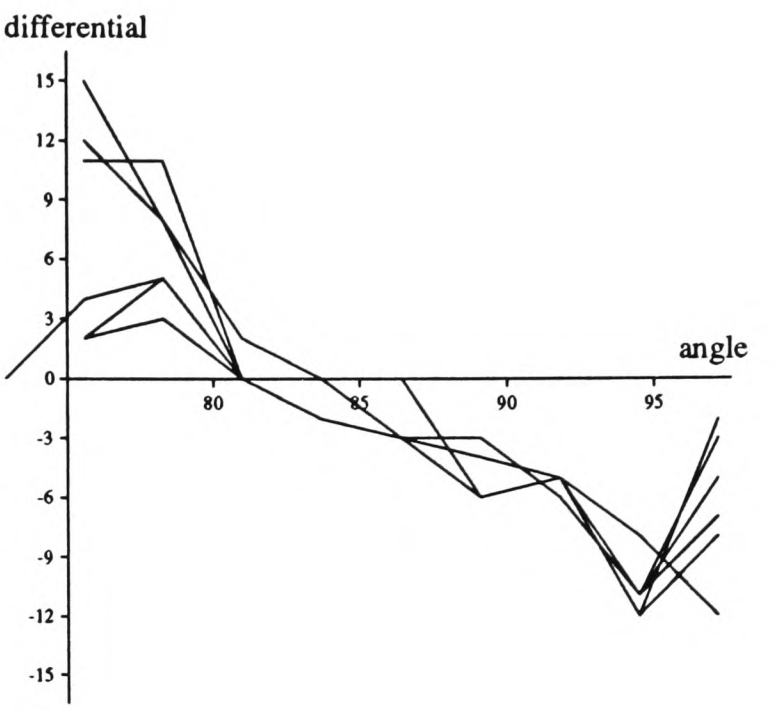


Figure 5.7: An extracted plane RCD and the associated differential. (a) shows the RCD and (b) the differential. The vertical scale in (b) is scaled by $3.2\mu s$ and the horizontal axis is the angular extent of the RCD in degrees.

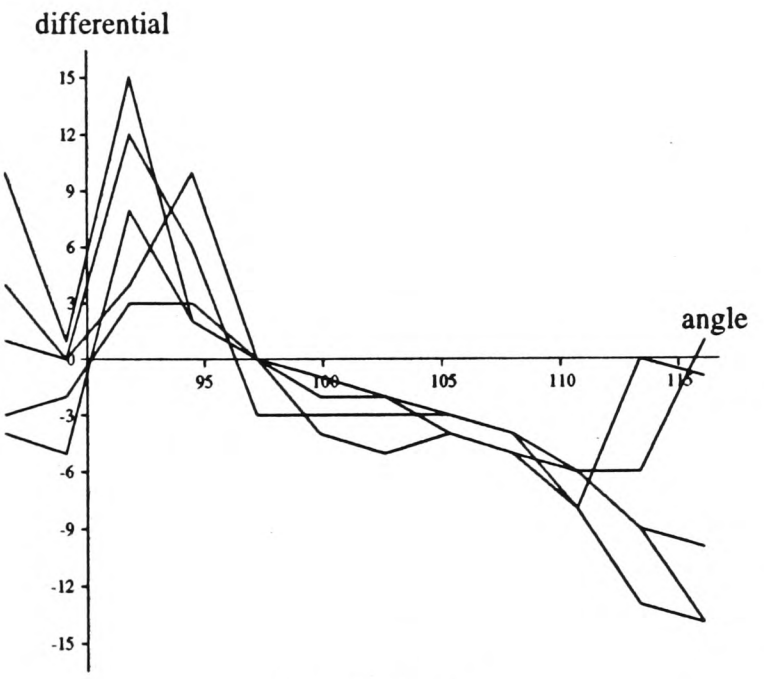
are determined at the T/R transducer. Hence the center of the RCD computed as $(\theta_l+\theta_r)/2$ is not the true center because the RCD is centered at the T/R device which is not at the center of the two Polaroids. An additional consideration is the asymmetry in the Polaroid beam which varies from transducer to transducer [38]. Following on from Figure 5.4 it can be seen that correct alignment of the Polaroid devices is crucial for the precise centering of the differential zero.



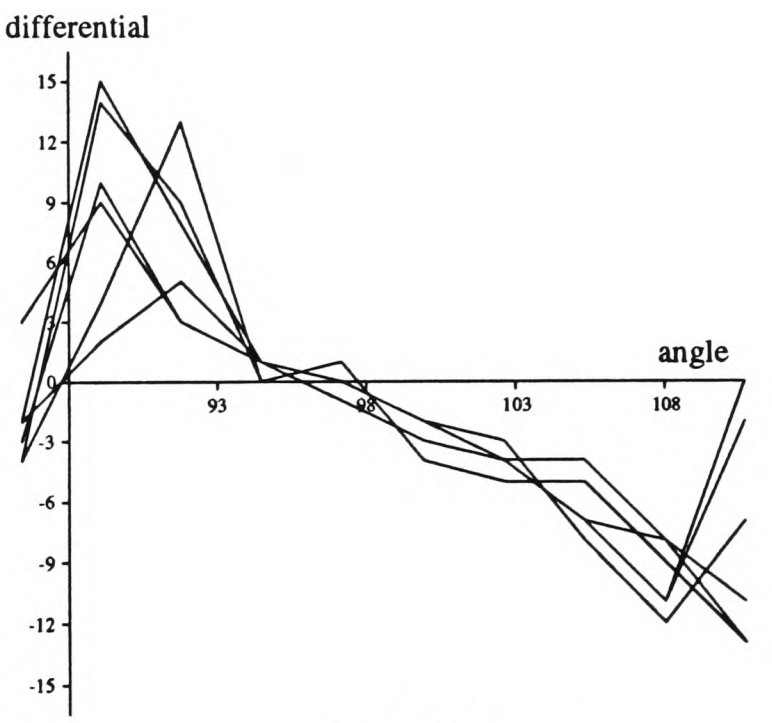
(a) Plane



(b) Corner



(c) Cylinder



(d) Edge

Figure 5.8: Plots showing multiple (5) differentials within the width of the RCD corresponding to each of the four target types. The angle axis is the orientation from some fixed datum of the ranges making up the RCD. Only differential information within the RCD width is of significance. The vertical scale is the differential TOF $\times 3.2\mu s$.

It is important to note that, for our chosen working range and base-line, the predicted differentials for the different target types are almost identical, the actual differences (Discussed in Appendix B.2) going unnoticed by the hardware. Therefore, the differentials of Figure 5.8, serve to highlight the uncertainty associated with sonar measurements in typical application environments.

5.2 Developing A Tracking Sonar Sensor

With each return from the differential sonar we obtain the following data

$$\text{return};[r,\ \theta,\ |\ \Delta\ |,\ sign(\Delta)]. \tag{5.6}$$

Given that the return is within the bounds of an RCD (within θ_l and θ_r), from Figure 5.9 it can be seen that θ is a computable distance ϵ away from the precise location of the target. Simple ratios can be used to compute ϵ, from a consideration of Figure 5.9. The

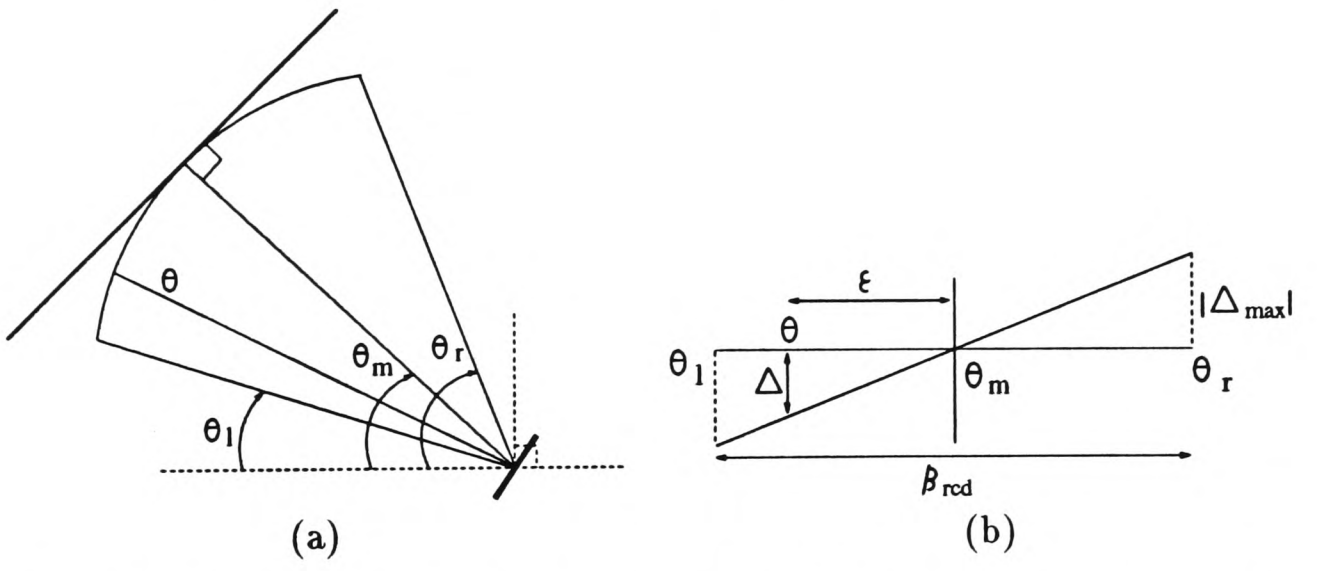


Figure 5.9: Information obtained from a return within the bounds of an RCD.

direction of the return with respect to the true location of the target is given by $sign(\Delta)$, which is obtained from the hardware as a logic value indicating the first Polaroid device to receive. An immediate consequence of this is that with a *single* return at an orientation

θ within the width of the RCD, the precise orientation θ_m of a feature can be computed thus drastically reducing the angular uncertainty. Therefore, differential information can be used to optimize RCD extraction by reducing the number of range returns necessary to establish the RCD in the following way; once one edge of an RCD is detected during a scan, the precise location of the corresponding feature can be predicted without requiring any additional returns. Perhaps more importantly a simple tracking algorithm can be implemented which continually "corrects" for each non-zero differential by steering towards the zero (null) differential. This is the basis of the Tracking Sonar.

5.2.1 Reducing Differential Uncertainty

In typical application environments, it turns out that the measured differential is not as linear as predicted and the results of Figure 5.8 illustrate this. To further highlight the uncertainty in the actual measurement of the differential, it is observed that if the differential sonar is fired repeatedly at the same orientation within the bounds of the RCD the differential measured is not constant as would be expected. This is illustrated in Figure 5.10 where several hundred observations are taken under various environmental conditions (clutter, different feature materials, *et cetera*) and at various offsets within the angular extent of the RCD.

From the differential plots shown in Figures 5.8 and 5.10, it can be seen that some smoothing or filtering operation is required to remove outliers and spurious readings given that we can predict the form of the differential from Section 5.1.3. An approach which is in keeping with the other work we have so far presented, is the use of estimation techniques. With this approach, we treat the measured differentials as noise corrupted observations of the true differentials. We then proceed to estimate the true differential given the noisy

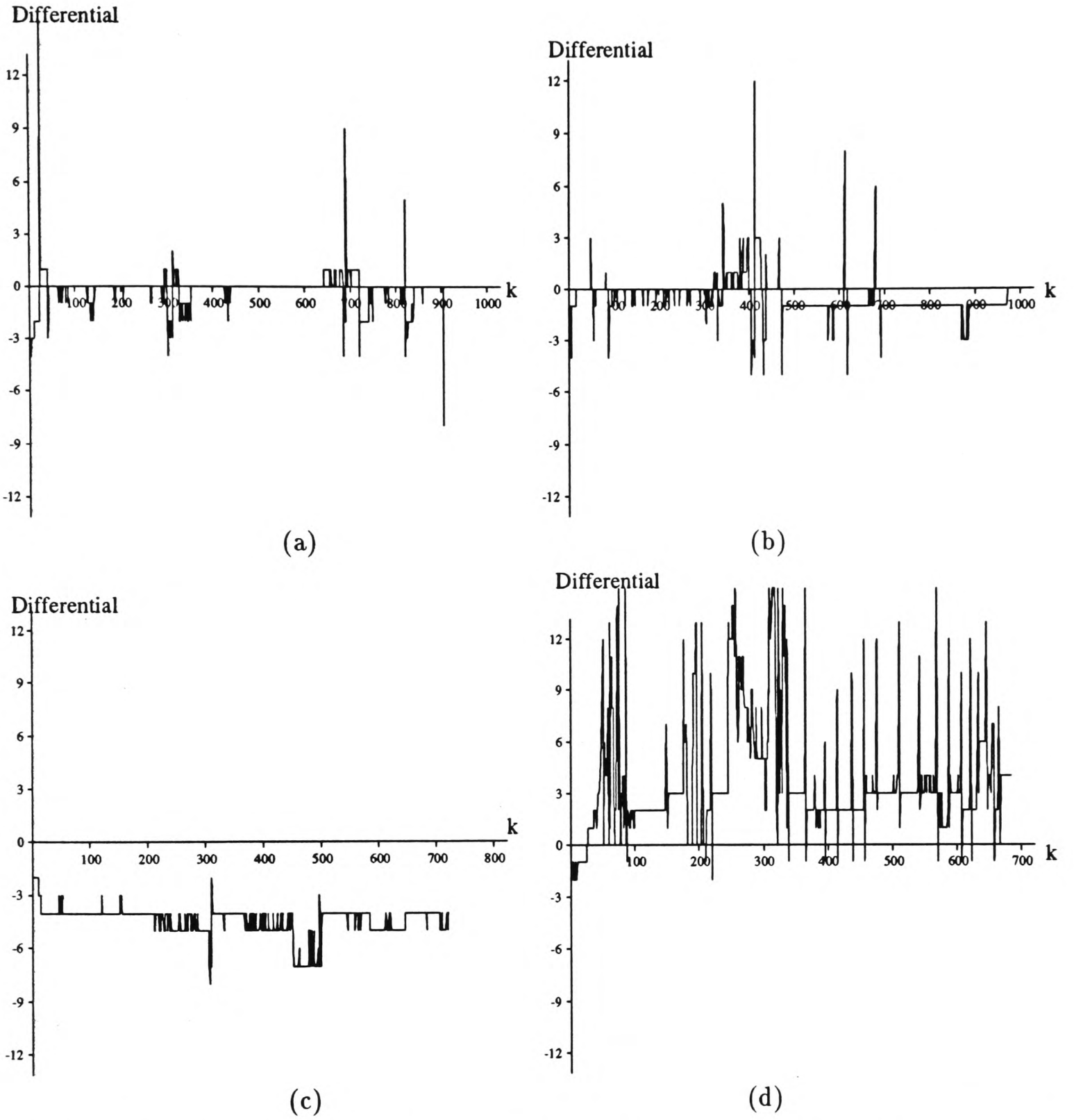


Figure 5.10: The differential at a fixed orientation within the bounds of an RCD highlighting the uncertainty associated with the differential. Theoretically, the differential should remain constant at some value depending on the degree of offset from the center or null position. Although the plots shown are for a plane target similar results are obtained with the other target types. In (a) and (b) there is no offset, in (c) offset is in negative direction and in (d) offset is in the positive direction.

measurements. The Least Squares algorithm [12] estimates an unknown *constant* parameter from uncertain measurements. However, in a real-time tracking situation the parameter being estimated is dynamic since the algorithm is continually correcting for the differential arising from error or relative motion. The correction motion of the servo can be considered as a control input which changes the state of the parameter being estimated. The Kalman Filter [12] (whose information form we have presented in Chapter 3) is well suited to the estimation of time varying parameters.

Using this approach, the state is simply the differential, that is, $x(k) \triangleq \Delta$. Since we can observe the state directly, the observation equation is given by

$$z(k) = x(k) + v(k), \tag{5.7}$$

where the observation noise is assumed to be zero mean and Gaussian

$$v(k) = N(0, \sigma_\delta^2). \tag{5.8}$$

Assuming a stationary feature, the changes in the true differential are only affected by a control input $u(k)$ on the servo. And so we can write the transition equation as follows

$$x(k) = x(k-1) + u(k-1) + w(k), \tag{5.9}$$

where the transition noise is also zero mean Gaussian

$$w(k) = N(0, \sigma_u^2). \tag{5.10}$$

The resulting filter estimate $\hat{x}(k \mid k)$, is the estimated differential $\hat{\Delta}(k \mid k)$.

This simple filter can be improved in several ways: A model of the motion of the feature can be introduced into the state transition Equation 5.9. In the formulation of the filter, it is possible to have a state which includes "velocity" as well as "acceleration". Such a larger

state vector would improve the system's response to the dynamics of the differential in the tracking control loop by making available estimates of velocities and possibly accelerations. This is useful if one is interested in improving the response of the differential estimate to changes in the measured differential and in the control and damping of the servo. We leave the exploration of these ideas for further research.

The differential estimates using this simple filter with no control input $u(k)$, are illustrated in the results of Figure 5.11. The results are obtained when there is no relative motion between the feature and the sensor, hence the measured differentials are similar to those of Figure 5.10. The estimated differential is in agreement with the predicted differential from the theory.

5.2.2 A Real-time Tracking Sonar

Using the differential filter outlined in Section 5.2.1, we can now implement a real-time Tracking Sonar. The control input $u(k)$, has the effect of reducing the differential by steering towards the center of the RCD. We define $s(k)$ as the move step towards the center at time k in correcting for the differential. The move step can be ϵ from our previous discussion however, better performance is obtained by correcting by a smaller step due to hardware limitations. We can obtain the control as

$$u(k) = (-sign(\hat{\Delta}(k \mid k))) \times s(k) \times \mid \Delta_{max} \mid / \beta_{RCD}. \tag{5.11}$$

The tracking algorithm can now be written in full as shown in Figure 5.12. In a simple strategy, r is validated by a comparison with the current RCD range. If the error is acceptable, then the validated r becomes the new RCD range, else the range r is rejected resulting in a termination of the track. This simple approach can be improved upon with

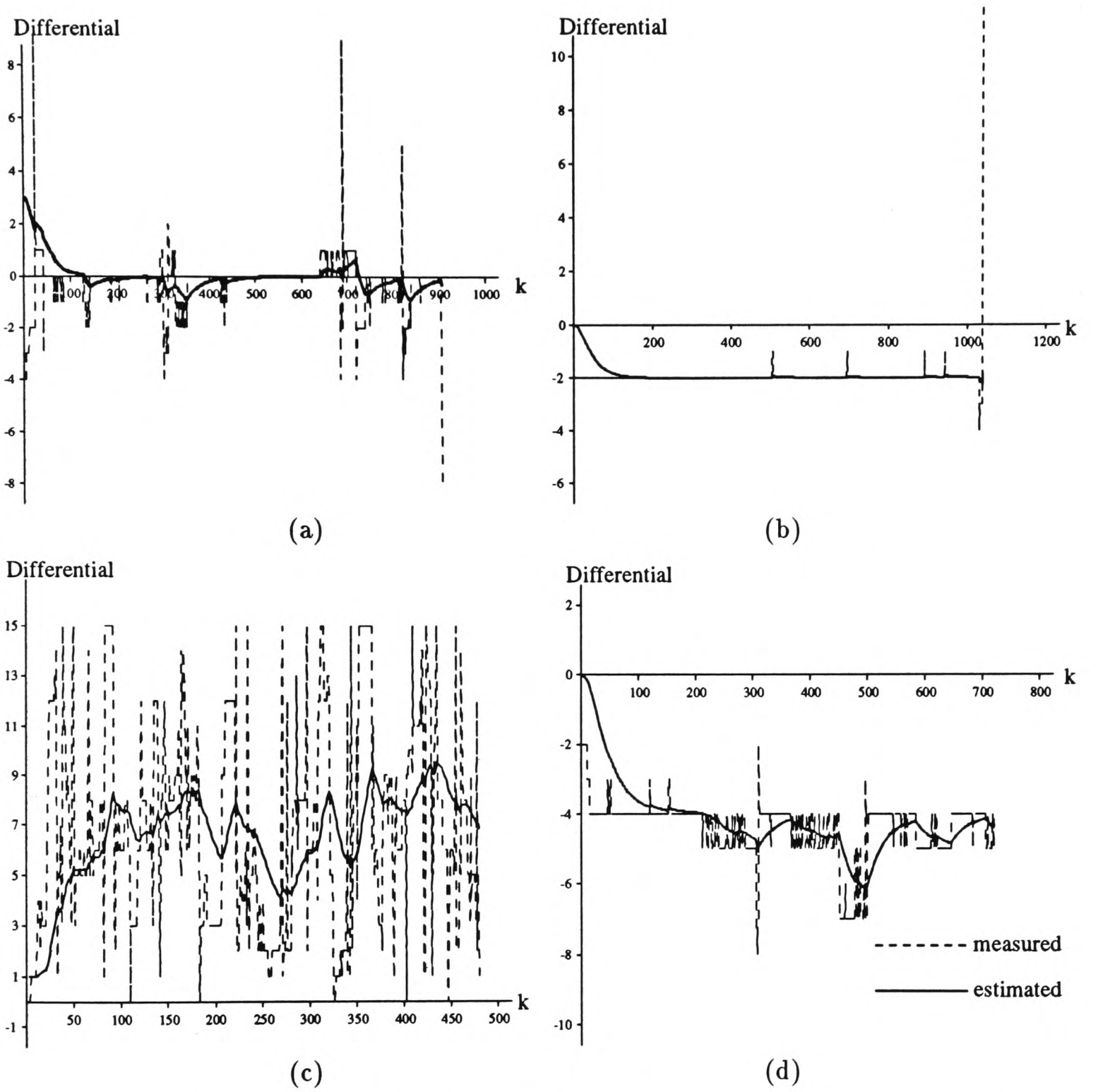


Figure 5.11: The measured and estimated differential at a fixed orientation within the bounds of an RCD. In (a) the truncated predicted differential is 0, that is, no offset, in (b) the truncated predicted differential is -2, in (c) 6 and in (d) -4. All values of the differential being $\times 3.2\mu s$.

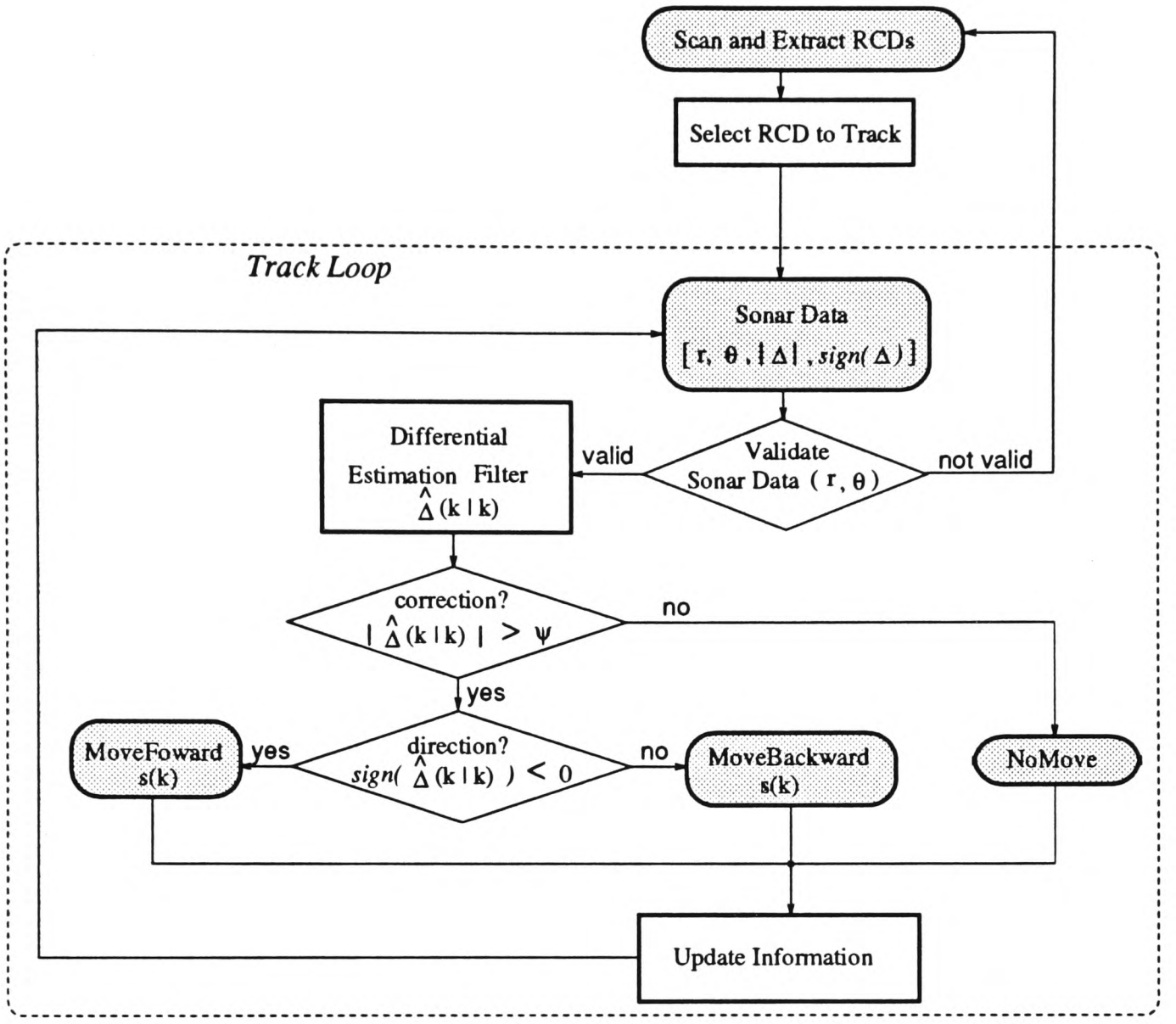


Figure 5.12: Tracking Sonar Algorithm. The shaded areas indicate hardware related procedures. The selection of RCD to track is application dependent and shall be discussed in the next Chapter.

more detailed range and bearing validation incorporating checks for "spurious" measurements *et cetera*. The criteria for RCD selection depends on the application or goals of the system and shall be discussed later. The parameter ψ defines the *dead-band* within which correction for the differential error is deemed unnecessary.

The results in Figure 5.13(a) show the differential (estimated and observed) while tracking a feature which moves at the indicated time-steps. Figure 5.13(b) shows the results of the differential sonar tracking a stationary RCD target with the sonar platform being ro-

tated occasionally at the time steps as indicated. In these results we have set a dead-band ψ equal to 2 differential counts (2 $\times 3.2\mu$ s). This dead-band is set because the smallest step we are able to move with our present prototype hardware corresponds to a differential of the order of 2 ($\times 3.2\mu s$). Consequently, a dead-band much smaller than 2 would result in continual correction regardless of relative motion due to the lack of precision in the positioning of the Polaroids by the servo. In Figure 5.14(a), we show the motion of a feature which moves arbitrarily as observed by the sensor tracking it. In Figure 5.14(b) we have the feature stationary and the sensor platform rotating about its axis and thus we show the relative positions of the feature as observed by the sensor in ego-centric coordinates. In all these results, the sonar data rates are approximately $30Hz$.

5.3 Probabilistic Model and Error Analysis

The Tracking Sonar provides range r and orientation θ data, where the orientation is referenced to a *datum* line on the sensor itself. With respect to our discussion in Chapter 2, the sensor provides a 2-dimensional observation vector $\mathbf{z}$ given by

$$\mathbf{z} = [r, \theta]^T . \tag{5.12}$$

The range and orientation incorporates some uncertainty. We now develop a probabilistic model for this uncertainty. We assume the uncertainty in the measurements is in the form of additive noise. Therefore, we can write that

$$\mathbf{z} = \mathbf{z}_{true} + \mathbf{v}(\zeta, \mathbf{z}_{true}), \tag{5.13}$$

where $\mathbf{z}_{true}$ is the true state being measured and $\mathbf{v}$ the observation noise which is dependent on the true value and on ζ the sensor control parameters (differential filter parameters, dead-band *et cetera*). It will be noted that Equation 5.13 has been written in terms of the

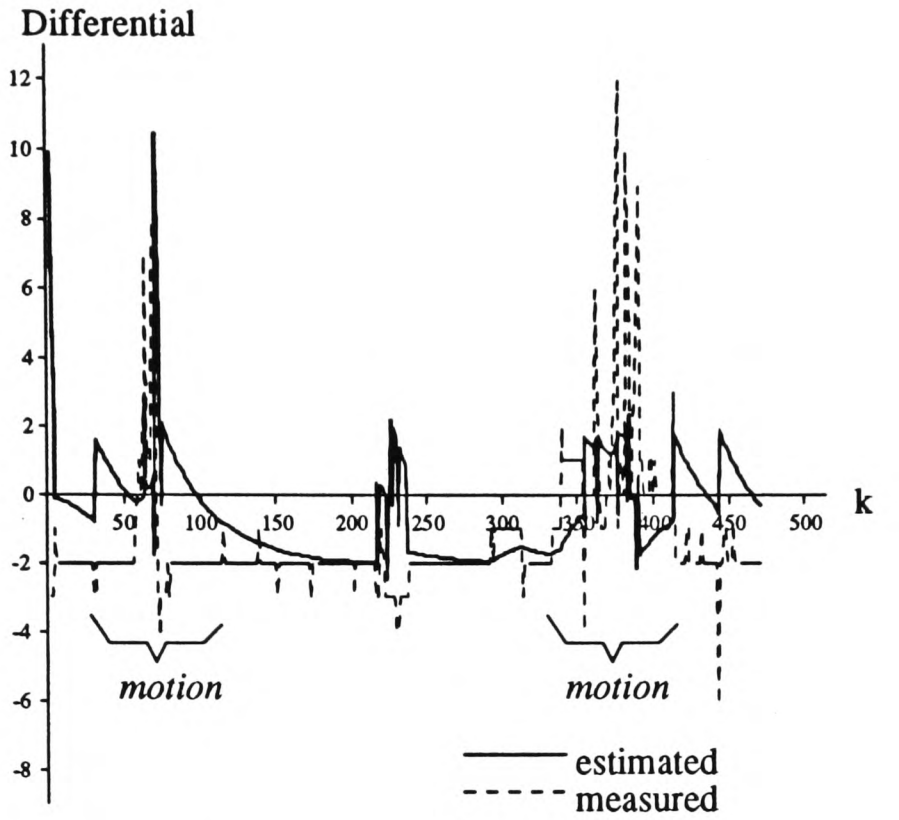
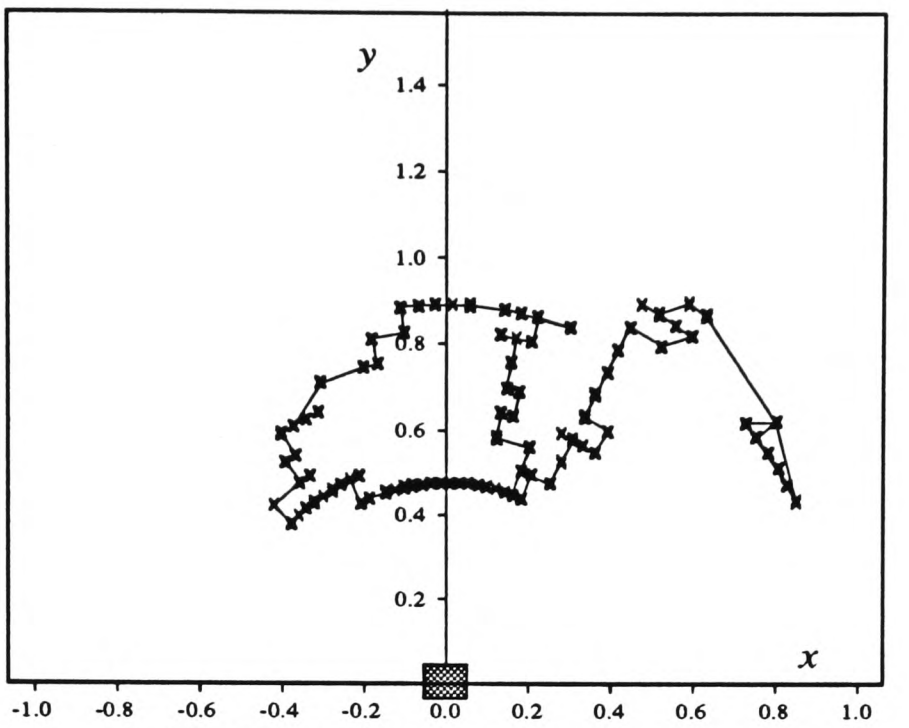
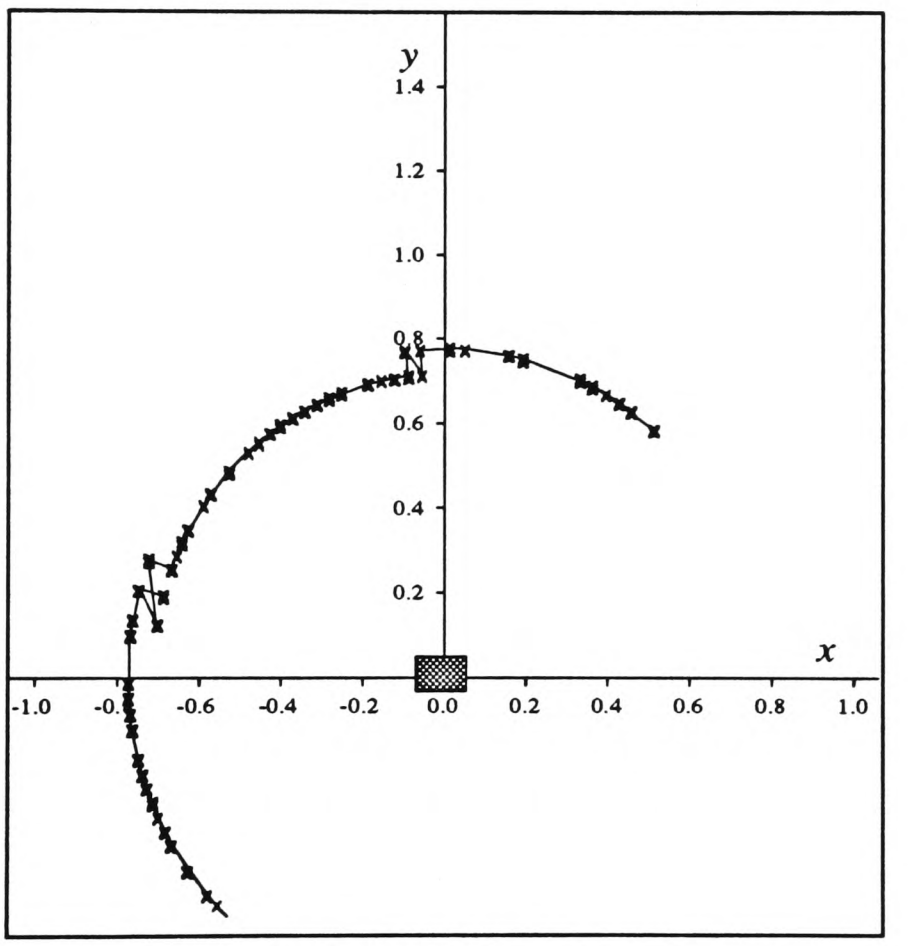


Figure 5.13: Results showing the measured and estimated differentials in the tracking of an RCD target. In (a), the differential sonar platform is stationary ABA the target is in motion from step 50 - 75 and from 350 - 390. In (b), the differential platform is rotating with the RCD target stationary. Platform rotates from step 1000 - 1200. In both (a) and (b) there is a no-correction differential dead-band of 2 ($\times 3.2\ \mu s$).



(a) Arbitrary Feature Motion



(b) Sensor Rotational Motion

Figure 5.14: Illustrating motion of a tracked feature in sensor ego-centric coordinates as calculated from the range and orientation as a result of focussing attention on the target. In (a) the feature moves arbitrarily whilst the Tracking Sonar unit is stationary. In (b) the feature being tracked is stationary and the Tracking Sonar is rotating about a fixed point.

measured values themselves rather than a state $\mathbf{x}$ to avoid the need to state a model relating $\mathcal{X}$ and $\mathcal{Z}$, this being application dependent. Here we take the same approach as Hager and Mintz [67] and assume that $\mathbf{v} \in \mathcal{V}$, where $\mathcal{V}$ is a class of random variables. It is therefore desirable to develop a model of the additive noise $\mathbf{v}$ for a given range of sensor control parameters. It is usually assumed that the noise $\mathbf{v}$ is Gaussian, indeed the information filter derived in Chapter 3 makes such an assumption. It is now our intention to justify Equation 5.13 and determine the statistical model of the disturbance $\mathbf{v}$.

We commence by considering the factors determining the performance of the Tracking Sonar and the uncertainty in its observations. These are as follows:

- **Differential estimator parameters.** A large value of differential observation noise σ_δ^2 (Equation 5.8) is desirable when tracking features whose motion relative to the sensor is slow and deliberate. However when motion is sudden and quick, what may appear as spikes in the differential may actually represent the sudden motion. Hence in such situations, an observation noise model which assumes relatively little noise in the measurements results in better performance as it gives more credence to the actual observations themselves.

- **Differential dead-band.** The dead-band is a consequence of hardware limitations, in particular limitations in positional accuracy and motion step size of the servo. Another consideration resulting in the dead-band is the electrical noise which results when the correction motion rate of the servo is high. This electrical noise causes jitter in the positioning of the servo thus giving unreliable differential measurements. This can be alleviated with reduced correction motion rates hence the dead-band.

A model should incorporate how the statistics of the information obtained from the sen-

sor vary with these factors. In the following section we analyse the data using spectral estimation techniques.

5.3.1 Use of Spectral Estimation Techniques

In modelling the data from the Tracking Sonar, the range r and orientation θ over time can be treated as random processes. If there is no relative motion between the feature being tracked and the sensor, then the random processes can be viewed as stationary[4]. When there is relative motion between the feature and the sensor, the processes then cease to be stationary. In modelling the information from the sensor we shall only consider the stationary case, that is, when there is no motion between the sensor and the feature being tracked [5]. Our interest in the proceeding analysis only lies in determining whether the process noise is white or not.

Random processes are defined in terms of their ensemble averages which can be estimated [109]. Our model shall be in terms of such averages. In practice, we require to estimate these averages from finite sequences. Therefore, we consider a process y_k as realized by the finite sequence $y(k)$, for $0 \leq k \leq N-1$ as an estimate of the random process $\mathbf{y}_k$ and this being made plausible by a consideration of ergodic processes [109]. Correspondingly, we can estimate the averages as follows: The mean is estimated by

$$\hat{\mu}_y = \frac{1}{N} \sum_{k=0}^{N-1} y(k), \tag{5.14}$$

[4]This is not strictly true, because the differential used to maintain track, is essentially the result of a range measurement. Range measurements even with no relative motion can only be treated as stationary over very long periods since the measured range depends on the temperature of the intervening air which is usually treated as a random process having low frequency components(of the order of seconds). The end results being that closely timed range samples in the short interval, are not really stationary. However, the error due to this is small enough to be ignored in our case.

[5]The case when there is motion requires that we consider the residuals of the process given an adequate process model, and attempt to describe these in terms of their averages by considering the residuals as a stationary process [79].

and the variance by

$$\hat{\sigma}_y^2 = \frac{1}{N}\sum_{k=0}^{N-1}(y(k) - \hat{\mu}_y)^2. \tag{5.15}$$

In a similar way, the *unbiased* autocorrelation is estimated by

$$\hat{\phi}_{\mathbf{yy}}(m) = \frac{1}{N - \mid m \mid}\sum_{k=0}^{N-|m|-1} y(k)\cdot y(k+m), \tag{5.16}$$

where $\mid m \mid < N$. A *biased* estimate of the autocorrelation is given by

$$\tilde{\phi}_{\mathbf{yy}}(m) = \frac{1}{N}\sum_{k=0}^{N-|m|-1} y(k)\cdot y(k+m). \tag{5.17}$$

The variance of the unbiased autocorrelation is given by

$$\hat{\sigma}_{\hat{\phi}} = \frac{\sqrt{N}}{N - \mid m \mid}\hat{\sigma}_y^2, \tag{5.18}$$

and that for the biased autocorrelation by

$$\tilde{\sigma}_{\tilde{\phi}} = \frac{1}{\sqrt{N}}\hat{\sigma}_y^2. \tag{5.19}$$

The power spectrum can be estimated using

$$P_N(\omega) = \sum_{m=-(N-1)}^{N-1} \hat{\phi}_{yy}(m)e^{-j\omega m}. \tag{5.20}$$

Some form of correction in the autocorrelation estimate is required for the low frequency trends due to the finite nature of the sequence used for estimation[6]. For this and other practical reasons [79][109] the estimated autocorrelation should realistically be used only as a guide to modelling or describing random processes.

For a truly zero-mean white process, the autocorrelation is large when $m = 0$, is ideally zero elsewhere. The autocorrelation at $m = 0$ gives the variance of the process. For a finite sequence, basic tests for whiteness are as follows [79]:

[6]This can be done in the simplest cases by subtracting a constant term or by a differencing or filtering operation. The finite nature of N results in less damping in the autocorrelation than would be expected from theoretical considerations. Hence Jenkins and Watts [79] draw the conclusion that "..it is sometimes dangerous to read too much into the visual appearance of the acf [autocorrelation function] especially from a short series.".

- For a white process the autocorrelation can be shown to be normal with a zero mean and a variance $1/N$, provided that N is large enough. Hence any process whose autocorrelation approximates to this can be considered as white. The degree of closeness to the true white process can be determined by considering statistical regions of confidence. For 99% confidence, the bounded region for whiteness is $\pm 2\sigma$ and for 95% the region is $\pm 3\sigma$. Oppenheim and Schafer [109] go as far as saying that as long as the normalized autocorrelation for $1 \leq m \leq 512$ is "much less" than the value at $m = 0$, then the random sequence can be considered as uncorrelated.

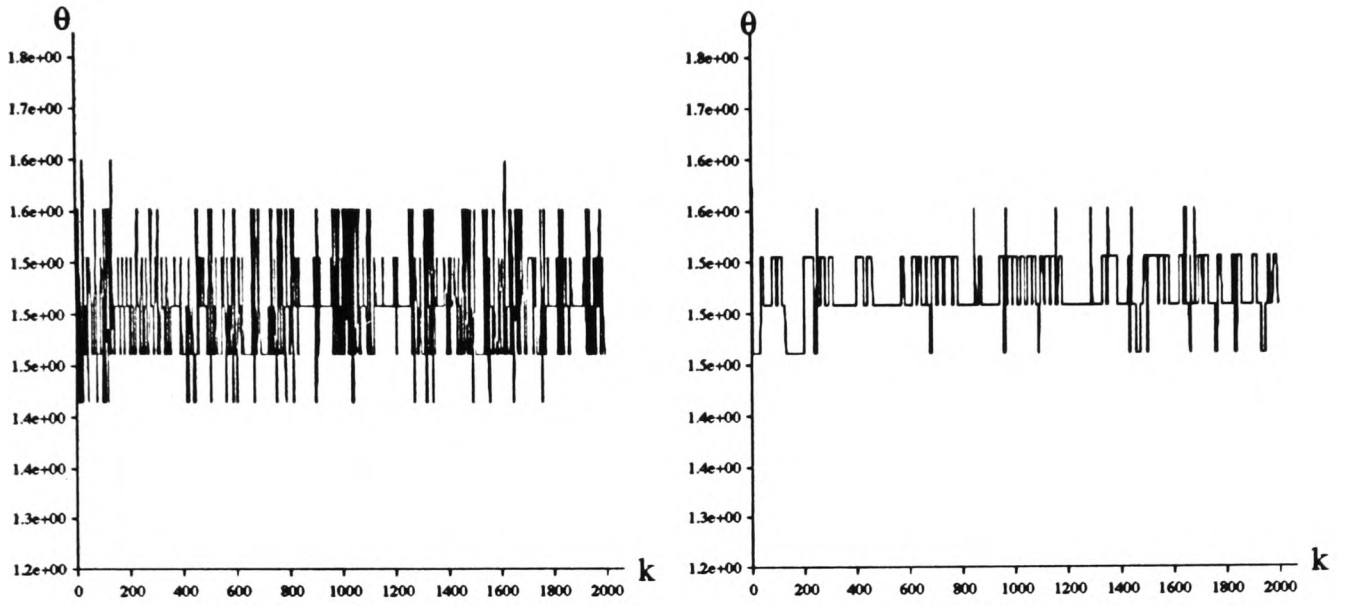
- In addition to the autocorrelation, the power spectra can be considered by taking the discrete Fourier transformation of the biased autocorrelation estimate. For a perfectly white sequence the power spectrum is minimally flat, that is, equally distributed over all frequencies. Thus for a given power spectrum the degree of closeness to a flat spectrum can be used as a test for whiteness.

More sophisticated tests, based on Kolmogoroff-Smirnov [79] confidence regions can be done to complement these basic tests as described by Barshan for a gyro-scope in [13].

5.3.2 Analysing Measurement Uncertainty

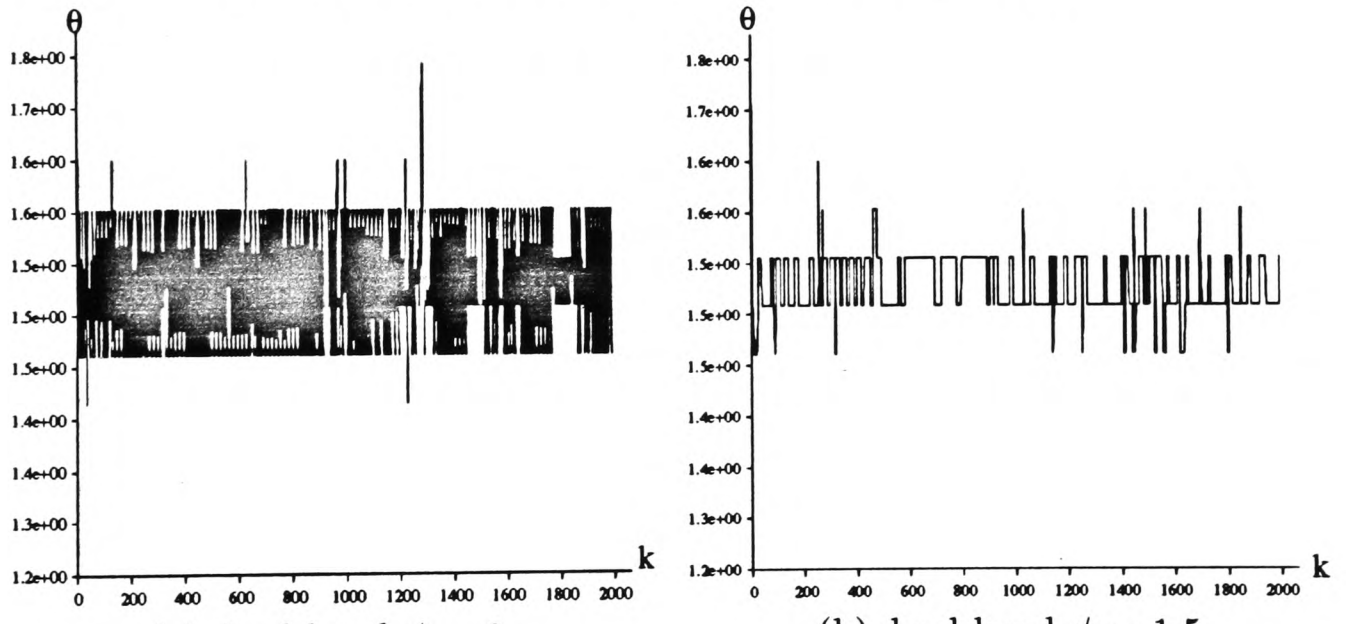
By considering the tracking of a stationary feature, a probabilistic model of the data can be developed by estimating and analysing the process averages as described in the previous section using finite sequences of the data. We consider the averages within ranges of operating parameters which give satisfactory performance of the Tracking Sonar.

We commence by considering the mean and variance of the range and orientation data resulting from the tracking and the variation of these with respect to the tracking algorithm parameters. Figure 5.15 shows variations of orientation θ with observation noise σ_δ^2 for the



(a) $\sigma_\delta^2 = 0.00001$ (b) $\sigma_\delta^2 = 0.001$

Figure 5.15: Plots showing the variation in bearing θ over time with no relative motion between the feature and the sensor for various differential estimator noise models σ_δ^2. With $\sigma_\delta^2 = 0.0000001$ (not shown) the feature was lost after about 500 steps. Variance of the process noise in the differential estimation filter is fixed at $\sigma_u^2 = 0.002$.



(a) dead-band $\psi = 0$ (b) dead-band $\psi = 1.5$

Figure 5.16: Plots of orientation (in radians) over time for various dead-band settings for a stationary feature. The differential estimation filter noise models are fixed at $\sigma_u^2 = 0.002$ and $\sigma_\delta^2 = 0.001$.

differential. The results show that for a stationary target the variance in the orientation of the target is reduced with a larger observation noise variance σ_δ^2. This is further illustrated in Table 5.1 for a fixed range and observation. The table shows values of the mean and variance of the range and orientation to the feature when there is no relative motion between the feature and the sensor. The results of Table 5.1 show that increasing the noise model eventually results in some instability. A compromise has to be reached between increasing the noise model and having a small enough model to give a good response to motion. And so the value used for the noise models depends very much on the application and anticipated motion. In practice with our particular hardware, it has been found that for a process noise

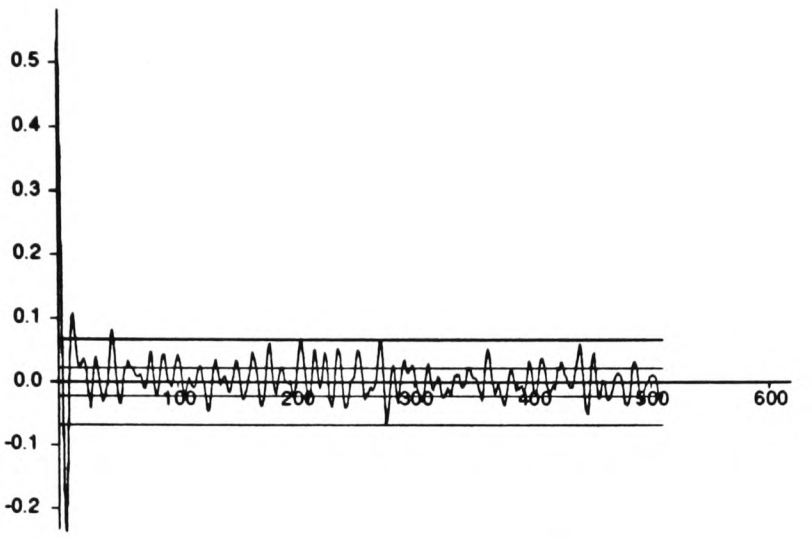
Δ noise variance σ_δ^2	orientation mean $\bar{\theta}$	θ variance σ_θ^2	range mean $\bar{r}$	range variance σ_r^2
0.0000001	1.4943	0.0130	1.0141	0.0061
0.00001	1.5080	0.0033	1.0135	0.0028
0.0001	1.5055	0.0027	1.0125	0.0011
0.001	1.5215	0.0020	1.0115	0.0005
0.01	1.5362	0.0018	1.0115	0.0005
0.1	1.5134	0.0015	1.0115	0.0005
0.5	1.5216	0.0017	1.0115	0.0005
1.0	1.5171	0.0019	1.0115	0.0005
5.0	1.5037	0.0025	1.0115	0.0005
50.0	1.5131	0.0031	1.0165	0.0080

Table 5.1: Table showing the mean and variance of the range and orientation of the target with varying noise models at a fixed range and orientation. Value of the process noise fixed at $\sigma_u^2 = 0.002$.

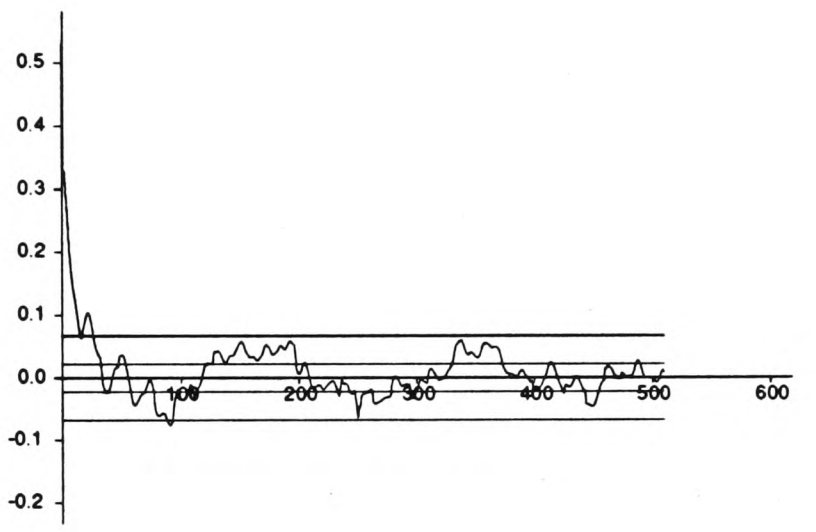
(Equation 5.10) fixed at $\sigma_u^2 = 0.002$, observation noise models σ_δ^2 in the range 0.00001 - 0.1 give satisfactory fixating performance. The specific value in this range being determined by the nature of the features and the relative motion involved. Thus the noise model is inversely proportional to the maximum angular speed of relative motion. Analysis similar to

that in Table 5.1 shows that with everything else the same, the variances of r and θ increase with range. In addition, these variances also vary with the orientation angle θ in that given that the field of view of the sensor is limited, the variance of the orientation increases at the edges of the angular field of view making the variance dependent on the observation itself. Appendix B.2 gives functions which can be used to model this dependence of the variances on range and orientation. Figure 5.16 shows the variation of the orientation with the dead-band for a stationary feature and stationary sensor. In practice it is found that a dead-band ψ in the range 0.5 - 2.0 gives satisfactory performance with the specific value being determined by the relative motion expected.

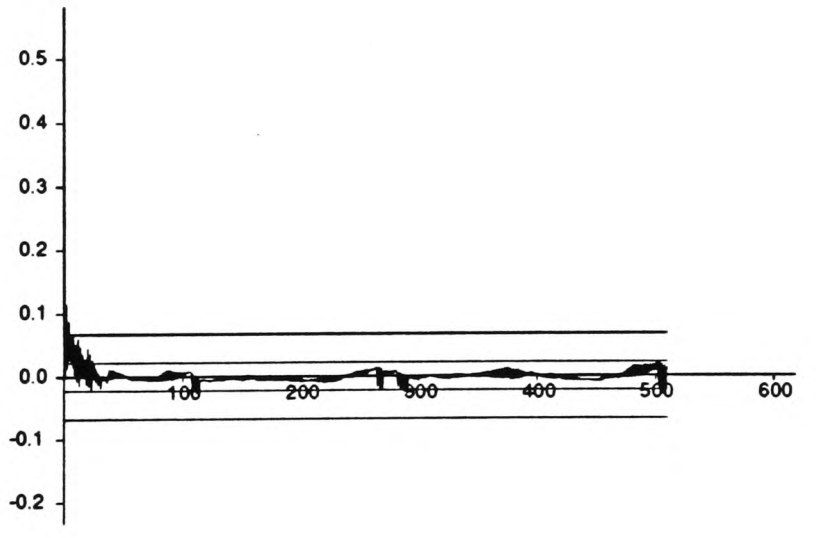
For the ranges of the observation noise model σ_δ^2 and dead-band ψ which give satisfactory fixating performance the data stream can be modelled as a true value corrupted by white noise. This can be justified by considering the autocorrelation and power spectra of the data streams under various conditions. We start by considering the extreme values in our σ_δ^2 which give satisfactory performance. For these values we plot the estimated biased autocorrelation functions (Equation 5.17) for the orientation data. From Figure 5.17 using the confidence tests described by Jenkins and Watts [79], it can be seen that the autocorrelation function is within the $\pm 3\sigma^2$ (for 95% confidence) region. Based on this together with the discussion by Oppenheim and Schafer [109] on the relative levels of the normalised autocorrelation function, we can model the orientation data as being corrupted by white noise. We gain further confidence by considering the corresponding power spectra. Figure 5.18 shows the corresponding power spectra for the autocorrelation functions shown in Figure 5.17. These are approximately minimally flat as would be expected for a white sequence. However, it can be appreciated from Figure 5.18 that as σ_δ^2 is increased, the modelling of the process as white becomes crude. However bearing in mind the caveat by Jenkins and Watts



(a) $\sigma_\delta^2 = 0.00001$

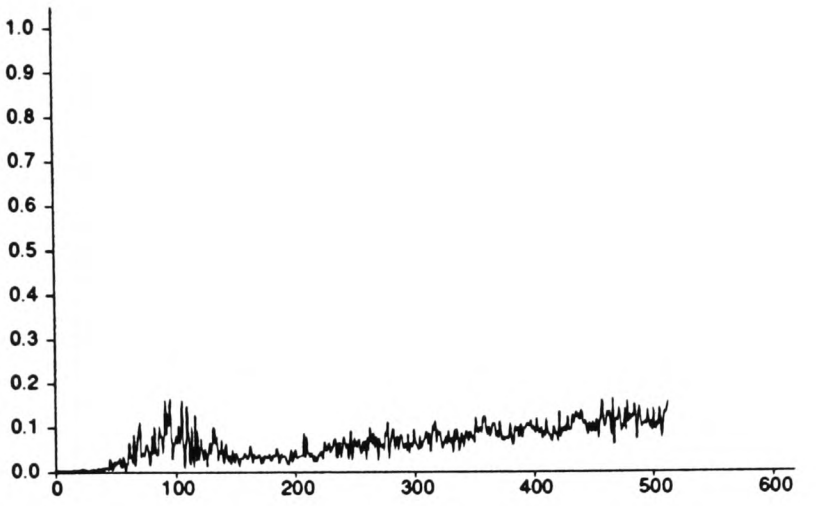


(b) $\sigma_\delta^2 = 0.001$

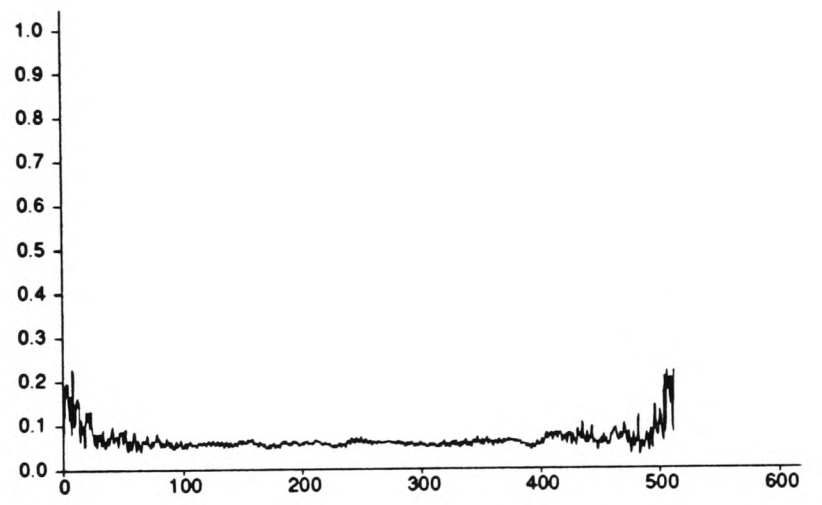


(c) $\sigma_\delta^2 = 0.1$

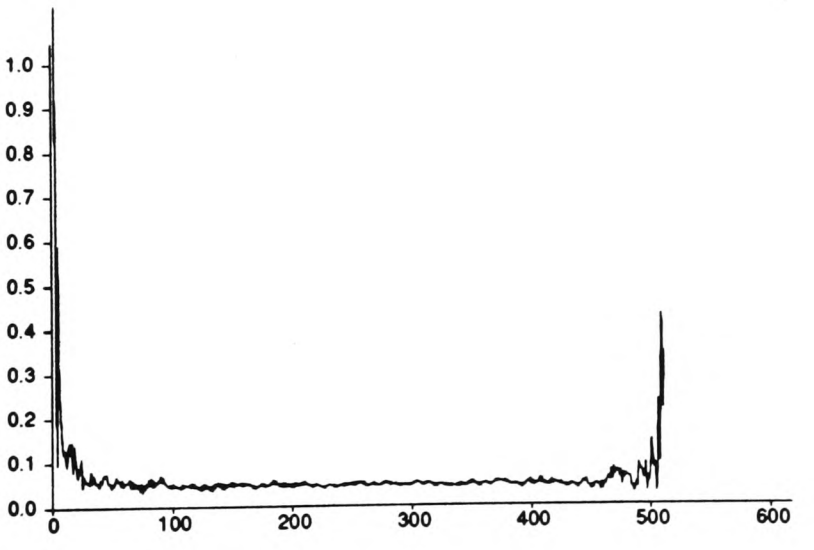
Figure 5.17: Autocorrelation estimates for the orientation during tracking of a stationary feature. The plots are for various values of the observation noise model within the range giving satisfactory fixating performance. The autocorrelation is normalized, that is, $\hat{\phi}_{yy}(0) = 1$. The plots also show $\pm 3\sigma^2$ and $\pm\sigma^2$ corresponding to whiteness confidences of 95% and 99% respectively.



(a) $\sigma_\delta^2 = 0.00001$



(b) $\sigma_\delta^2 = 0.001$



(c) $\sigma_\delta^2 = 0.1$

Figure 5.18: Power spectra using a Bartlett window $m = 512$ for the orientation during tracking for a stationary feature. The plots are for various observation noise models within the range giving satisfactory fixating performance.

concerning the need for correction at low frequencies in the autocorrelation, modelling the data stream as white is justified.

The range data can be shown to exhibit similar behaviour to the orientation data, the difference being that the variance is approximately an order of magnitude less. This is as a result of the range validation on which continued fixation is dependent. This modelling of the data from the sensor has also been shown to hold for variations in the dead-band which show similar trends in the autocorrelation and power spectra.

5.3.3 Sensor Probabilistic Model

From the preceding analysis, the information supplied by the sensor can be modelled as having a mean about the true value with uncertainty due to additive white noise whose variance depends on the measured quantities themselves and the operational parameters of the sensor. Thus we can write the components of the observation vector of Equation 5.13, that is, $\mathbf{z} = [r, \theta]^T$ as

$$\begin{aligned} r &= r_{true} + v_r(\zeta, r_{true}) \\ \text{and} \quad \theta &= \theta_{true} + v_{\theta,}(\zeta, \theta_{true}), \end{aligned} \tag{5.21}$$

where the noise written as $v_r(\zeta, r_{true})$ signifies the dependence on the sensor control parameters ζ (differential filter parameters, dead-band) and the true value itself r_{true} and similarly for $v_\theta(\zeta, \theta_{true})$. Therefore, the noise vector $\mathbf{v}$ has a 2-by-2 covariance matrix $\mathbf{R}$ composed of the variances σ_r^2 and σ_θ^2 along the diagonal. The values of these variances can be determined for particular operational parameters as shown in the results presented (See Table 5.1). We assume that noise associated with the range measurement v_r is uncorrelated with the noise associated with the orientation v_θ. This assumption is in the absence of a modelling of the possible cross-correlations between v_r and v_θ at the same timestep.

This is particularly useful since we can determine for a given scenario, the statistics of the data obtained and consequently take cognizance of the uncertainty quantitatively when making use of the information from the sensor. The noise covariance at the time step k (using the notation of Chapter 3) is written

$$\mathbf{R}[k, \mathbf{z}(k)] = \begin{bmatrix} \sigma_r^2(\zeta) \times g_r(r) & 0 \\ 0 & \sigma_\theta^2(\zeta) \times g_\theta(\theta) \end{bmatrix}. \tag{5.22}$$

The functions $g_r(r)$ and $g_\theta(\theta)$ model the dependence of the variances on the measurements themselves. Details of the functions g_r and g_θ can be found in Appendix B.2. Such an observation dependent covariance $\mathbf{R}[k, \mathbf{z}(k)]$ is sometimes known as a *location covariance.*

5.3.4 Performance and Limitations

As a complete sensor, the differential sonar, comprising the Polaroids, driver electronics and servo can be operated in several modes. Its use in extracting RCDs is similar to the work which has already been described in the literature [88][75]. In this area, the differential sonar offers the improvement that the RCD extraction algorithm can be optimized with the availability of centering information from the differential as discussed in Section 5.2. It is the use of the differential sonar in its novel tracking or fixating mode which is innovative and has interesting implications to the other work presented in this thesis. It is the performance in this latter mode which we now discuss in terms of the following:

- **Speed of relative motion.** The limited sonar return rate coupled with the fact that the servo requires a finite time to effect correctional motion, places a limit on the data rate of the sensor. The maximum data rate of the sensor is approximately $30Hz$. Relative motion with regard to this sensor, can be decomposed into *angular* motion and *linear* motion along the normal to the feature being tracked. This particular decomposition of the relative motion is as a result of the fact that the tracking is

effected by correcting for the angular displacement and updating the corresponding range to the feature as shown in the algorithm of Figure 5.12. Assuming the worst case scenario that every return requires correctional-motion, the angular speed of the relative motion is limited to a maximum of approximately $0.6rad/s$ while tracking an RCD of width $0.17rad$ using the simple algorithm of Figure 5.12. However, this maximum speed can be increased to about $0.9rad/s$ through the use of filter parameters, dead-band settings and a variable size of the correctional move step, optimized for a particular environment.

Of secondary importance is the linear speed along the normal to the target. In addition to the firing speed, the limiting factor here is the nature of the range validation and update scheme. Assuming no relative angular motion and with the range validation and update scheme as we have implemented it, the maximum linear speed along the normal to the target giving satisfactory results is approximately $2.5m/s$.

- **Nature and range of the tracked feature.** If the feature is a plane, corner, edge or cylinder, then the differential characteristic is well-defined in the sense that it approximates a linear transition such as that predicted in Appendix B.2. Therefore, satisfactory performance is obtained if the feature being tracked has a well-defined differential characteristic. Within our operating range, the fixating algorithm works better with diffractive targets, that is, edges and small cylinders. This can be explained from the fact that these features tend to have better behaved differentials as was discussed in Section 5.1.4. And so if it is known *a priori* that the features to be tracked are reflective, then the differential noise model σ_δ^2 can be increased to take account of the increased observation noise. Irregular surfaces such as human bodies

have ill-defined differentials, however these can be tracked with limited success at short ranges, $0.45 - 2m$ and low relative motion angular speeds of the order of $0.35rad/s$. At short range $(0.45 - 1.5m)$, firing speed can be as high as $30Hz$ and at ranges of about $4m$ the maximum firing speed giving satisfactory results is about $25Hz$. For a fixed base-line d, the ability to measure the differential accurately deteriorates with range as shown in Appendix B.2. As a nominal indication of the above, the Tracking Sonar is able to fixate on a cylindrical feature of radius as small as $0.4m$ at a range of $3m$ moving at an angular speed of $0.6rad/s$.

- **Hardware Limitations.** With the current hardware, the differential can only be measured to a precision of $3.2\mu s$ up to a maximum value of $15 \times 3.2\mu s$. The smallest step possible with the servo is $0.047rad$ which is crude given that the minimum RCD width which can be tracked is $0.17rad$ (See β_{min} in Section 5.1.1). However, these limitations can be overcome with improved hardware designs.

- **Operation together with other Tracking Sonars.** An important consideration in the context of data fusion, is the performance of the Tracking Sonar in an environment where other Tracking Sonars are operating. When such sensors are used to track different features which are not in close proximity, the performance attainable is the same as in the preceding discussion for a single sensor. However, when the sensors are used to track the same feature or features in very close proximity ($\approx 0.2m$ apart), interference may occur if very high data rates of the order of $30Hz$ are maintained. This is because sensors begin to receive echoes from each other causing erroneous differential measurements. However, reducing the data rates to a maximum of approximately $20Hz$ gives reliable performance. Such a reduction in maximum data rates reduces

the angular speeds of relative motion possible. As before, various optimizations are possible, such as introducing an "interlacing" scheme whereby sensors do not fire synchronously for example. Such techniques can be used to raise the maximum rates albeit to levels below the maximum attainable with a single sensor; for instance, we can reach rates of about about $26Hz$ for two such sensors in close proximity ($0.3m$) tracking the same feature at a range of 0.8m through such optimizations.

Therefore, in summary, use of several Tracking Sonars in the same environment only results in a slight lowering of the performance possible with a single Tracking Sonar.

Given these factors the parameters in the Tracking Sonar can be optimized to give the best results depending on the anticipated relative motion, nature of feature and operating range and hardware capabilities. Insight into the setting of these parameters is built up with experience and consideration of some of the variations discussed in the preceding sections. The results of Section 5.3.2 suggest possible settings for various filter and other parameters. With the tracking implemented as in Figure 5.12, the fixation on a feature is somewhat indiscriminate. This is due to the fact that the sensor essentially tracks the null of the differential and updates the range accordingly. This means that if another feature coincides spatially with a feature being tracked, it is possible for the sensor to switch to the new feature apparently non-deterministically. It is possible to get round this with a more stringent validation scheme and for this, increased accuracy is required in the parameters which are measured and in the positioning of the servo. In this way the basic algorithm could even be extended to allow for momentary obscuring of the feature. This is highly desirable in cluttered environments where targets could become partially or momentarily obscured. The basic algorithm of Figure 5.12 assumes that there is always a clear path between the

feature and the sensor thus limiting this implementation to environments where this can be guaranteed.

5.4 Discussion

What we have developed and presented in this chapter is a model for a Tracking Sonar which is able to maintain focus of attention on a specified feature in real-time. There is ample room for further development of the hardware itself and optimization for various scenarios and applications. We leave the improvement of this technique to further research. Even in this prototype phase the differential sonar can be made use of effectively in its various modes of operation.

The differential sonar falls short in one area and that is our desire to be able to *directly* differentiate or obtain attribute information about the different target types. As we have shown in Section 5.1.3, the differential characteristics for the different features are indistinguishable to the hardware even though there are minor differences. Thus, the information obtained with each return (Equation 5.6) in itself gives no indication as to the nature of the target. However, as we shall show later, as an indirect consequence of the ability to fixate on a given feature, a method based on the classification algorithm presented in Chapter 3, can be developed which results in classification of the feature being tracked.

As regards the probabilistic model, the sensor parameters ζ, were found to be the significant factor determining the variances of the measured quantities r and θ. However, the measured quantities themselves have also been shown to be factor. Hence, a more rigorous model of the dependence of the variances of r and θ with the measured quantities themselves is required. The functions $g_r(r)$ and g_θ given in Appendix B.2 are only approximations obtained through measuring the actual variances, similar to those presented in Table 5.1,

for various ranges and orientations and obtaining functions which approximately fit the observed results.

5.4.1 A Modular Implementation

The Tracking Sonar is packaged in a modular fashion, based on a generic decentralized architecture that we have developed. The architecture provides the requisite processing power using a T805 Transputer with full floating point ALU. Communication links are available for connection to other similar sensing nodes. Appendix C describes the architecture in more detail. Figure 5.19 shows a complete Tracking Sonar based sensing node built using this architecture. Such a modular implementation of the Tracking Sonar has a significant

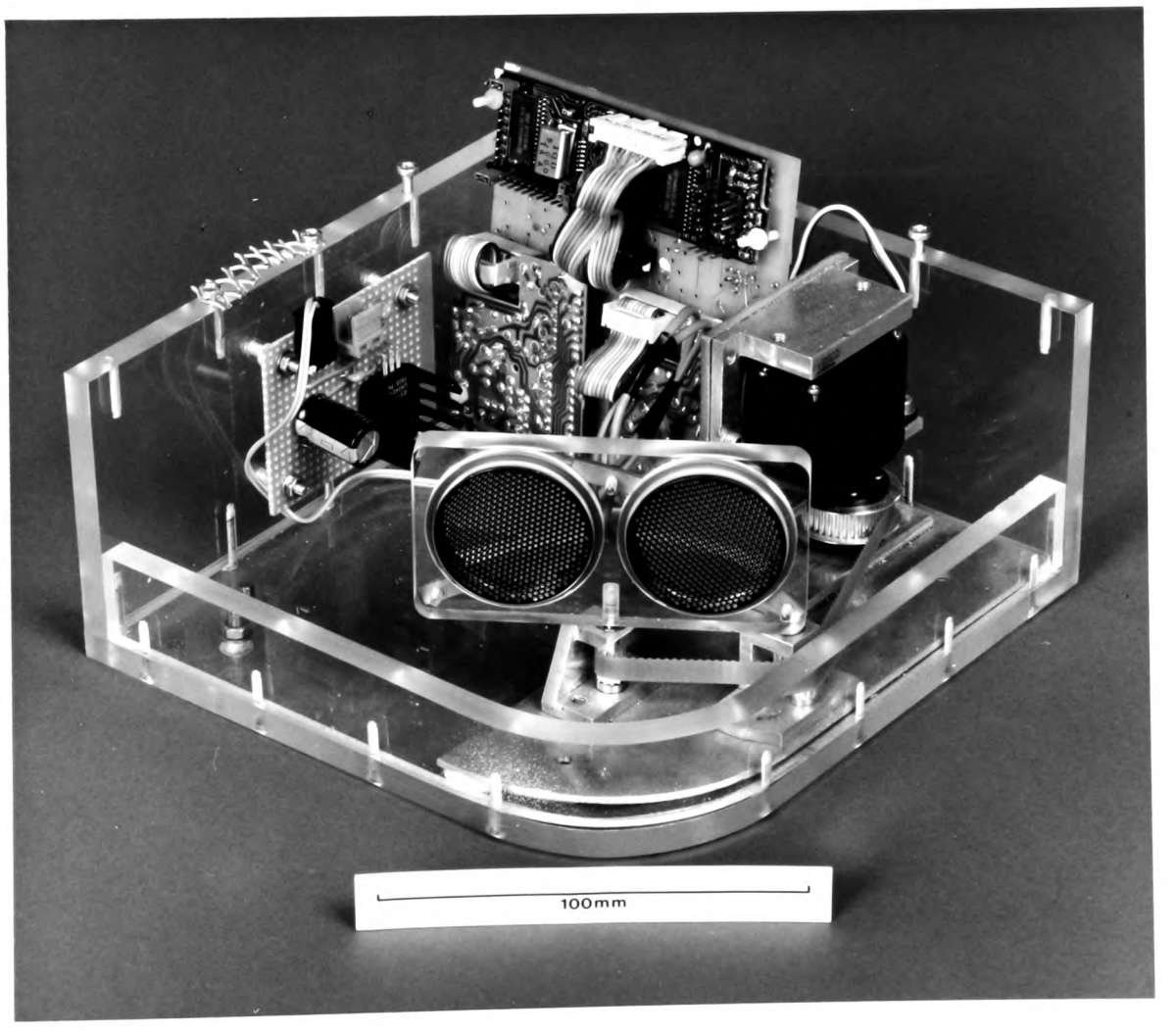


Figure 5.19: A completed modular Tracking Sonar sensing node.

impact on the way multi-sensor fusion algorithms, for vehicle applications in particular, can be realized:

- Firstly, the sensor is fully autonomous and as such well suited to the distributed and decentralized multi-sensor architectures developed in Chapter 3. The availability of processing power means that complex algorithms can be implemented *locally* on each sensor node.

- Secondly, the ability of each Tracking Sonar to focus attention on a particular given feature means that in a decentralized system of such sensors, each one can focus on a particular aspect of the problem and all the results fused to give a global result. This makes the sensor an ideal candidate for the implementation of the algorithms developed in Chapter 3.

- Thirdly, the ability to focus attention only on a selected feature presupposes ability to decide on features to focus on. This thus presents an ideal situation requiring the making of decisions given possibly several alternatives, this being the sensor management problem.

And so, we have developed a sensor for data fusion which also needs to be managed, these being the subjects of Chapter 6 and 7 respectively.

Chapter 6

Decentralized Data Fusion for Navigation

6.1 Robot Navigation

Mobile robotics research has been dominated by work aimed at developing a competence for navigation and guidance. There are several aspects of this to be considered; firstly, low-level capabilities of *obstacle avoidance*, *localization* and *map-building*, secondly higher-level competences for *path planning* and *task planning*. The higher level competences presuppose the lower level ones. The work outlined in this thesis thus far permits us to make contributions towards developing a competence for localization and map-building. Localization is defined as the process of answering the question "*where am I* ?"[90][45]. The basis of most localization techniques is the use of beacons with respect to which location can be computed [40][23] [1] . Put in the context of this thesis, localization is thus concerned with the gain of information as in Equation 2.4 where the state $\mathbf{x}$ is the location of a vehicle based on observations $\{\mathbf{Z}^k\}$ of beacons.

Although the process of localization is well understood in maritime and aerospace systems, the difficulties in autonomous robot navigation arise from the use of uncertain sensor observations to correlate or maintain correspondence with navigational beacons. The problem is exacerbated when attempts are made to use naturally occurring features in the

[1] There are alternatives to this such as the use of *occupancy grids* described by Moravec [103] and Elfes[51].

environment as beacons. Representative of the state of art in navigation without artificially-setup beacons, is the work by Leonard and Durrant-Whyte [90] on which the work presented here is based. Leonard develops a competence for localization using sonar and an algorithm based on the standard centralized EKF which matches extracted RCD information with a given map and estimates vehicle location on the basis of matched RCDs. This being done every processing cycle. The net result of this approach is that the vehicle executes a "move-stop-look" sequence with considerable time being spent extracting RCDs and matching them with the map [2]. In suggesting improvements, Leonard states;

> For fast localization we want to build a system that maintains *continuous* map contact, ... The power behind this approach is that when correspondence can be assured, by "locking on" to a target, a high bandwidth stream of easily usable [correctly associated] information becomes available...We envision a robot which maintains *continuous map contact* "grabbing hold" of corners, planes and cylinders in the environment, using them as hand rails.. by maintaining continual correspondence with some subset of map features perception can be put in the loop to provide high bandwidth position control. [88]

It is with respect to this, that this thesis makes a contribution to vehicle navigation. Our starting point is the Tracking Sonar developed in the previous chapter with the ability to focus attention on a given geometrical feature such as the ones naturally occurring in indoor environments. The following are the areas that are addressed in this chapter:

1. Using the algorithms of Chapter 3, we develop a single Tracking Sonar into a self-contained guidance sensor which having established correspondence with a known feature in an *a priori* map provides a high bandwidth of localization estimates regardless of the vehicle or platform motion for as long as focus of attention is maintained! This is extended into a decentralized localization scheme.

[2] Leonard describes using high density scans taking up to 2 minutes to obtain. However, he reports at best location estimates taking about 1 second to obtain [88]!

2. Focus of attention can be used to initiate tracks of previously unknown or unassociated RCDs to build maps. In addition to using the Tracking Sonar for map-building by tracking new features, map-building can also be enhanced by classifying the features being observed into their geometric types. In this respect, a technique based on the classification algorithm of Chapter 3, is developed for classifying features focused on by the Tracking Sonar.

6.2 Tracking-Sonar Localization

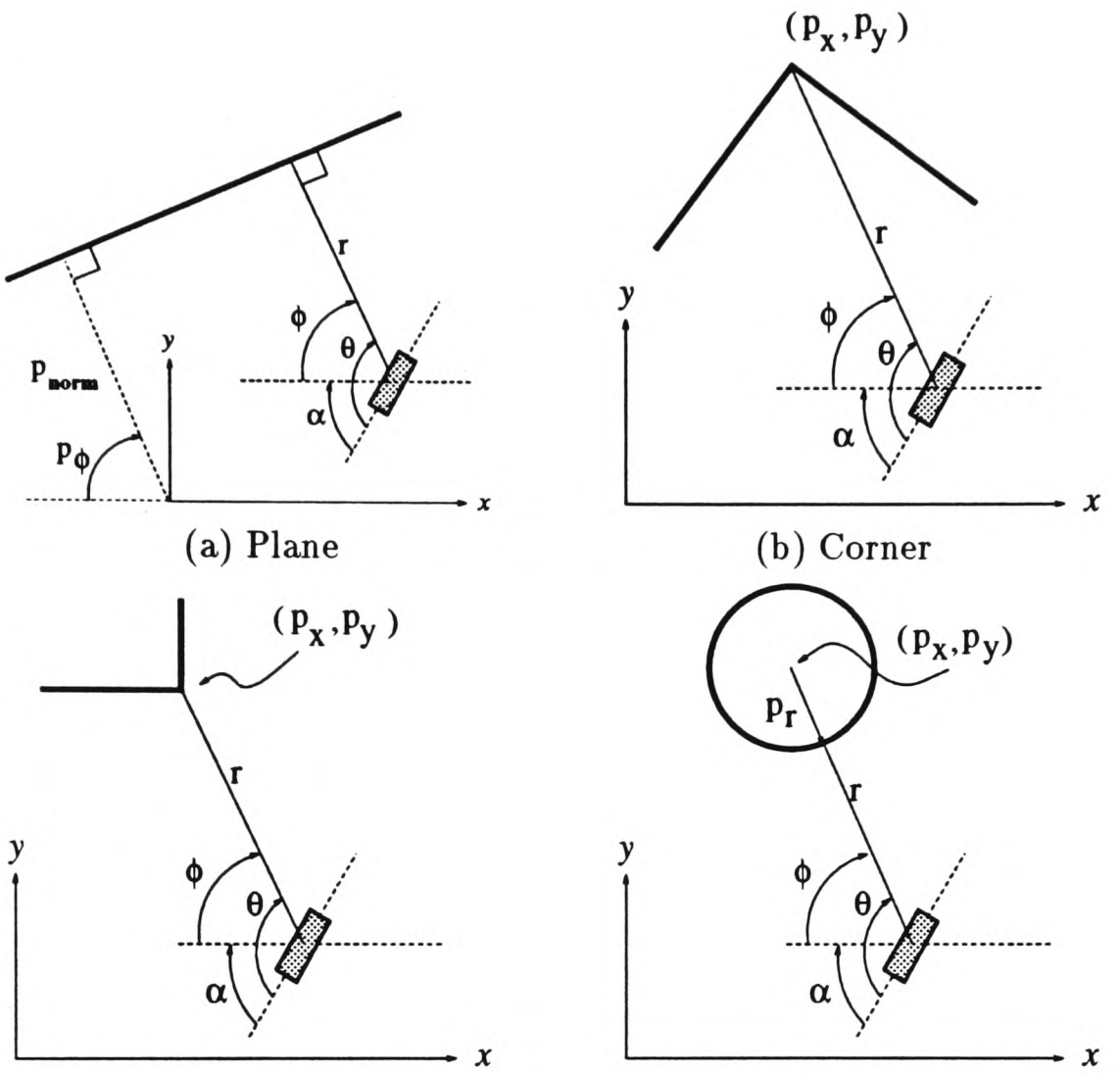
6.2.1 Feature Models

Prior to introducing the localization scheme, we define models for the naturally occurring environmental features that are used as beacons. The feature model assumes the same basic types that were considered when analysing the differential in the previous chapter, that is planes, corners edges and cylinders. In each of the following model descriptions, α is the orientation of the sensor with respect to the coordinate model of the environment, and ϕ is the bearing of the target from the sensor with respect to the coordinate model of the environment.

1. **Plane.** A plane is defined by

$$t_p = (p_{norm}, p_\phi, p_v), \tag{6.1}$$

where p_{norm} is the length of the normal from the plane to the Cartesian coordinate origin of the environment Cartesian coordinate origin and p_ϕ is the bearing of this normal and p_v defines which side the plane is visible from. These parameters are as illustrated in Figure 6.1(a).



(a) Plane (b) Corner

(c) Edge (d) Cylinder

Figure 6.1: Feature models in global coordinate system.

2. **Corner.** A corner is defined by

$$t_c = (p_x, p_y), \tag{6.2}$$

where p_x and p_y are the coordinates of the location of the point of intersection of the planes making up the corner. It is assumed as before that the corner is right-angled. The corner model is illustrated in Figure 6.1(b).

3. **Edge.** An edge is defined by

$$t_e = (p_x, p_y), \tag{6.3}$$

where p_x and p_y are the coordinates of the location of the point of intersection of the planes making up the edge. This is illustrated in Figure 6.1(c).

4. **Cylinder.** A cylinder is defined by

$$t_{cyl} = (p_x, p_y, p_r), \tag{6.4}$$

As before, p_x and p_y are the coordinates of the center of the cylinder and p_r is the radius of the cylinder. The cylinder is illustrated in Figure 6.1(d).

6.2.2 Localization Algorithm

The localization algorithm proceeds as in the non-linear information filter of Chapter 3. However, for this particular application we require to define the following:

State

The location of the sensor platform in the environment is defined as the state $\mathbf{x}(k) = [x(k), y(k), \alpha(k)]^T$, where $x(k)$ and $y(k)$ are the Cartesian coordinates of the platform location and $\alpha(k)$ is the orientation at time k. An implicit assumption is that the location of the sensor on the platform coincides with the center of the platform indicated by $\mathbf{x}(k)$ above. If however there is an offset, a simple transformation can be used to take account of this.

Observations

The Tracking Sonar makes an observation at time k of the parameters r and θ for a given RCD target whose definition t_a is known. The observation vector $\mathbf{z}(k) = [r(k), \theta(k)]^T$, is a function of the state and the feature being observed that is

$$\mathbf{z}(k) = \mathbf{h}[k, t, \mathbf{x}(k)] + \mathbf{v}(k). \tag{6.5}$$

From Figure 6.1(a), the observation model for the plane is given by

$$\mathbf{h}[k, t_p, \mathbf{x}(k)] = \begin{bmatrix} r(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} p_{norm} - x(k)\cos(p_\phi) - y(k)\sin(p_\phi) \\ p_\phi - \alpha(k) \end{bmatrix}. \quad (6.6)$$

From Figure 6.1(b) the observation model for a corner is given by

$$\mathbf{h}[k, t_c, \mathbf{x}(k)] = \begin{bmatrix} r(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(p_x - x(k))^2 + (p_y - y(k))^2} \\ \tan^{-1}\left[\frac{p_y - y(k)}{p_x - x(k)}\right] - \alpha(k) \end{bmatrix}. \quad (6.7)$$

Since the t definition of the edge is the same as that for a corner, their observation models are the same that is $\mathbf{h}[k, t_c, \mathbf{x}(k)] = \mathbf{h}[k, t_e, \mathbf{x}(k)]$. The observation model for the cylinder is given by

$$\mathbf{h}[k, t_{cyl}, \mathbf{x}(k)] = \begin{bmatrix} r(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(p_x - x(k))^2 + (p_y - y(k))^2} - p_r \\ \tan^{-1}\left[\frac{p_y - y(k)}{p_x - x(k)}\right] - \alpha(k) \end{bmatrix}. \quad (6.8)$$

The observation noise in Equation 6.5 is as defined in Chapter 3 with covariance matrix $\mathbf{R}(k)$. The no-correlation assumption for the observation noise for the Tracking Sonar has been justified in Chapter 5, for particular operational ranges of parameters. As was observed in Chapter 5, the variances associated with the observations vary in magnitude with the observations themselves. Therefore, the covariance matrix of the observation noise is more accurately modelled as discussed in Chapter 5 as $\mathbf{R}[k, \mathbf{z}(k)]$.

It will be noted that unlike the algorithm of Leonard [88], we make use of *both* the range and and the orientation in the localization cycle. Leonard's reason for not using orientation is the uncertainty associated with the orientation using the standard Polaroid device. This is overcome by the fact that in addition to range, accurate orientation information with a known noise model can be obtained as described in Chapter 5.

State Transition

The state transition equation, from Chapter 3, is given by

$$\mathbf{x}(k+1) = \mathbf{f}[k, \mathbf{x}(k), \mathbf{u}(k)] + \mathbf{w}(k), \tag{6.9}$$

where $\mathbf{w}(k)$ is the noise associated with the transition with covariance matrix $\mathbf{Q}(k)$. The control input $\mathbf{u}(k)$ represents the motion of the sensor platform. Various methods have been proposed to model the kinematics of robot vehicles some of which are quite complex [127][2]. A computationally simple and useful model is the one based on the point kinematic model [127]. In this model, every motion can be represented by a rotational component and a linear component. And so the control is written as

$$\mathbf{u}(k) = [T(k), \Delta\alpha(k)]^T, \tag{6.10}$$

where $T(k)$ is the linear component of the motion at time-step k and $\Delta\alpha(k)$ is the rotational motion at time-step k. The transition model can be written

$$\mathbf{f}[k, \mathbf{x}(k), \mathbf{u}(k)] = \begin{bmatrix} x(k) + T(k)\cos(\alpha(k)) \\ y(k) + T(k)\sin(\alpha(k)) \\ \alpha(k) + \Delta\alpha(k) \end{bmatrix}. \tag{6.11}$$

Estimation

Estimation of the sensor platform location is done using the non-linear information filter outlined in Chapter 3, where Equation 3.47 gives the estimate $\hat{\mathbf{x}}(k \mid k)$.

6.2.3 Implementation

In the implementation of the localization algorithm, the following assumptions are made:

Assumption 3 The environment is a static two-dimensional world consisting of the feature types described above.

This assumption follows from the way our sonar sensing model has been developed. The features are expected to be static because the algorithm does not include target dynamics. However, the algorithm could be extended to include non-static features and a dynamic map.

Assumption 4 Available is an accurate map from which predicted RCD observations can be generated and a corresponding matching facility for matching observed RCDs and with the predicted ones.

This assumption is based on the fact that the localization filter expects information about the feature to be used for localization in the form of a model t for the particular feature as shown in Figure 6.1. After extracting RCDs, they are matched with those predicted from the environment map. Predicted RCDs are generated from a line-segment map of the environment using the sonar model of Chapter 5, given the location of the vehicle [85][88]. The matching algorithm can simply be based on a validation gate [19][133]. More complex methods exist such as that based on the Mahalanobis distance [53], Nearest neighbour association [116] and the PDAF and JPDAF [32]. Since we do not deal with the matching problem directly, we assume the matching algorithm as described in [90] and move on.

If after scanning no matches are made, the scan and extraction process is repeated. If several matches are made, one is selected based on some criteria. In a simple strategy the best matched RCD is selected. The parameters of the selected RCD can be used to initialize the information filter in addition to being used to initialize the Tracking Sonar algorithm (See Figure 5.12). The Tracking Sonar algorithm and the information filter are then executed in parallel. Of great significance is the fact that, the state estimates $\hat{\mathbf{x}}(k \mid k)$ are obtained at the sonar return rate. The parallel execution terminates when the tracking

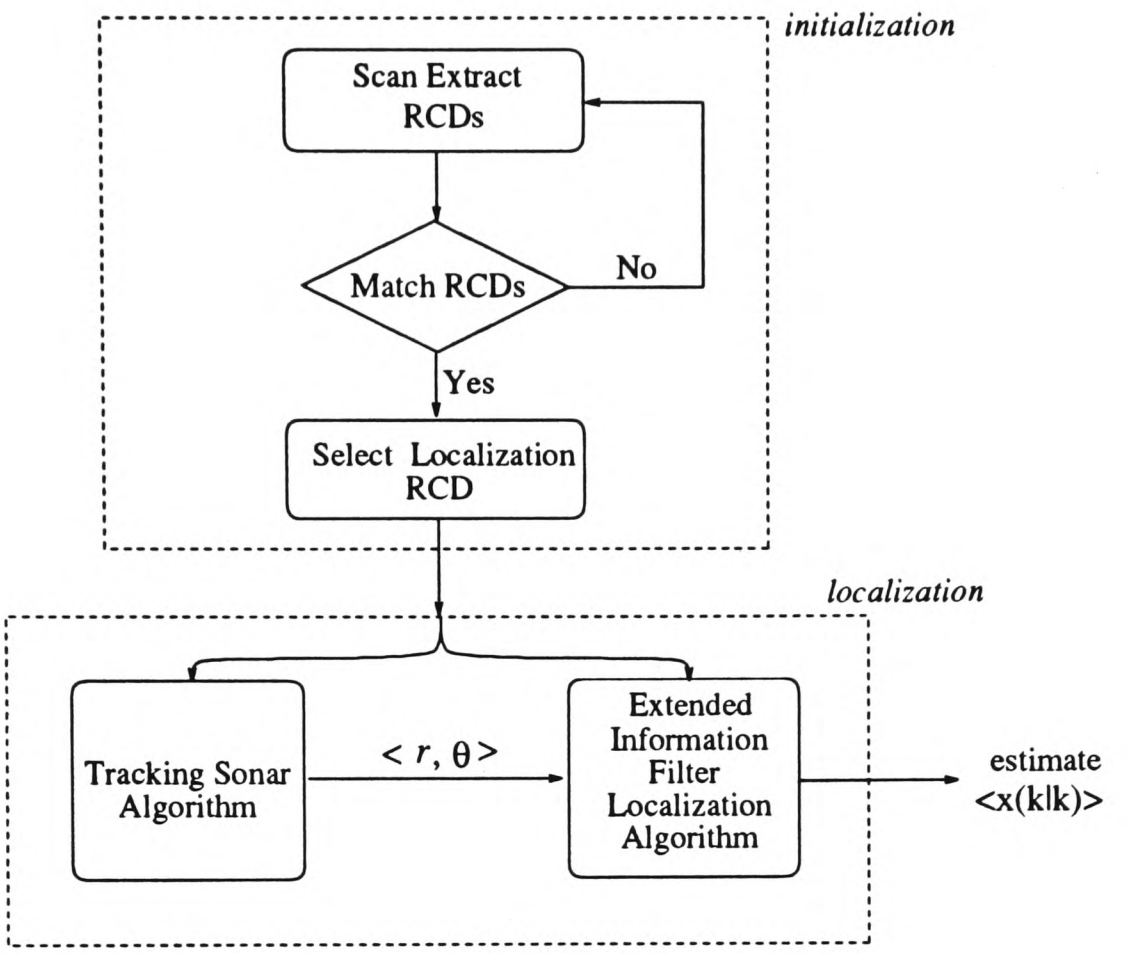


Figure 6.2: Localization algorithm. The Tracking Sonar and Localization algorithms are executed in parallel.

algorithm signals that the feature has been lost. The whole procedure outlined in Figure 6.2 can then be repeated.

6.2.4 Results and Performance

The following results illustrate the performance of the localization algorithm. The estimates shown in all the results that follow were obtained in real-time at a data rate of approximately $30Hz$.

In the first instance, the localization results for a stationary sensor platform for various observation noise models $\mathbf{R}[k, \mathbf{z}(k)]$ is illustrated. The observation noise variances for the observed parameters range r, and orientation θ have been discussed in Chapter 5 and some measured results are illustrated in Table 5.1. In all the results, the transition noise matrix

$\mathbf{Q}(k)$ is a fixed diagonal matrix with the diagonal elements set to 0.001. The results of Figure 6.3 show how increasing the observation noise model stabilizes the estimates for the stationary platform. The results of Figure 6.4 illustrate motion in the x-axis and y-axis for two observation noise models. Other types of motion such as pure rotations and linear and rotational motion are also possible as long as the control input $\mathbf{u}(k)$ of Equation 6.10 is given. Figure 6.5 shows some results of localization estimates for motion which includes rotation.

In order to evaluate the positional accuracy of the localization estimates, these can be compared with hand measured locations. In the results that follow, the platform moves from a known position B to a known position C and then back to B. The path followed between the points B and C is arbitrary in each case. The sensor platform stops at C for a fixed time. The effect of the observation noise model on the accuracy of localization and speed of response can be seen in the results of Figure 6.6. In Figure 6.6 (b) and (d) where the observation noise is modelled as being relatively large, the absolute error between the estimated positions and hand-measured ones at C and B is considerable. In fact, according to the estimates of Figure 6.6 (b) and (d), the platform never reaches C and does not return to the original position B.

The smaller observation noise model gives a faster response but is susceptible to biases and uncertainty in the actual observations. In practice therefore, the actual relative level of the observation noise model chosen depends on such factors as speed of the motion and the positional accuracy with which the feature used for localization is matched. In addition it must be noted that the motion in this case is modelled by providing a relatively large process noise model [19][90]. With a knowledge of the control $\mathbf{u}(k)$, better estimates can be obtained. In all the results shown, the feature being used for localization was an edge.

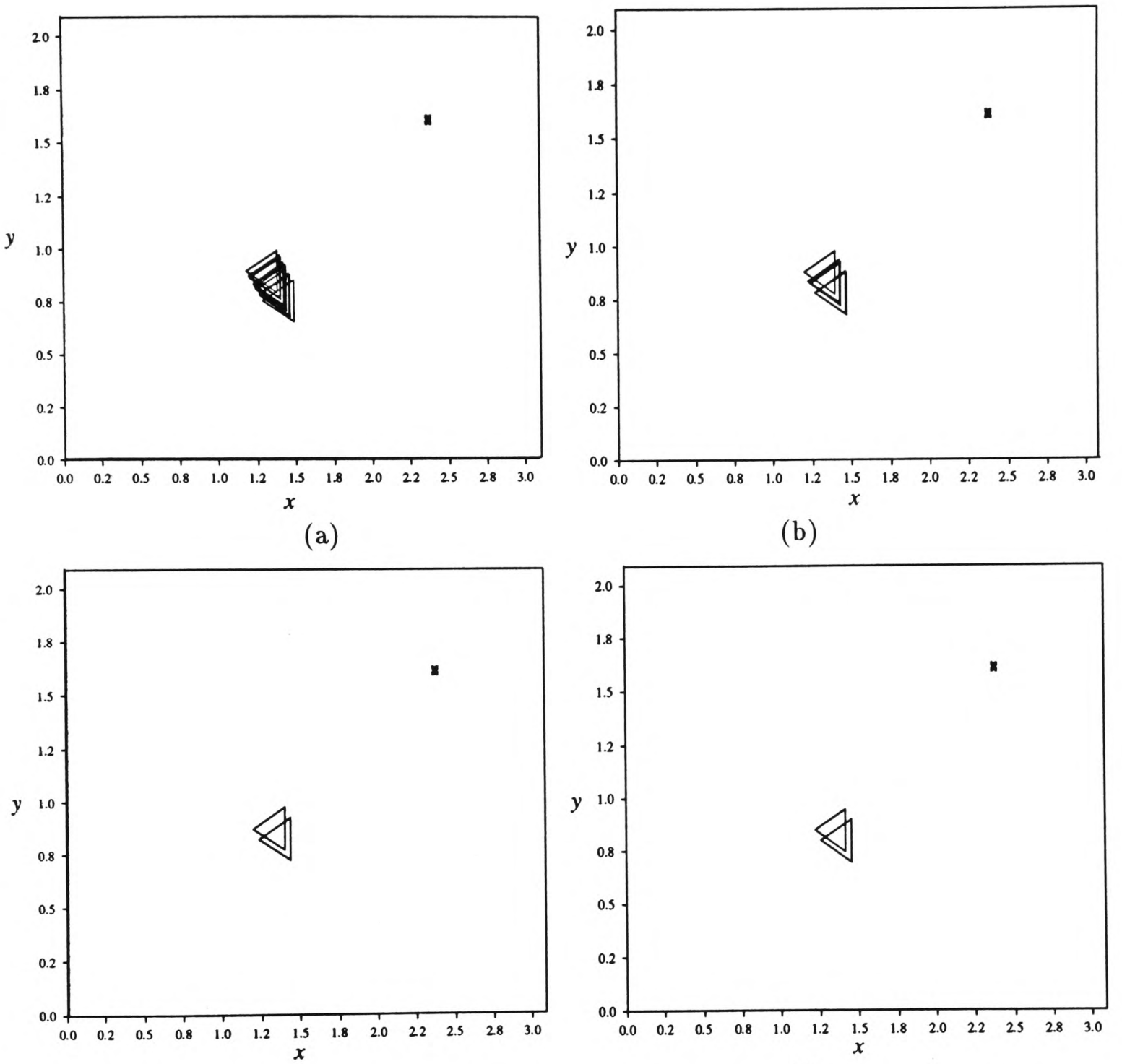


Figure 6.3: Localization estimates in global coordinates for a stationary sensor platform using various observation noise models. In the figures, positional changes of at least 0.04m are noted. In (a) $\sigma_r^2 = \sigma_\theta^2 = 0.0001$, (b) $\sigma_r^2 = 0.001$, $\sigma_\theta = 0.001$ (c) $\sigma_r^2 = 0.005$, $\sigma_\theta^2 = 0.01$ (d) $\sigma_r^2 = 0.01$, $\sigma_\theta^2 = 0.1$. The transition noise matrix $\mathbf{Q}(k)$ was fixed. The position "X" marks the location of the feature used for localization.

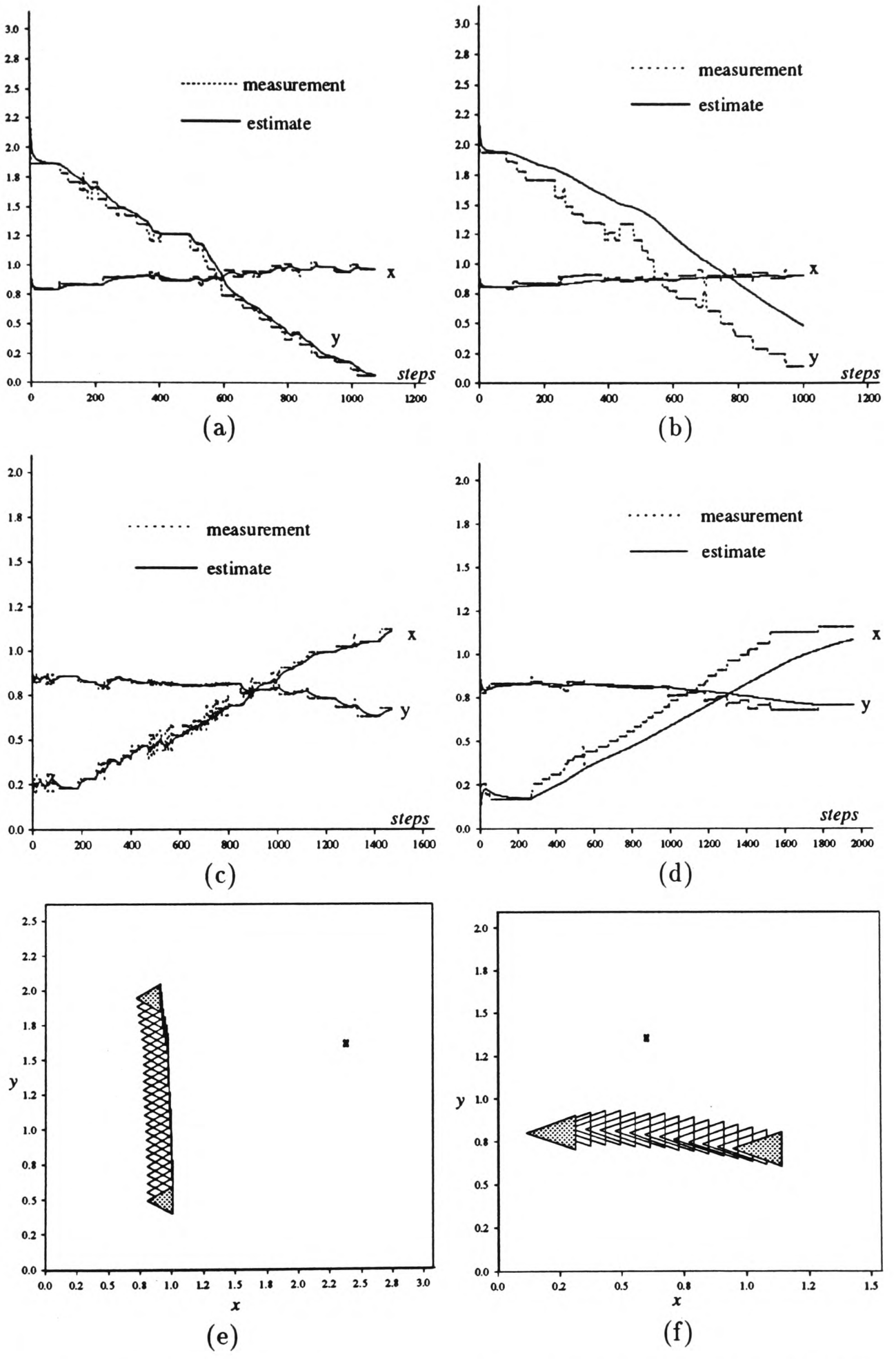


Figure 6.4: Localization for motion in x-axis and y-axis for 2 observation noise models. In (a) and (b) motion is in the y-axis and in (c) and (d) motion is in the x-axis. In (a) and (c) $\sigma_r^2 = 0.005$, $\sigma_\theta^2 = 0.01$, (b) and (d) $\sigma_r^2 = 0.05$, $\sigma_\theta^2 = 0.5$. The transition noise matrix $\mathbf{Q}(k)$ was fixed. (e) and (f) show the motion plotted in (b) and (d).

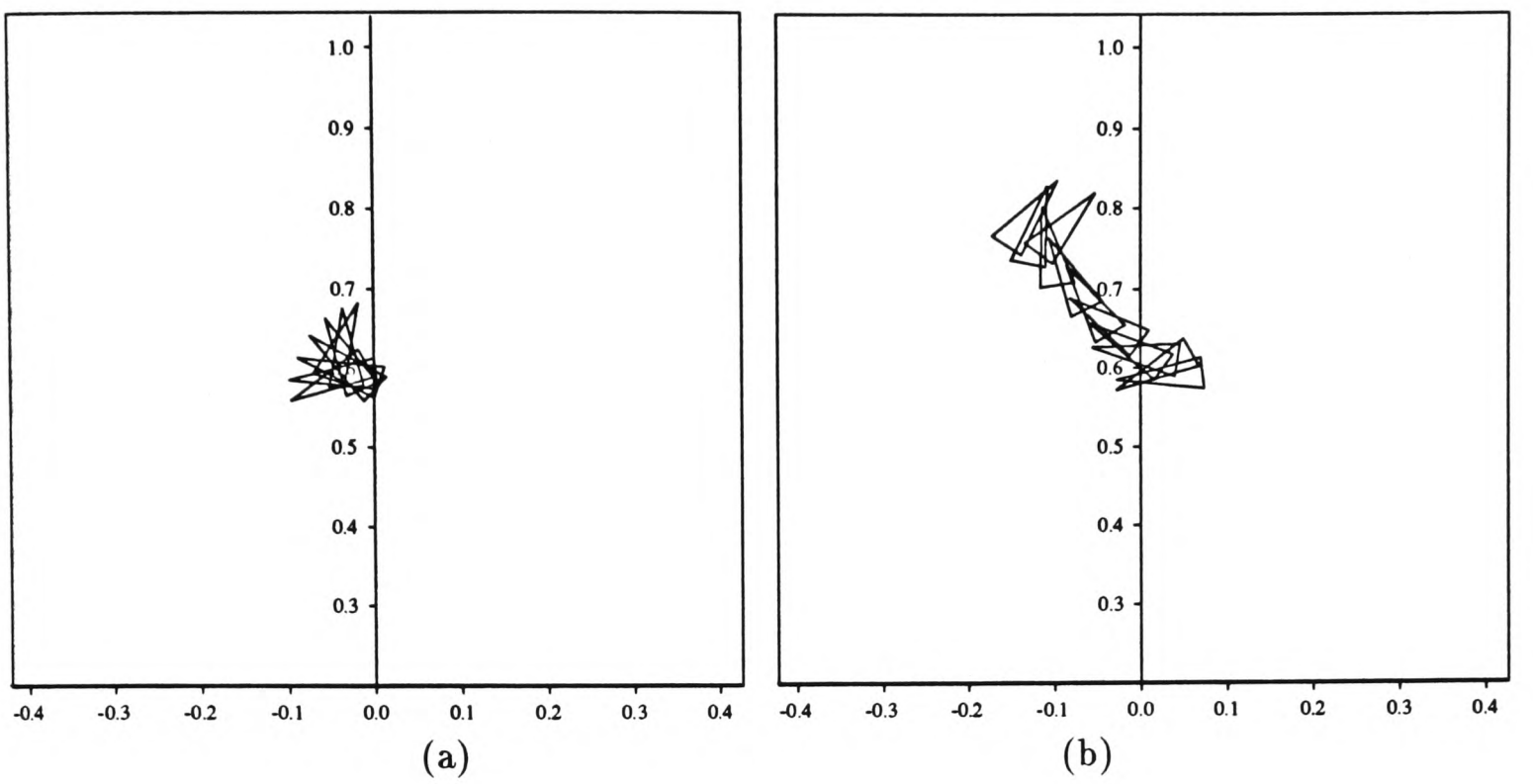


Figure 6.5: Localization estimates for sensor platform incorporating rotational motion (a) pure rotation (b) linear and rotational motion.

Similar results are obtained with the other feature types presented in Section 6.2.1.

6.3 A Decentralized Localization Scheme

The Tracking Sonar localization algorithm of Section 6.2 can be implemented decentrally by making use of several Tracking Sonars. The estimates from each sensor are fused to give a global estimate of the position of a vehicle on which they are mounted. This provides a fast way of estimating a vehicle's position given a map of the environment. A description of the implementation of a such a decentralized localization scheme is described next.

6.3.1 Algorithm and Implementation

The localization scheme assumes the same feature models as discussed in Section 6.2 and makes the same assumptions. Use is made of the decentralized non-linear information filter of Chapter 3. Each sensor operates as described in Section 6.2 by scanning and

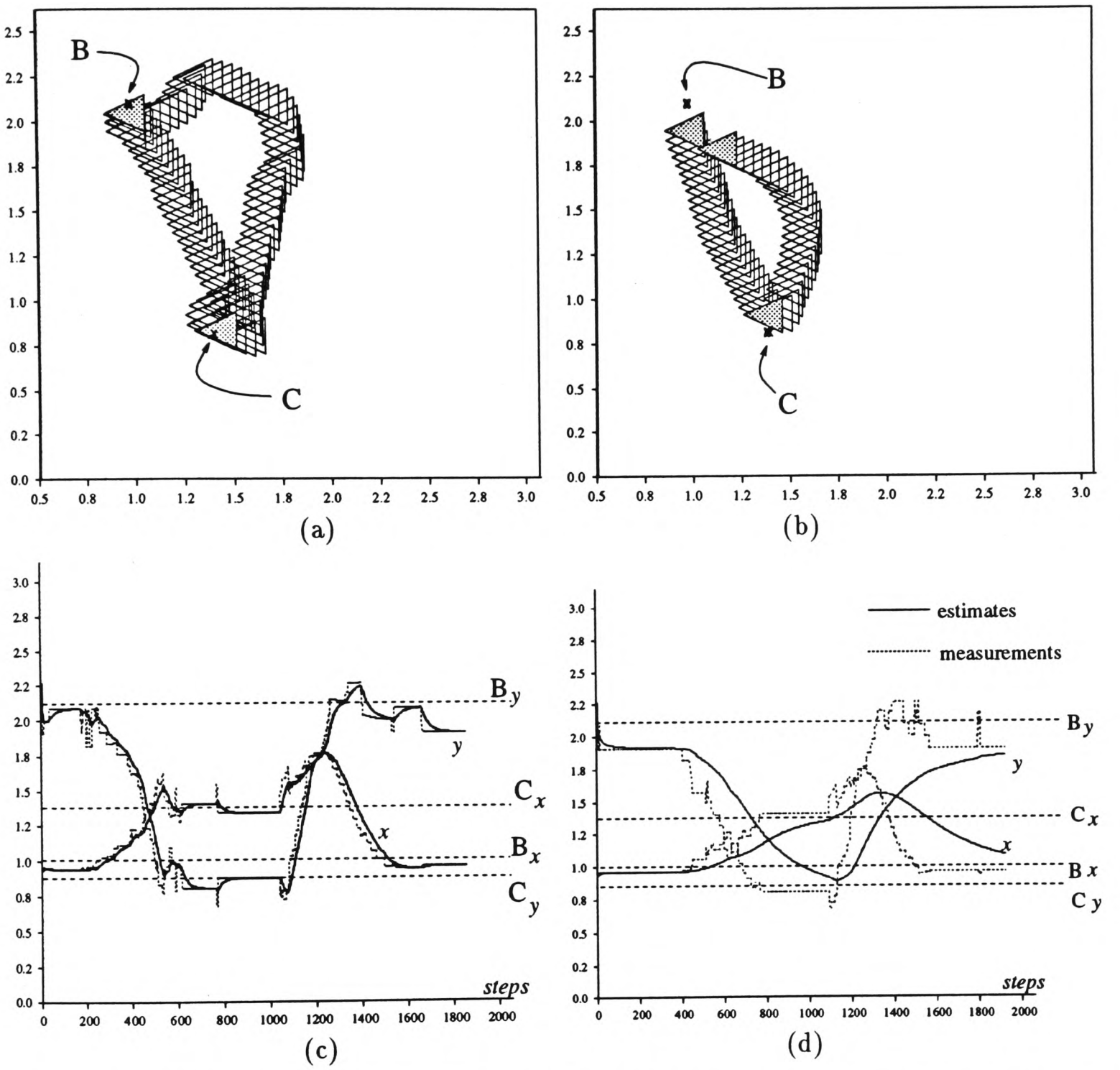


Figure 6.6: Motion from B to C and back to B stopping at C and B for a fixed period (10 seconds). The true (hand-measured) positions B and C are indicated. (a) and (b) show the estimated location of the platform. (c) and (d) show x y positions calculated directly from observed r and θ and the estimated x and y from the localization information filter. In (a) and (c) $\sigma_r^2 = 0.0001$, $\sigma_\theta^2 = 0.001$, in (b) and (d) $\sigma_r^2 = 0.05$, $\sigma_\theta^2 = 0.5$.

extracting RCDs. Each Tracking Sonar fixates and tracks the most confidently matched RCD. Various other criteria are used for selection of the RCD to be tracked. In our prototype implementation the sensor chooses to track the closest RCD matched.

Each sensor currently tracking a matched RCD generates estimates of its location rather than the location of the vehicle centroid. This necessitates the redefinition of the state to be the location of the central axis of the vehicle rather than the sensor itself (as in Section 6.2). A simple geometric transformation Φ, is applied to the location of the sensor to give the location of the vehicle given knowledge of the location of the sensor on the vehicle

$$\mathbf{x}(k) = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} = \Phi \mathbf{x}_s(k) = \Phi \begin{bmatrix} x_s \\ y_s \\ \alpha_s \end{bmatrix} \tag{6.12}$$

Bearing in mind this transformation and using Equation 3.86 to 3.88, the state corresponding to the location and orientation of the vehicle can be estimated. Partial estimates and information is communicated only between the sensors that are *currently* tracking a matched RCD. At each of these sensors, the estimate $\hat{\mathbf{x}}_i(k \mid k)$ is obtained from Equation 3.100.

For such a scheme, the sensors are configured as shown in Figure 6.7 for the OxNav vehicle (See Appendix C). Several situations arise: Firstly, a sensor may extract RCDs which fail to match the expected ones. In this case the sensor may track the "most significant" RCD observed for purposes of map-building that is track initiation [90] and is therefore left out of the decentralized localization fusion. Secondly, a sensor may "lose" the RCD which it is currently tracking as will occasionally happen, as discussed in Chapter 5. Such a sensor re-initializes by scanning and extracting RCDs and then attempting matches as before.

The results of an implementation using two such sensors on the OxNav vehicle are presented. In this implementation the matching facility is simulated by having features in known specified locations, that is features whose models t are known (See Section 6.2.1) are

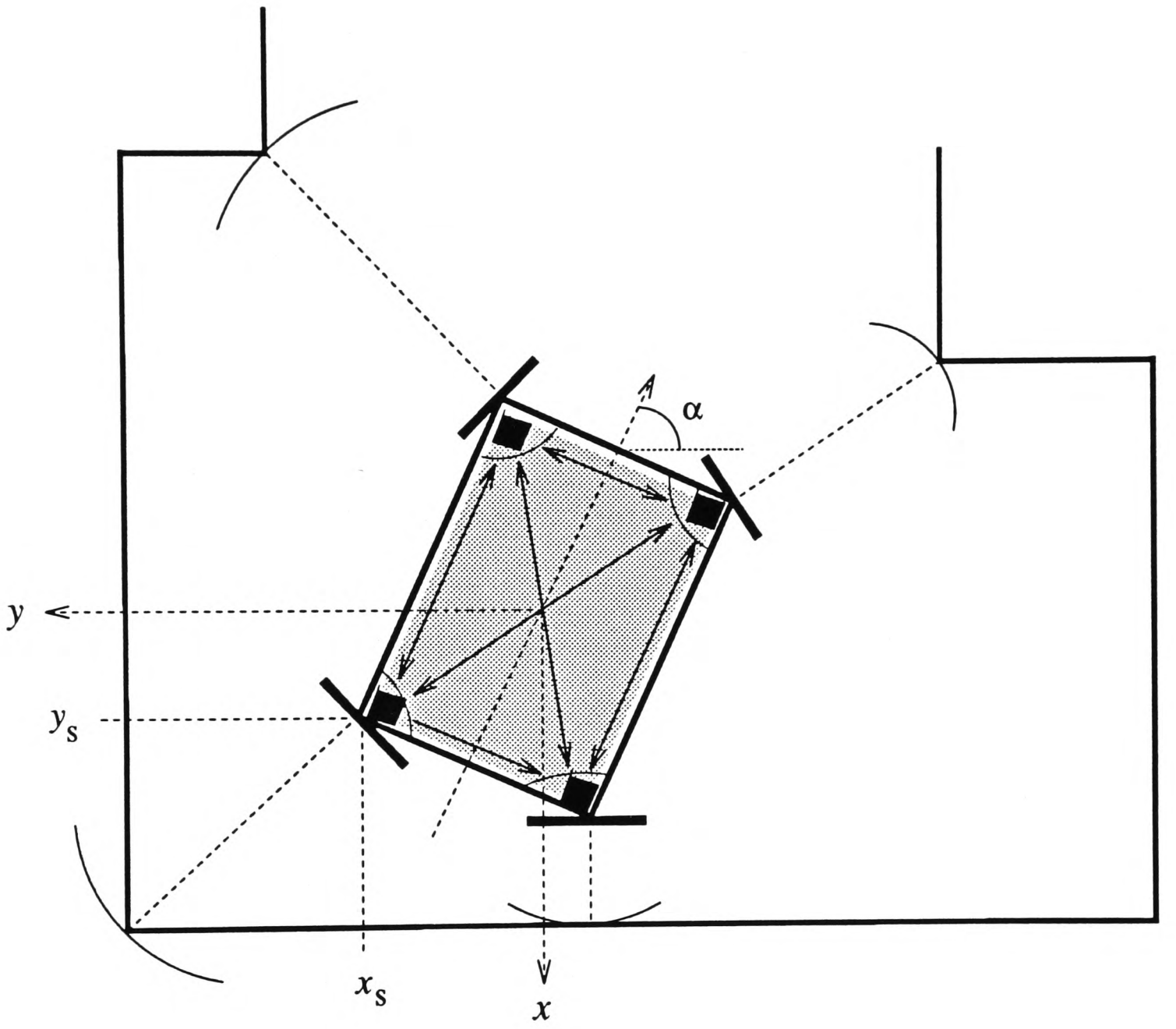


Figure 6.7: Tracking Sonar configuration for the OxNav vehicle. Each sensor focusses attention on a particular feature of the environment for purposes of localization or map-building. The sensors are decentralized and fully connected.

used for localization.

6.3.2 Results and Performance

In all the results that follow, the sensors track features which are in the $0.40 - 4.00m$ operational range. The estimates showed are obtained at the data rate which for the 2-sensor system is approximately $22Hz$.

Shown first is the effect of having two sources of observation, firstly, from different features and secondly from the same feature. The results of Figure 6.8 (a) and (b) show the localization estimates for a stationary vehicle localizing with each sensor tracking a different feature. In Figure 6.8 (c) and (d), the sensors track the same feature. In the results of Figure 6.8(c) and (d), sensor 2 loses the feature it is localizing on at time-step 550. This can be explained by the inference which can result when a sensor receives acoustic signals from the other (See discussion in Chapter 5). This has the effect that the estimate of the vehicle's location then tends towards that suggested by the remaining sensor's observations.

The vehicle can be moved in an "un-smooth" manner by pushing it along or introducing jerky motion. Results from such un-smooth motion are useful as they test the localization for robustness when vehicle motion is not properly modelled. The results of Figure 6.9 show estimates when the vehicle is moving linearly in the x-axis. Use is made of two different parameter models which results in two different observation models. In addition, the results of Figure 6.9 are used to further show the effect of localizing on different features. In Figure 6.9 (a) and (b) each sensor tracks a different feature and in (c) and (d) the sensors track the same feature. As expected, the smaller observation noise gives the better time response. In these results, use is made of a high data rate of $30Hz$ which results in sensor 1 quickly losing the feature when the sensors track the same feature in (c) and (d). In the results of

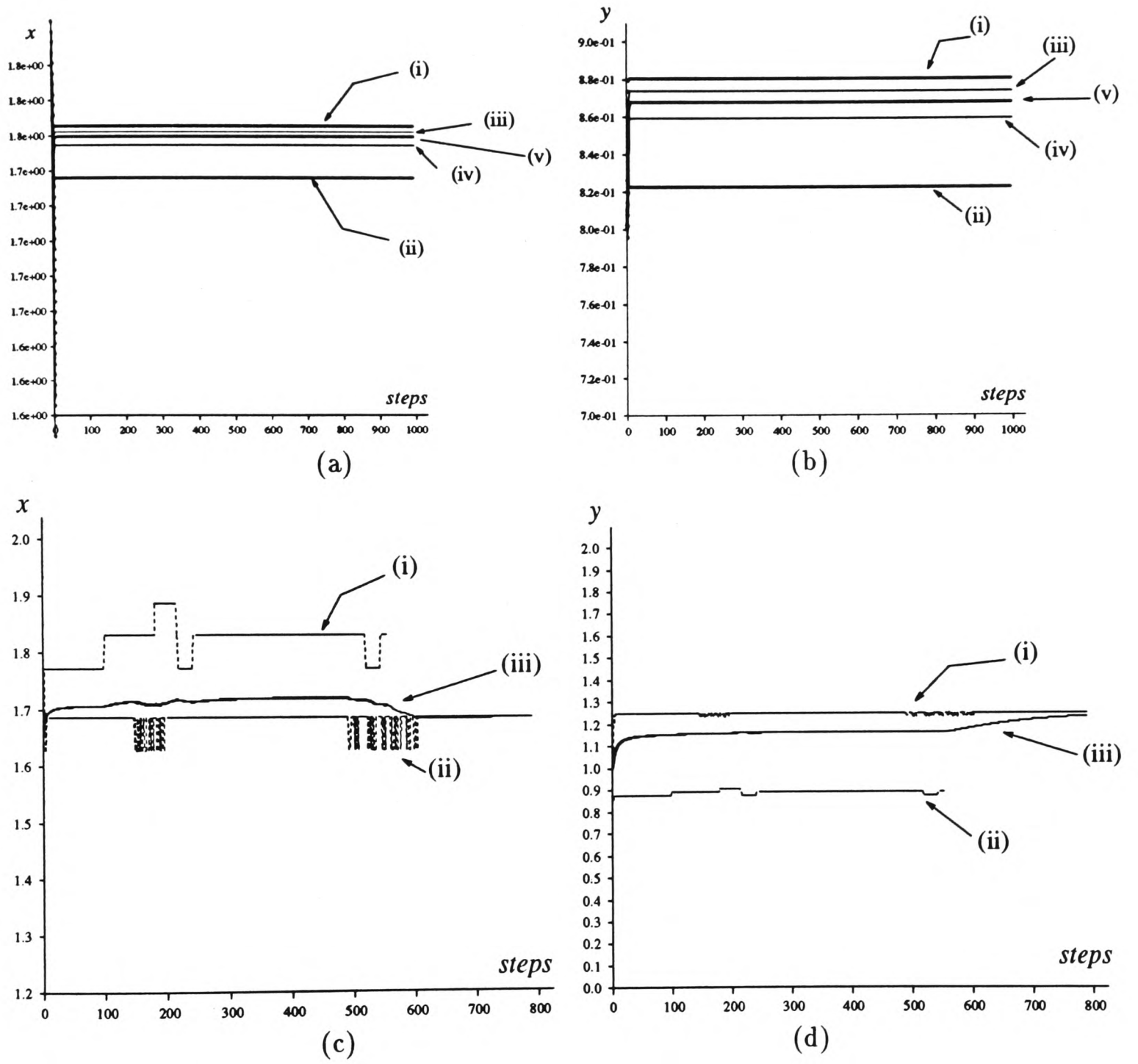


Figure 6.8: Location estimates for a stationary vehicle using 2 tracking sonars. In (a) and (b) the sensors track different features and in (c) and (d) they track the same feature. (i) Location calculated directly from observation at sensor 1 and (ii) calculated from observations at sensor 2, (iii) and (iv) are the partial estimates at each sensor and (v) is the global estimate at each sensor. In (c) and (d) (iii) are the partial and global estimates of location.

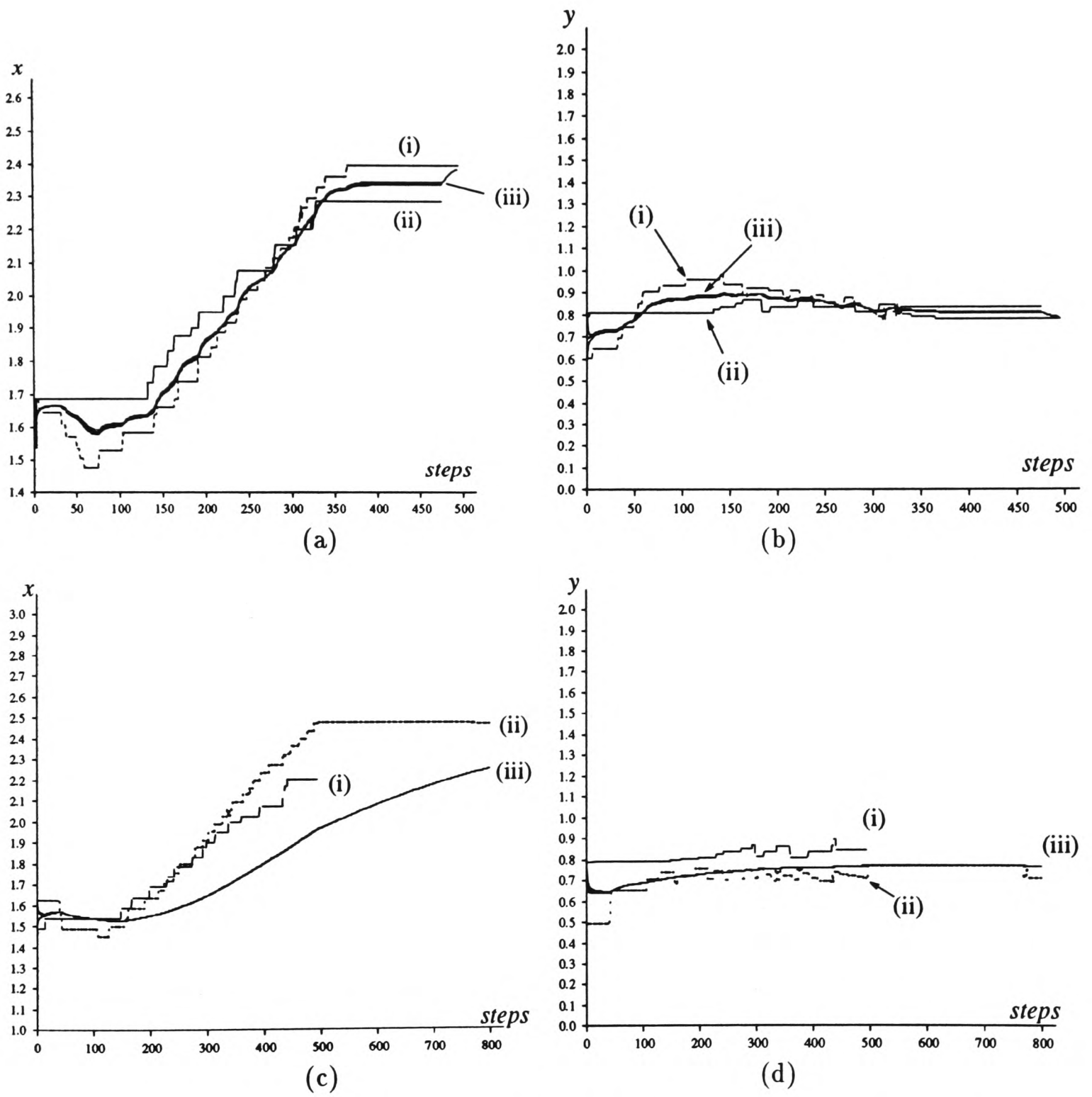


Figure 6.9: Location estimates for a vehicle which is moving in the x-axis. In (a) and (b) a small observation noise model is used and each sensor tracks a different feature. In (c) and (d), a large observation noise model is used and each sensor tracks the same feature. Sensor 1 loses its feature in (c) and (d).(i) sensor 1 observations, (ii) sensor 2 observations (iii) estimates.

Figures 6.8 and 6.9, it can be seen that the partial estimates $\tilde{\mathbf{x}}_i$ are very close to the global estimates which themselves are identical as expected in a fully connected system.

The accuracy of the localization can be compared with hand-measured values in a sequence in which the vehicle moves to and from known locations. The results in Figure 6.10 are for sensors localizing on the same feature at different ranges. It can be seen that at the closer range (Figure 6.10 (a) and (b)) the estimates are closer to the hand-measured values than at the longer range ((c) and (d)). The results of Figure 6.10 show that the accuracy decreases with range. This can be seen quite clearly in (c) and (d). In addition, although motion is only in the x-axis, at time $k = 900$ sensor 2 drifts off and starts tracking some other feature which results in a suggestion of motion particularly noticeable in the y-axis, which is incorrect. These results demonstrate the unreliability and inaccuracy which may result through localization using the same feature when motion is not smooth.

In practise, vehicles normally move in a smooth manner by being driven at constant speeds around the environment. In order to obtain localization results for such motion, we drive the OxNav vehicle (JTR) along a given trajectory while each of the sensors is tracking a different extracted and matched RCD. The results for such linear motion are shown in Figure 6.11, where it can be seen that the estimates are a lot smoother and closer to the measured ones. Figure 6.12 also shows the variation of performance with observation model while the sensors track different features. These results show the effect of larger observation noise on the both the observations themselves and the estimates.

To further demonstrate the accuracy and detail which can be achieved consider the following: When making turns on the OxNav vehicle the trajectory can be made to follow a spline which is precisely defined. Therefore, the location of the vehicle is estimated and compared with the trajectory defined by the spline. It can be seen in the results in Figures

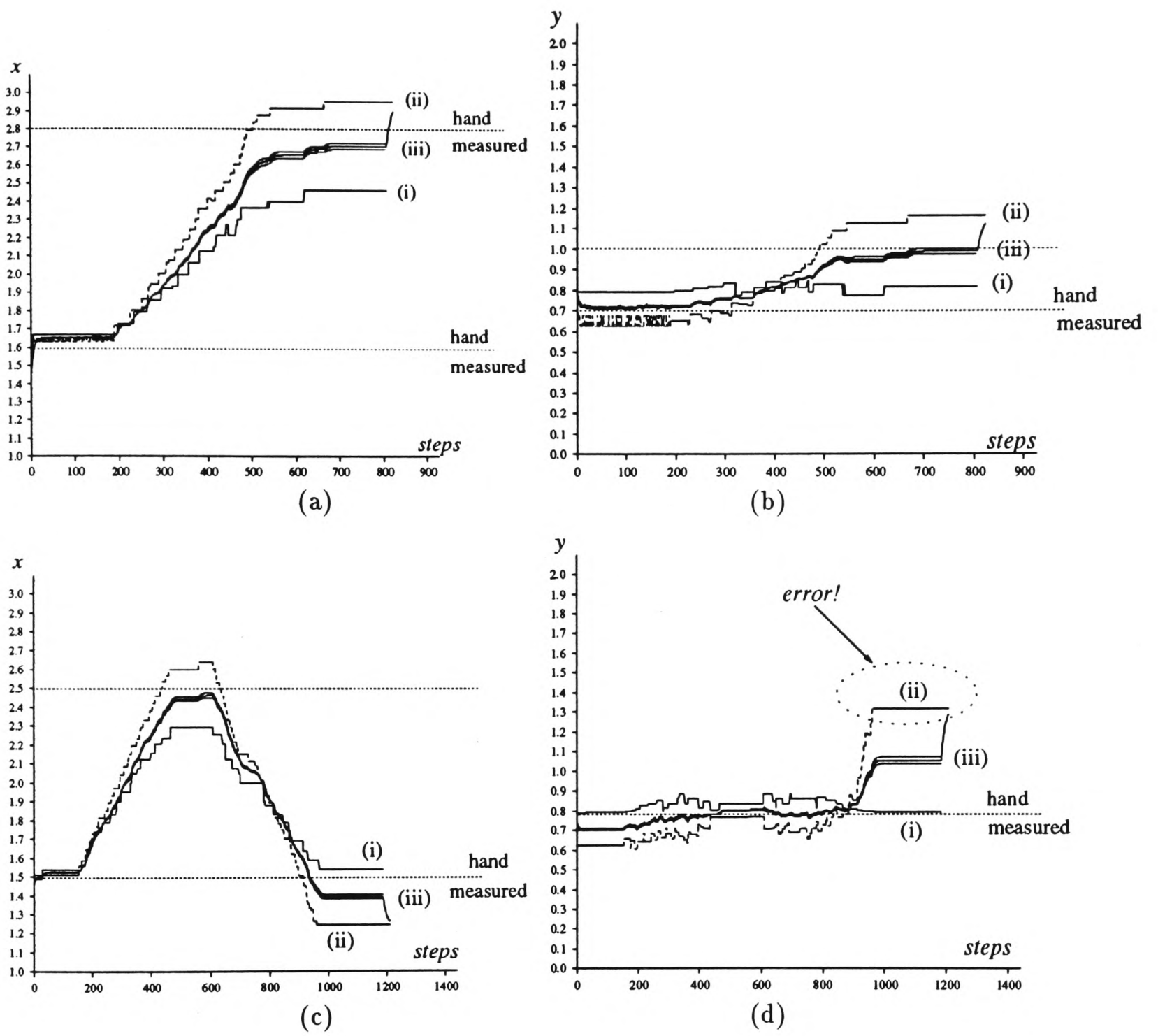


Figure 6.10: Comparing estimates with hand measured values while localizing on the same feature at different ranges. For this motion vehicle motion was not smooth. In (a) and (b) the feature being used for localization is at a range of $1.5m$ and motion is in both x-axis and y-axis. In (c) and (d), the feature is at a range of $4m$ and motion is only in the x-axis. In both, (i) is sensor 1's observations (ii) sensor 2's observations and (iii) estimates. Notice that in (c) and (d) sensor (ii) at time $k = 900$, drifts and starts tracking some other feature causing an error in the estimate.

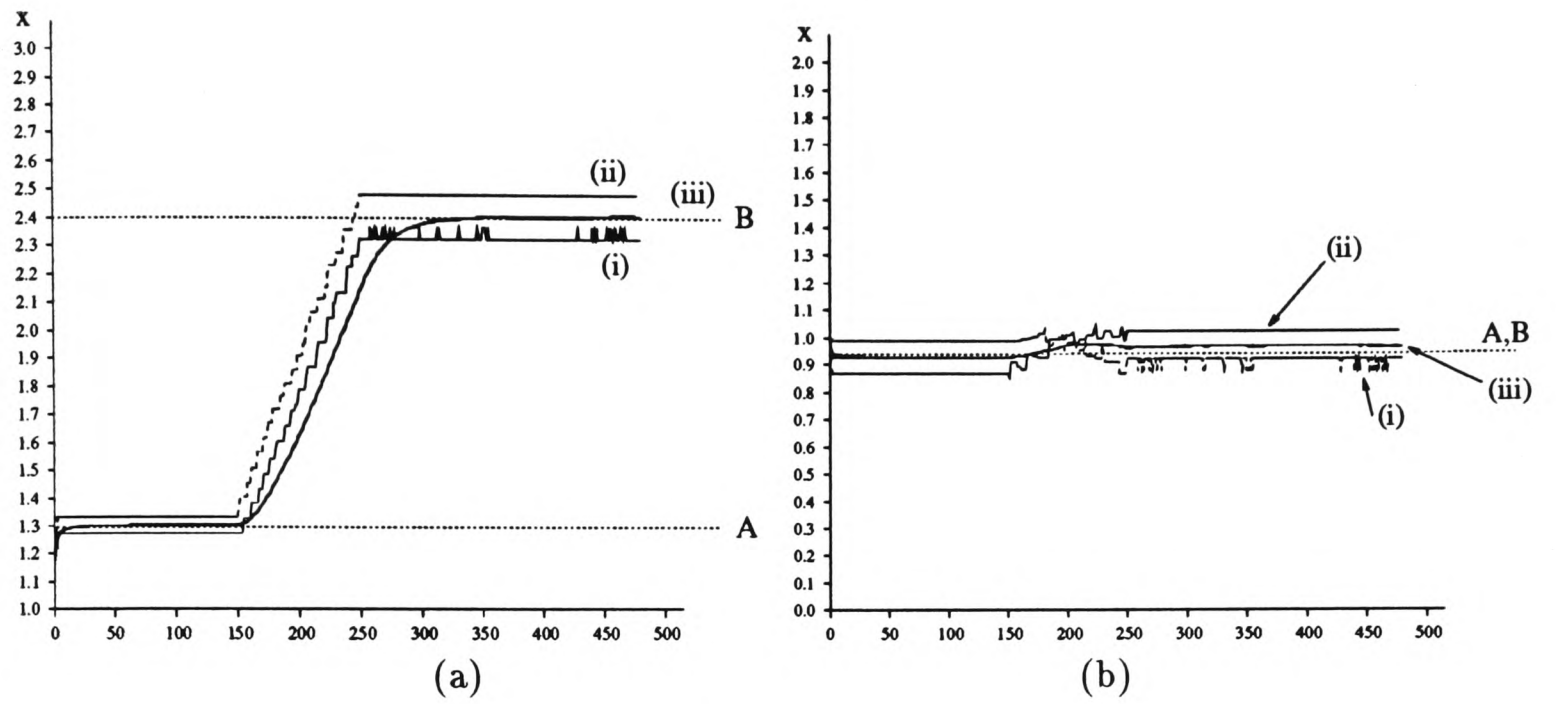


Figure 6.11: Smooth motion on the OxNav Vehicle JTR. Motion is in the x-axis while the sensors track the same feature. (a) and (b) show the estimates together with hand measured values and (c) gives the estimates in a Cartesian coordinate system.

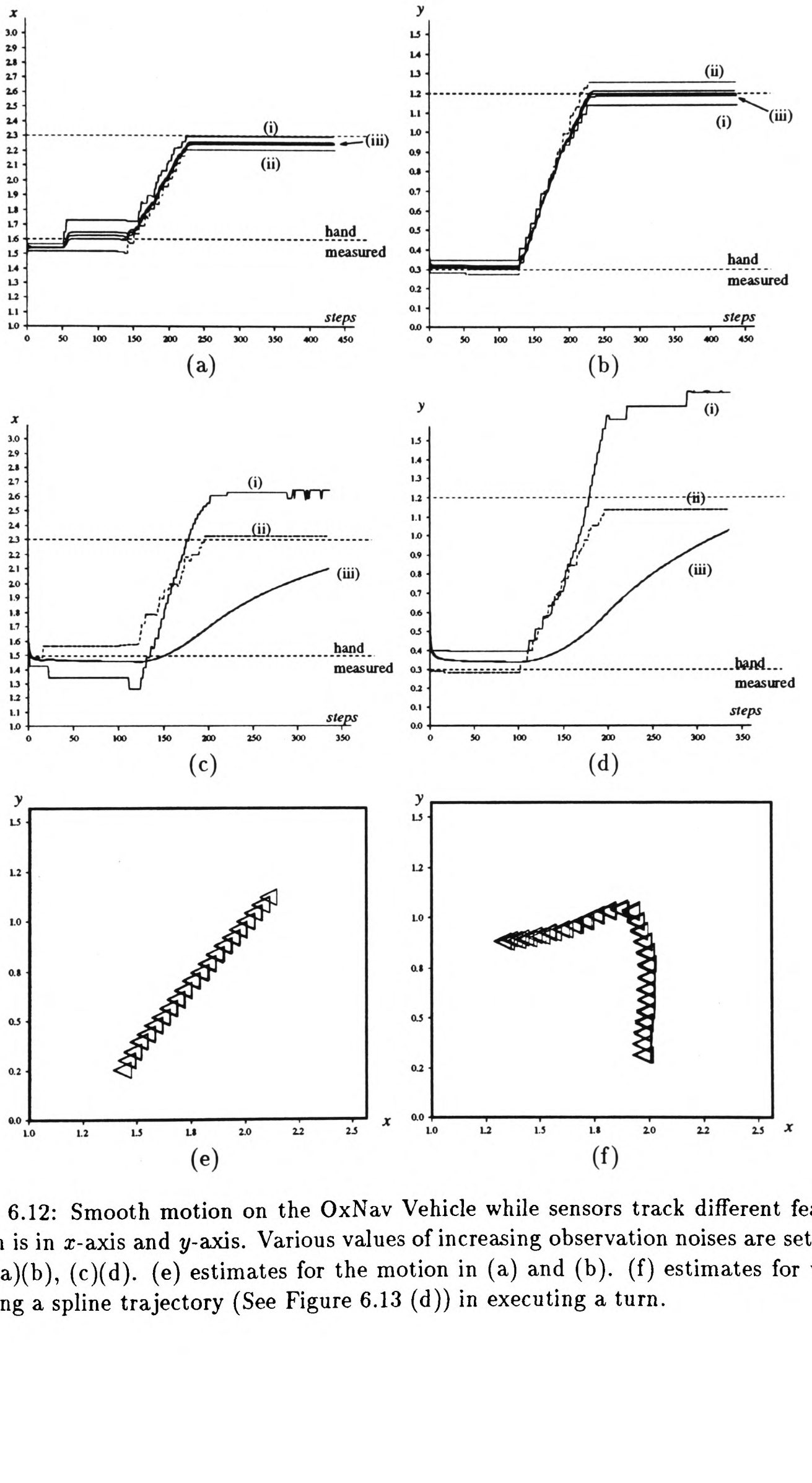
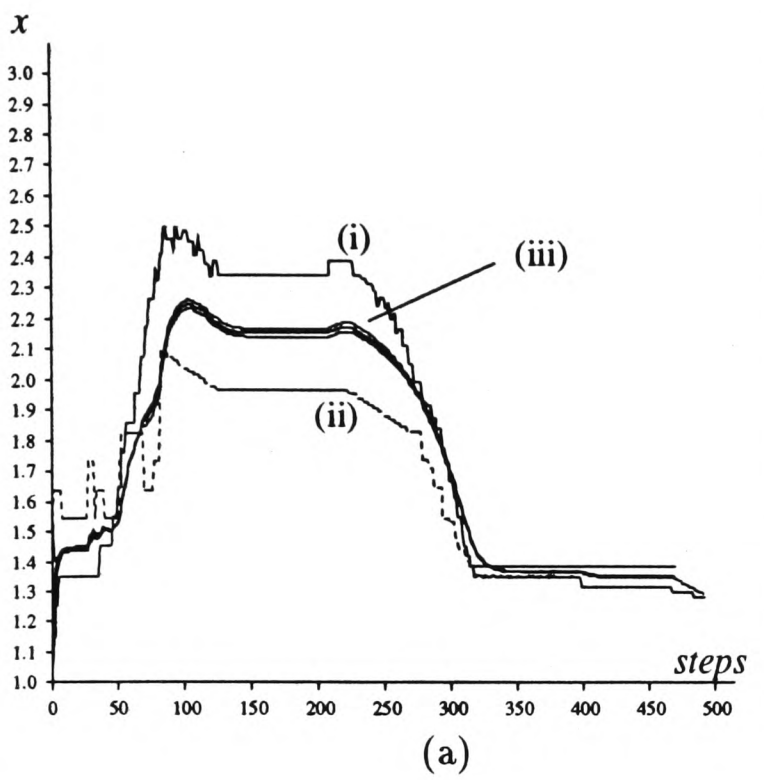
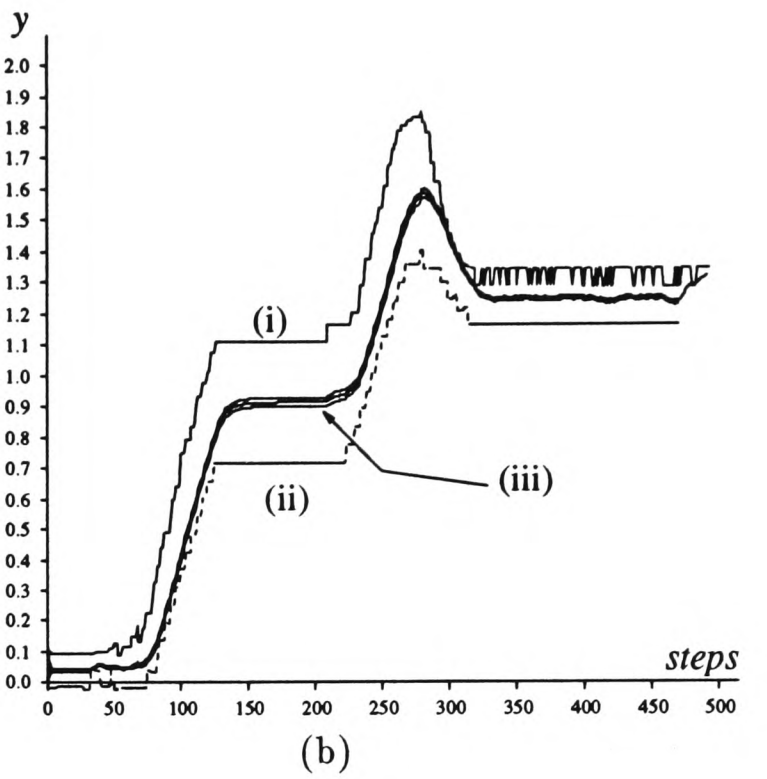
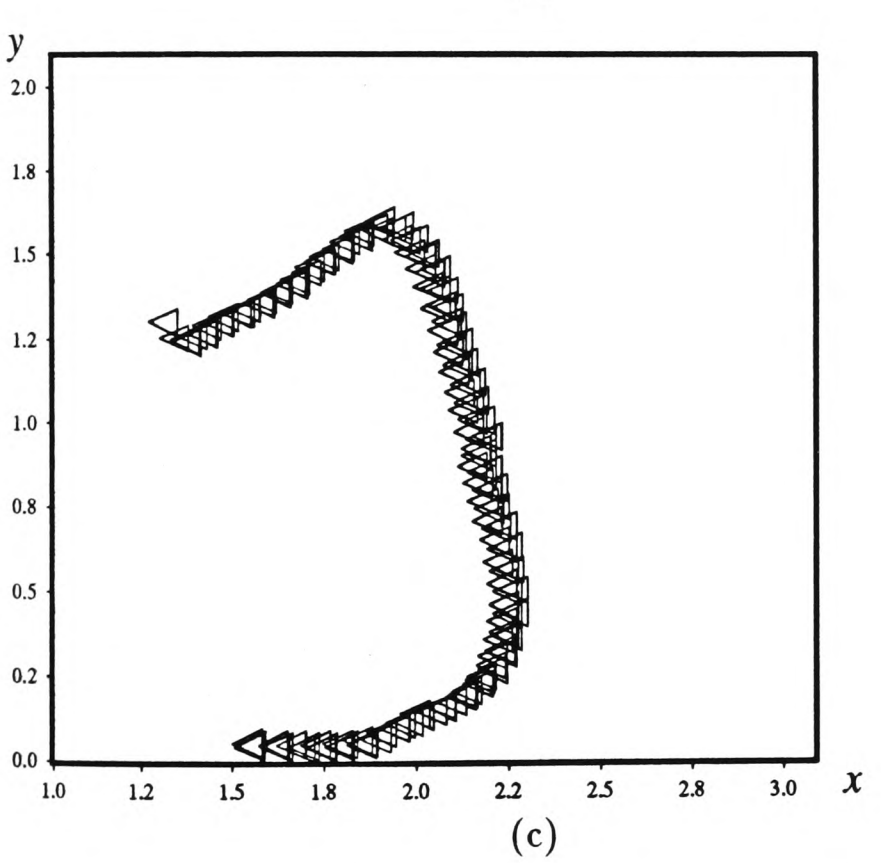


Figure 6.12: Smooth motion on the OxNav Vehicle while sensors track different features. Motion is in x-axis and y-axis. Various values of increasing observation noises are set in the pairs (a)(b), (c)(d). (e) estimates for the motion in (a) and (b). (f) estimates for vehicle following a spline trajectory (See Figure 6.13 (d)) in executing a turn.







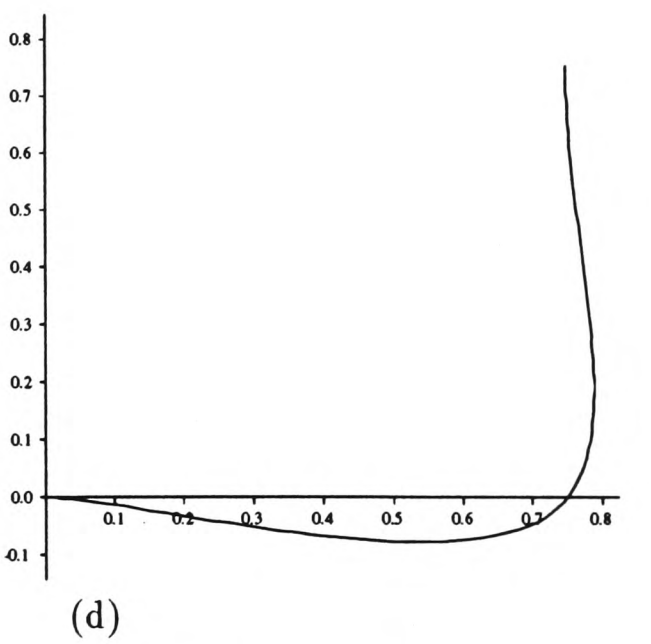


Figure 6.13: Estimates for motion incorporating 2 turns based on a trajectory as defined by a spline for turning a corner in a continuous motion. The estimates are shown in (a) and (b) and (c) shows these in Cartesian coordinates. The trajectory followed by the OxNav vehicle in executing the turns is defined by the spline shown in (d).

6.12 and 6.13 that the estimates closely follow the path as defined by the spline which is plotted in Figure 6.13 (d).

6.3.3 Discussion

The results presented demonstrate the following: Firstly, that the use of two sensors decentrally improves the accuracy obtained by providing estimates obtained by fusing observations from two viewpoints. Secondly, the decentralized algorithm performed better when sensors were tracking different features and this is reflected in the above results. Therefore, a good strategy would be to always ensure that sensors always localize on different features. The importance of these results is that they demonstrate how each of the Tracking Sonars becomes a complete guidance sensor obtaining localization estimates at rates of up to $25Hz$ and can be used together with other similar sensors in a decentralized fashion. In this respect, these are fundamentally new results which greatly improve vehicle guidance. While being a compelling demonstration of this navigation approach, the results do show some limitations; because of the absence of a stringent validation scheme for the Tracking Sonars, when a sensor may drift away (non-deterministically) onto a an adjacent feature and begins to track this feature rather than the one which was initially matched. This was seen in the results of Figure 6.10(c) and (d). However, such limitations can be overcome in the first instance, by better validation and in the second instance, by improved hardware design from which most of the limitations stem (see Appendix C).

Mention has been made of the requirement for a facility for matching extracted RCDs with those predicted from a map. Realization of this has been done in the work for the OxNav project based on the work in [88]. This is the so-called "EKF Black Box" which takes as input map details and observations from the Tracking Sonars and validates these using an

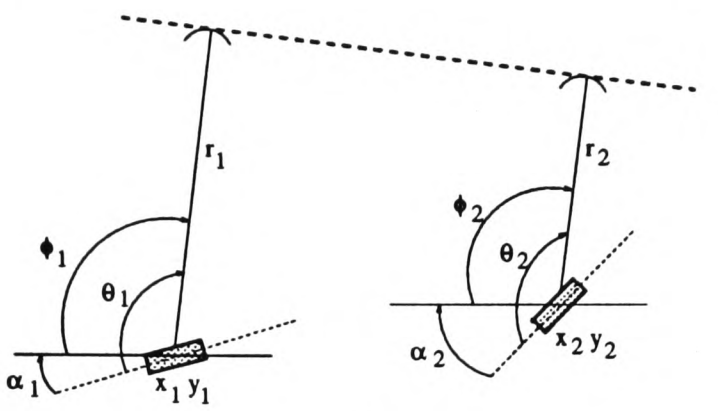
EKF-based validation scheme. Because the Black Box contains the entire map, validation rates are far slower than the data rates attainable with the Tracking Sonars. However, given the scheme outlined in this chapter, this handicap is only encountered at initialization, that is when the sensor scans and extracts RCDs. Subsequent to this, matching is only required when there is a need to re-initialize or to occasionally check the validity of features currently being tracked. These ideas also suggest the possibility of each Tracking Sonar maintaining its own map segment so as to speed up validation.

6.4 Modular Feature Classification

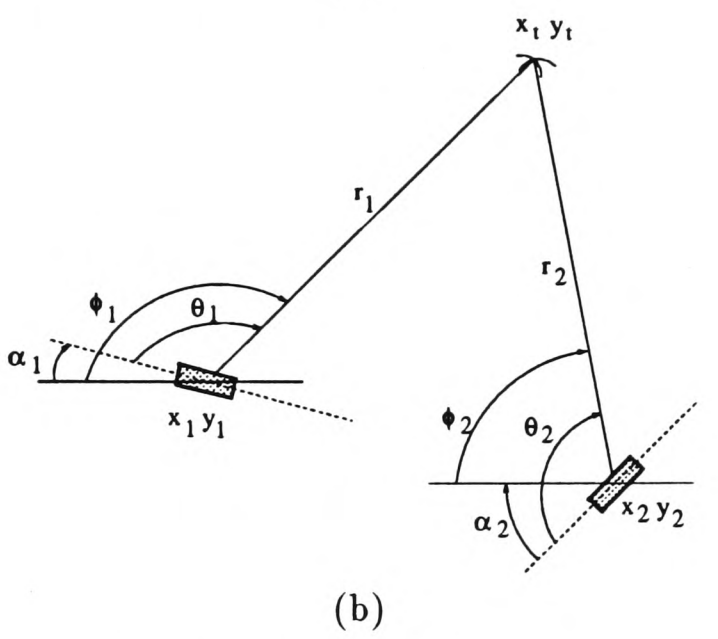
A modular algorithm for classifying features using the Bayesian Classification algorithm of Chapter 3, is now described. The fundamental idea in the algorithm is the use of displacement of the sensor due to motion to differentiate between types of features being observed in the environment. With observations from two displaced points, an algorithm can be developed to differentiate between *lines* and *points* using an approach based on triangulation. In our environment feature model consisting of planes, corners, edges and cylinders, observation of a line suggests a plane and observation of a point suggests an edge, corner and even a cylinder with a small enough radius. We exploit this simple model further.

6.4.1 Classification from Displacement

If the observation is of a line, then the observation is as shown in Figure 6.14 (a). In Figure 6.14 (a), it can be seen that if $|\phi_1 - \phi_2| < \epsilon$, where ϵ is small enough then the observed feature is likely to be a plane. If the observation is as shown in Figure 6.14 (b), and the point (x_{t1}, y_{t1}) coincides with (x_{t2}, y_{t2}) then the observation is likely to be that of a point. This is a qualitative way of differentiating between lines and points. In order to develop a quantitative algorithm, consider Figure 6.15. It can be appreciated from Figure 6.15



(a)



(b)

Figure 6.14: Illustrating a model for observations from two positions (x_1, y_1) and (x_2, y_2). (a) shows the observation of a line from two positions and (b) observation of a point.

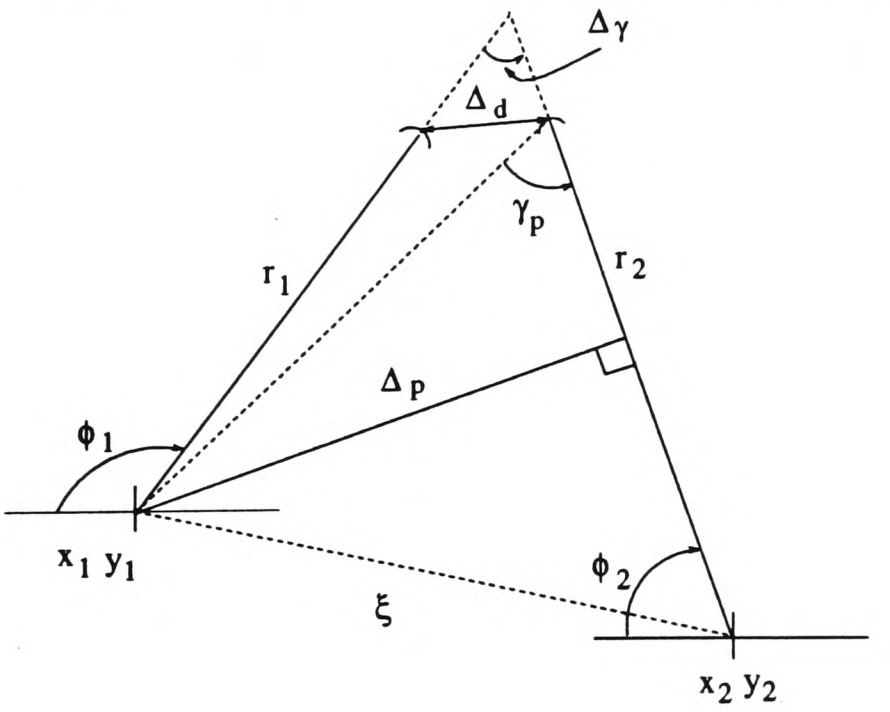


Figure 6.15: Observation model for an arbitrary feature illustrating the parameters used for differentiation. Observations are made from two positions (x_1, y_1) and (x_2, y_2).

that if the ratio $\Delta_d/\Delta_p \rightarrow 0$ then the observation is likely to be that of a line [3] . If the ratio $\Delta_d/\Delta_p \rightarrow 1$, then the observation is likely to be that of a point. This suggests using some thresholding technique in order to make the decision. However, one can do better by introducing "variance" terms as follows

$$\sigma_d^2 \triangleq \left(\frac{\Delta_d - \Delta_p}{\Delta_p}\right)^2 \text{ and } \sigma_\gamma^2 \triangleq \left(\frac{\Delta_\gamma - \Delta\gamma_p}{\Delta\gamma_p}\right)^2. \tag{6.13}$$

In this way, the variance σ_d^2 has values in the interval $[0,1]$ with values close to 1 suggesting a point and values close to 0 suggesting a line. Similarly, σ_γ^2 has values in the interval $[0,1]$, with values close to 1 suggesting a line and values close to 0 suggesting a point. Therefore, the likelihood of the observation being a line is given by

$$p(\mathbf{z}' \mid line) = \sigma_\gamma^2, \tag{6.14}$$

and the likelihood of the observation being a point is given by

$$p(\mathbf{z}' \mid point) = \sigma_d^2, \tag{6.15}$$

where $\mathbf{z}'$ is the set of observations that is $\{(r_1,\theta_1),(r_2,\theta_2)\}$, obtained from the two observation positions (x_1,y_1) and (x_1,y_1). In order to make these into true probabilities, they need to be normalized. In keeping with the notation of the algorithm in Chapter 3, the observation parameter model $\{M\}$ is given by $\{M\} = \{M_1, M_2\}$ where the parameters are $M_1 =$ line and $M_2 =$ point, and so

$$p(\mathbf{z}'(k) \mid M_1) = \frac{\sigma_\gamma^2}{\sigma_\gamma^2 + \sigma_d^2}, \tag{6.16}$$

and

$$p(\mathbf{z}'(k) \mid M_2) = \frac{\sigma_d^2}{\sigma_\gamma^2 + \sigma_d^2}. \tag{6.17}$$

[3] Δs here are not to be confused with the differential Δ, from Chapter 5.

The model relating these parameters to the state vector that is $p(\{M\} \mid \mathbf{x})$, this is given by

$$\begin{bmatrix} p(M_1 \mid X_1), & p(M_1 \mid X_2) \\ p(M_2 \mid X_1), & p(M_2 \mid X_2) \end{bmatrix}, \tag{6.18}$$

where the state is given by

$$\mathbf{x} = [X_1 = plane, X_2 = point]. \tag{6.19}$$

The distinct classification $X_2 = point$, represents features such as corners, edges and cylinders whose radius is small compared to their range from the sensor, and so $point$ = {corner, edge, smallcylinder}. It is evident that it is not possible to distinguish between elements of *point* with the above model. Having decided that the observation is of a point, differentiation between edges, corners and cylinders can be done using visibility angles or matching with environment maps.

6.4.2 Algorithm and Implementation

The algorithm requires knowledge of sensor displacement. Such information can be obtained from various sources depending on the application. In this presentation, the algorithm is run concurrently with the localization algorithm and so the displacement information can be computed from a knowledge of the estimated location of the sensor.

Observation

The sensor makes an observation $\mathbf{z}(k) = [r_1, \theta_1]^T$ and the classification algorithm obtains the location vector $\mathbf{x}(k) = [x_1, y_1, \alpha_1]^T$ from the localization algorithm (see Section 6.2). We define a minimum displacement ξ which gives the minimum base-line used in Figure 6.15. Using the geometry of Figure 6.15, the following parameters are computed; $\{\Delta_d, \Delta_p, \Delta_\gamma, \Delta_{\gamma_p}\}$. And using Equations 6.13 and 6.13, σ_d^2 and σ_γ^2 are computed.

Likelihood Computation

From Equations 6.16 and 6.17, the likelihood vector $\Lambda_{\mathbf{z}(k)}$, is computed as

$$\Lambda_{\mathbf{z}(k)} = p(\mathbf{z}(k) \mid \mathbf{x}) = [p(\mathbf{z}(k) \mid \mathbf{M_1}), p(\mathbf{z}(k) \mid \mathbf{M_2})] \begin{bmatrix} p(M_1 \mid X_1) & p(M_1 \mid X_2) \\ p(M_2 \mid X_1) & p(M_2 \mid X_2) \end{bmatrix}. \quad (6.20)$$

Classification

The posterior distribution (or belief vector) is given by; $p(\mathbf{x} \mid \mathbf{Z}^k) = [p(X_1 \mid \mathbf{Z}^k), p(X_2 \mid \mathbf{Z}^k)]$, where for each X_b the posterior is computed as

$$p(X_b \mid \mathbf{Z}^k) = p(X_b \mid \mathbf{Z}^{(k-1)'}) \, [\alpha p(\mathbf{z}(k) \mid X_b)], \quad (6.21)$$

where $(k-1)'$, refers the previous time step when the posterior was updated, the frequency of which depends on ξ, as opposed to $(k-1)$ which refers to the previous time that the localization algorithm provided location estimates. As discussed in Chapter 3, the inferred classification is given by

$$\hat{X} = \arg\max[p(X_1 \mid \mathbf{Z}^k), \; p(X_2 \mid \mathbf{Z}^k)]. \quad (6.22)$$

In the implementation of the above algorithm, the step ξ determines the minimum acceptable displacement which is used for classification. Therefore, if no motion is detected, that is, $\mid x_1 - x_2 \mid < \xi$ and $\mid y_1 - y_2 \mid < \xi$, then no updating occurs. It is important when obtaining the displacement information from the localization algorithm of Section 6.2, that the value of $\xi \not\approx 0$ because often the estimate is noisy and so would give a false impression of displacement in a situation where there has been no motion. Such a situation is illustrated in the localization results of Figure 6.3.

6.4.3 Results and Performance

We now present some results using the feature classification algorithm. In the following results, the sensor platform moves while observing a feature which is to be classified as either a plane or a point feature.

For the first set of results, the displacement information is very accurate, having been estimated from linear and very smooth motion on the OxNav vehicle (See Figure 6.11 to 6.13) and actual measurement data as observations. In the results shown in Figure 6.16, the following models were used for $p(\{M\} \mid \mathbf{x})$; in (a) and (c);

$$\text{model 1}: \quad p(\{M\} \mid \mathbf{x}) = \begin{bmatrix} p(M_1 \mid X = line) & p(M_1 \mid X = point) \\ p(M_2 \mid X = line) & p(M_2 \mid X = point) \end{bmatrix} = \begin{bmatrix} 0.82 & 0.15 \\ 0.18 & 0.85 \end{bmatrix}, \tag{6.23}$$

and in (b) and (d)

$$\text{model 2}: \quad p(\{M\} \mid \mathbf{x}) = \begin{bmatrix} 0.52 & 0.45 \\ 0.48 & 0.55 \end{bmatrix}. \tag{6.24}$$

The probabilities are updated whenever motion with a minimum displacement ξ, of $0.01m$ was detected. The displacement $\xi = 0.01m$, was chosen in order to give an appreciable baseline. However, if the sonar data could be obtained precisely without any uncertainty and absolute displacement information could be obtained, then a smaller value of ξ could be used, a result which can be verified by simulation [96].

The above was repeated for the case where displacement information is obtained from motion which is jerky and not necessarily linear. In the following results, the minimum displacement ξ was varied from $0.02m$ to $0.04m$ to illustrate the effect on the classification performance. In addition, the two models of $p(\{M\} \mid \mathbf{x})$ given in Equation 6.24 were used to illustrate their effect on classification using different values of ξ. In the results shown in Figure 6.17 and 6.18, it can be seen that the larger minimum displacement $\xi = 0.04m$

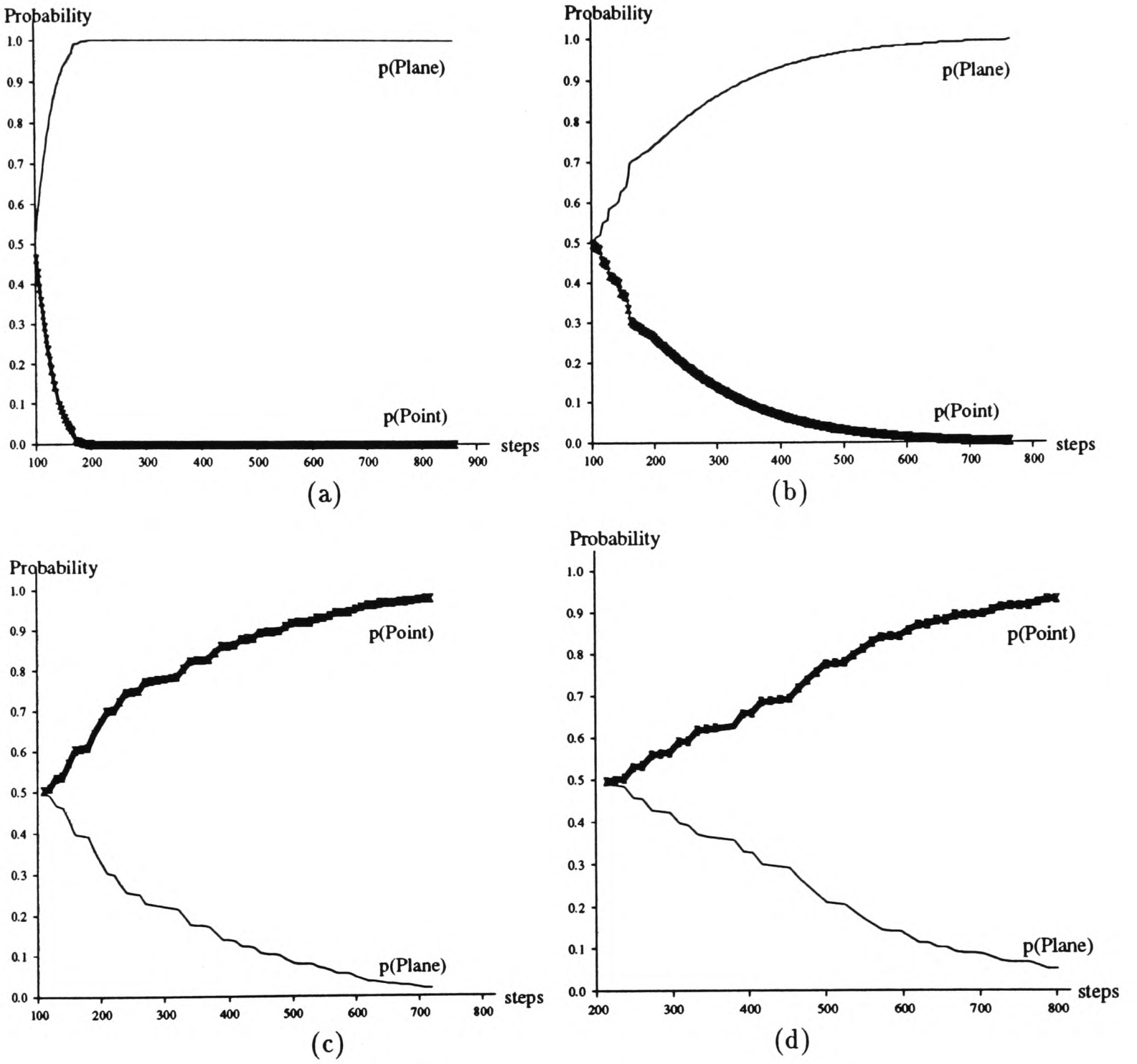


Figure 6.16: Probabilities for each feature type using very accurate displacement information (from very smooth linear motion). In (a) and (b) the feature is a plane. In (c) and (d) the feature is an edge. (a) and (c) use the same values in the model $p(\{M\} \mid \mathbf{x})$ and similarly (b) and (d).

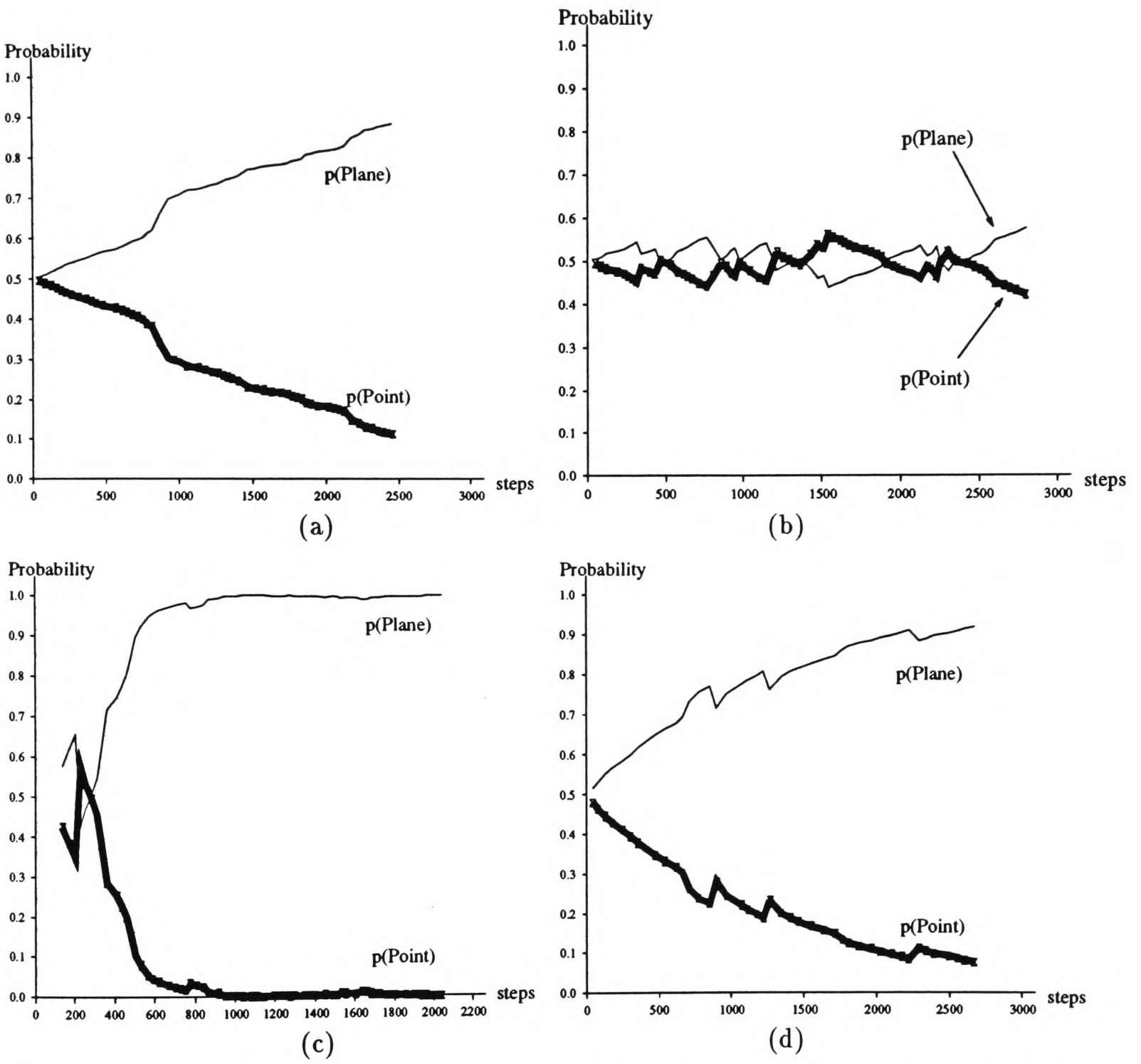


Figure 6.17: Classifying a plane using two models of $p(\{M\} \mid \mathbf{x})$ and two values for the minimum displacement ξ. In (a) and (b), the minimum displacement $\xi = 0.02m$ and in (c) and (d) $\xi = 0.04m$. In (a) and (c), model 1 is used for $p(\{M\} \mid \mathbf{x})$ and in (b) and (d) model 2 is used.

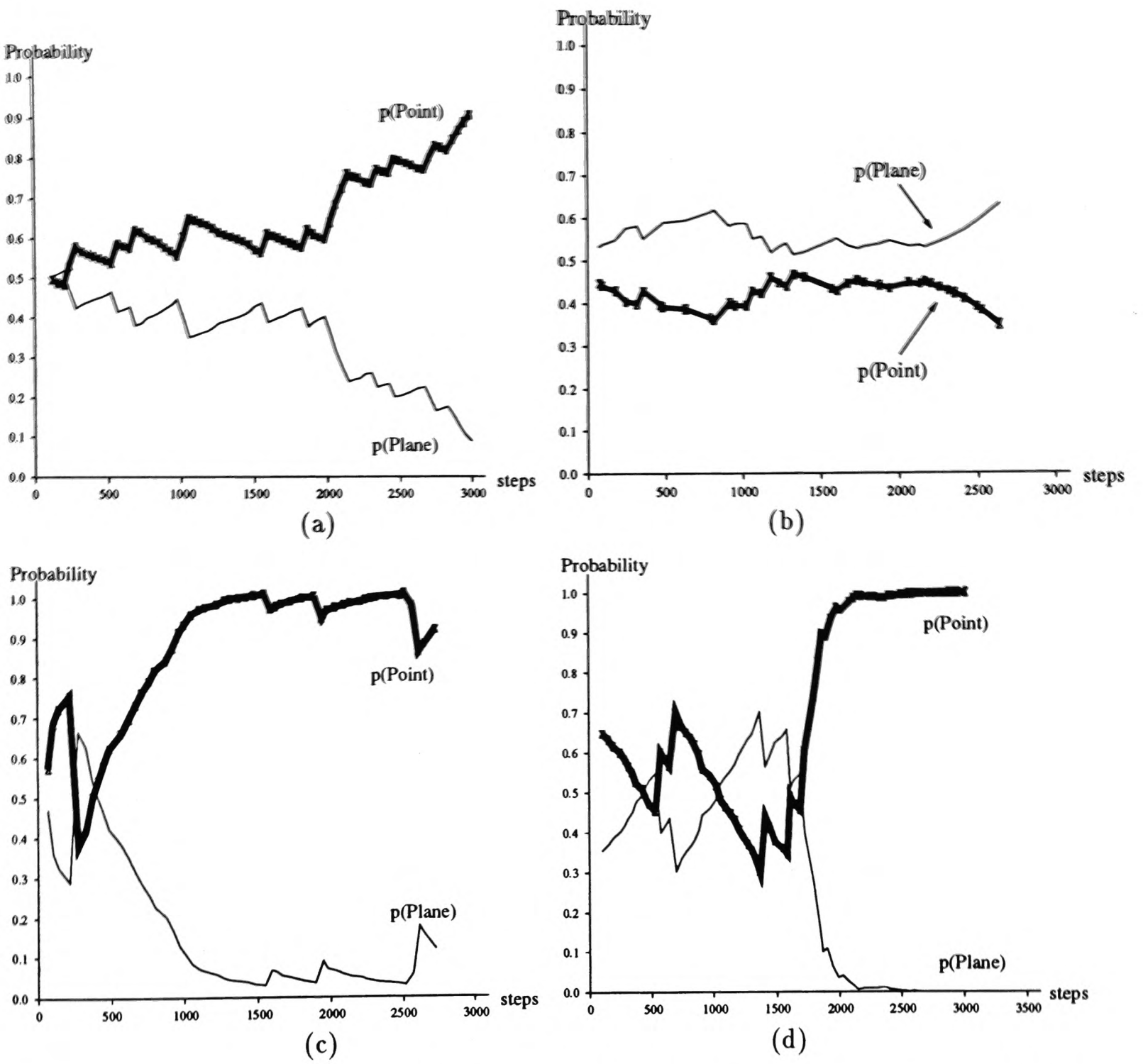


Figure 6.18: Classifying an edge using two models of $p(\{M\} \mid \mathbf{x})$ and two values for the minimum displacement ξ. In (a) and (b), the minimum displacement $\xi = 0.02m$ and in (c) and (d) $\xi = 0.04m$. In (a) and (c), model 1 is used for $p(\{M\} \mid \mathbf{x})$ and in (b) and (d) model 2 is used. In (b) the edge is incorrectly identified as a plane.

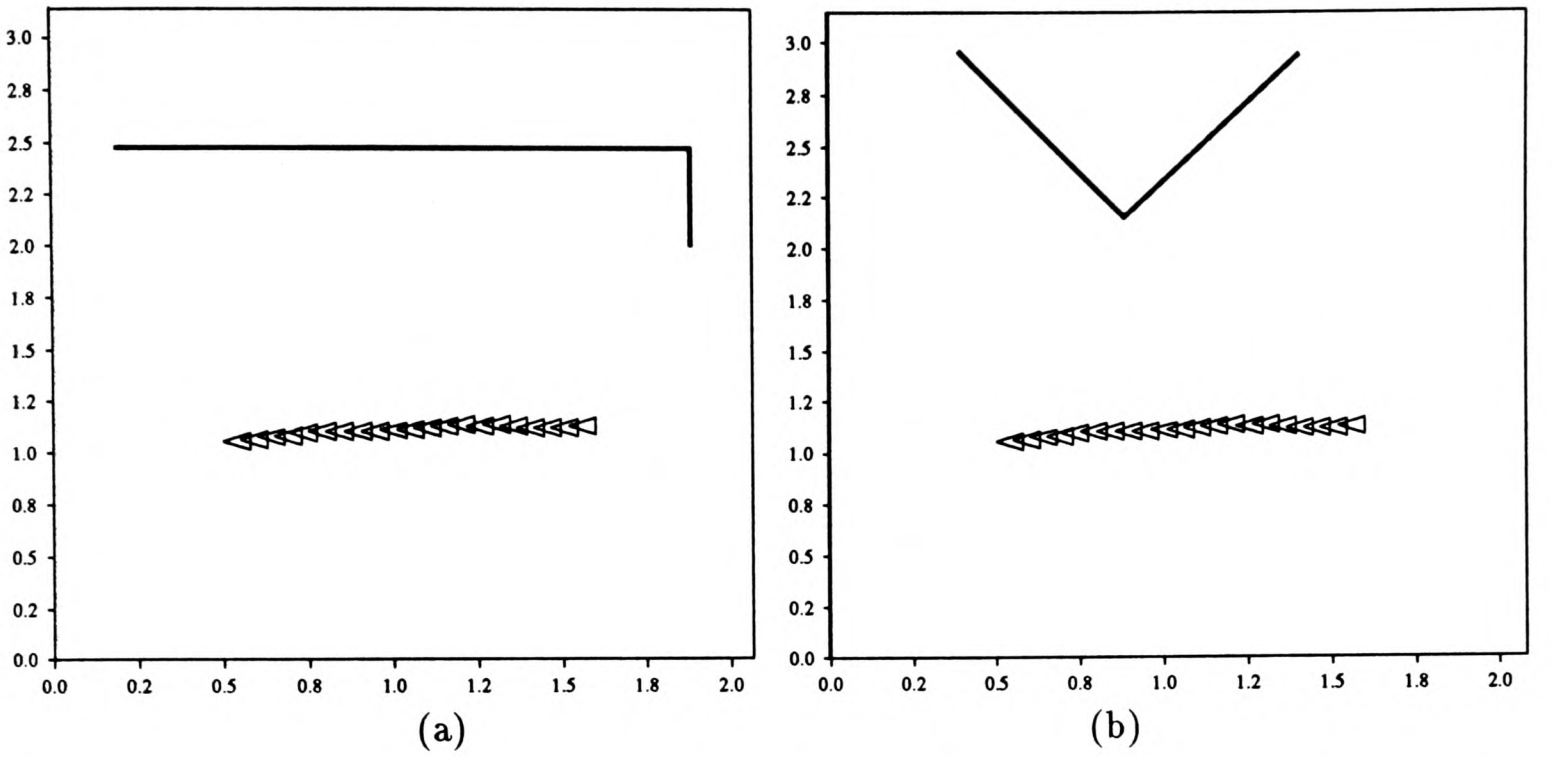


Figure 6.19: Location estimates during classification of (a) plane (b) edge. Note that in these estimates, only positional changes of $0.04m$ are shown.

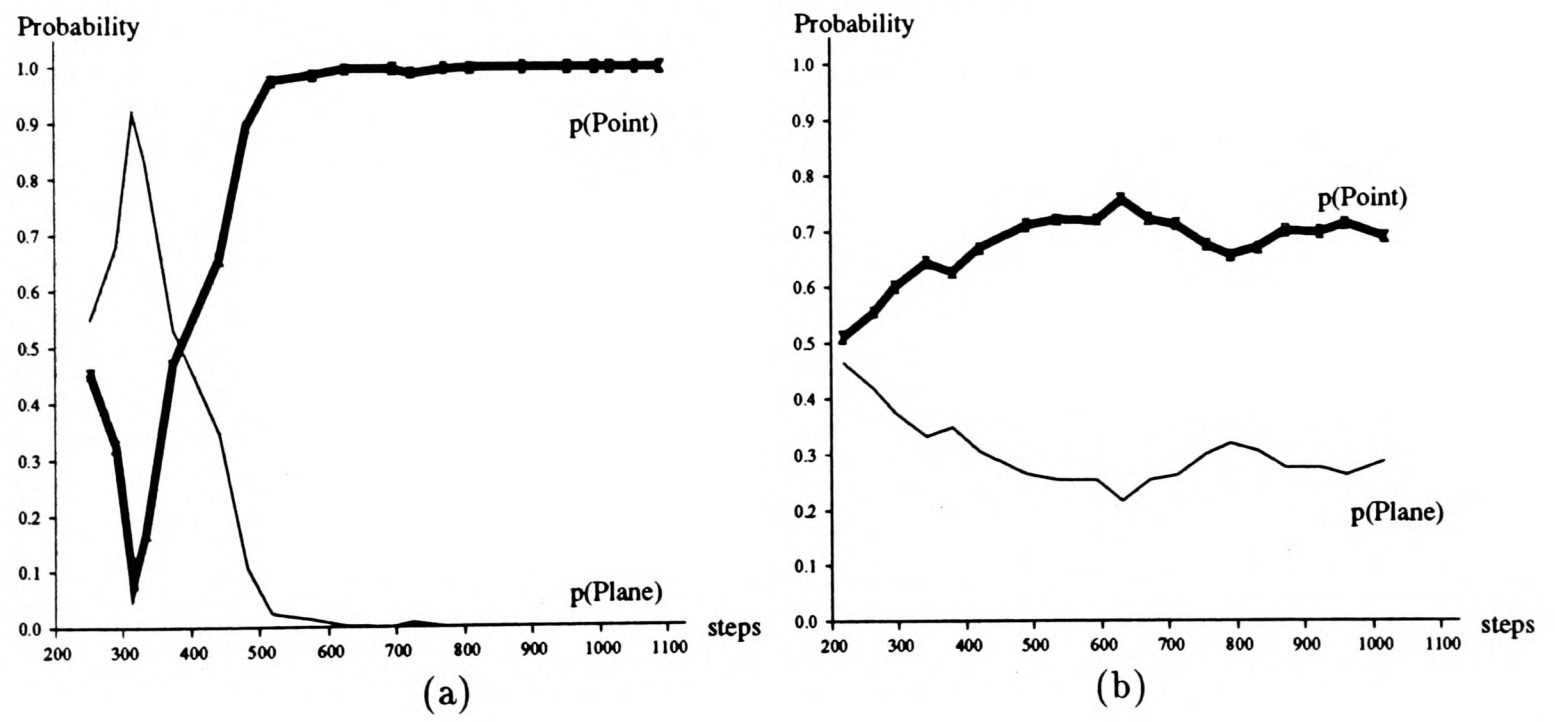


Figure 6.20: Classifying a corner feature using in (a) model 1 and in (b) model 2 for the likelihood $p(\{M\} \mid \mathbf{x})$.

results in correct classification for both the plane and the edge, using either model 1 or 2 for $p(\{M\} \mid \mathbf{x})$. Correct classification with $\xi = 0.02m$ is only obtained using model 1. When $\xi = 0.02m$ is used together with model 2, the classification for the plane is uncertain as shown in Figure 6.17 (b) and the edge is classified incorrectly as a plane as shown in Figure 6.18(b). This can be explained by the fact that the displacement data used in the algorithm derives from the localization algorithm and the degree of instability of this data is dependent on the models used in the localization algorithm and also in the uncertainty in the data observed as discussed in Chapter 5. This is not quite as evident from the plot of the sensor's motion in Figure 6.19 which only shows positional changes of at least $0.04m$. These results highlight the sensitivity of the classification algorithm to the accuracy of the input displacement information. These results can be shown for other types of features as shown in Figure 6.20 for a corner.

Optimum performance of the classification algorithm can be obtained through prudent setting of the minimum displacement ξ taking into account the uncertainty in the displacement data. Knowledge of this uncertainty can be obtained from the observation model used in the localization algorithm which in turn depends on the Tracking Sonar parameters as discussed in Chapter 5. With the correct setting of these parameters, reliable classification can be obtained within the operating range $0.40 - 4.00m$ with values of $\xi > 0.03m$. Setting the value of ξ very high, while providing a larger base-line, is not desirable because then the algorithm is not updated frequently and so distinct classifications are only obtained after considerable periods assuming the vehicle even moves that far. Therefore, the value of ξ in addition to localization performance, should also take into account the anticipated vehicle displacement and the urgency of the classification information in a given application.

6.5 Summary

The work presented in this chapter demonstrates the improvements to be gained from navigating using decentralized data fusion methods based on Tracking Sonars. The previously significant overhead of maintaining map contact at all times through validation of measurements every processing cycle is eliminated. Hence cycle time is greatly increased with estimates obtained at approximately $25Hz$. Previously published "state of the art" estimates are of the order of an estimate every $0.5s$ [90] at best. In addition, the implementation of localization capabilities at the sensor level frees up the central system on a robot to perform other higher level functions such as path planning *et cetera.* The localization algorithms are augmented by the ability to classify features being tracked. This classification is limited to plane and point features and requires further development making use of information such as visibility angles *et cetera* for further refine classification. It is important to emphasize that the implementations presented here are not intended as robust and fully functional navigation systems, but as demonstrations of new methods for navigation.

The algorithms that have been implemented in this chapter are not limited to vehicle navigation applications only. Indeed if the whole localization algorithm presented is reversed, the system then describes the tracking of dynamic targets by sensors in known locations. This being the usual application of tracking algorithms normally encountered in the literature [12][19]. This being very similar to the work presented by Rao [115] using CCD cameras. This chapter thus provides us with a basis for sensor management demonstrations which are the subject of the next chapter.

Chapter 7

Sensor Management Demonstrations

7.1 Introduction

This chapter presents results obtained from implementing the sensor management strategies developed in Chapter 4. It is generally acknowledged that demonstrating implemented sensor management activity is something of a challenge [114]. Demonstrating the advantages of having sensor management is made difficult by its feedback nature in the data fusion loop as illustrated in Figure 1.1. One method which can be employed however, is to show the quantities on which management decisions are based and the decisions themselves for a variety of situations. In addition, such results must also show that the management actions adopted are optimal given the management imperative. In our context, this amounts to showing that the resulting management decisions always maximize information as discussed in Chapter 4.

The results that we present are based on the implementation of vehicle localization using Tracking Sonars mounted on the OxNav vehicle (JTR) as described in the previous chapter. These results serve to show how the sensors in the decentralized scheme illustrated in Figure 6.7 can be managed. For localization, the information metrics on which management is based are with respect to the state $\mathbf{x}$ which represents the location of the vehicle. We also present management results from the implementation of the feature classification

algorithm as described in Chapter 6. For this algorithm, the information metrics are with respect to the state **x** which is the classification of a particular feature being tracked which localizing. Therefore given our discussion in Chapter 4, we can state from the onset that the management of these two algorithms cannot be coupled because the states refer to two distinct entities as described by **Case 2** in Section 4.7.2.

The quantities used for management are the information metrics introduced in Chapter 4, which we summarize here:

- **Global and partial posterior information.** When implemented on a single sensor, the posterior information metric is given by Equations 4.25 and 4.35, that is

$$\begin{aligned} \mathbf{I}(k) &= \frac{1}{2} ln\ [(2\pi e)^n \mid \mathbf{P}(k \mid k) \mid], && \text{for the information filter,} \\ &= \sum_b p(X_b \mid \mathbf{Z}^k)\ ln\ \left[p(X_b \mid \mathbf{Z}^k)\right], && \text{for the classification.} \end{aligned}$$

  For a sensor i in a decentralized system, there is a global posterior information metric given by Equations 4.31 and 4.39), that is

$$\begin{aligned} \mathbf{I}_i(k) &= -\frac{1}{2} ln\,[(2\pi e)^n \mid \mathbf{P}_i(k \mid k) \mid], && \text{for the information filter,} \\ &= \sum_b p(X_{ib} \mid \{\mathbf{Z}^k\})\ ln\ \left[p(X_{ib} \mid \{\mathbf{Z}^k\})\right], && \text{for classification,} \end{aligned}$$

  and a partial information metric given by Equation 4.33 and 4.40, that is

$$\begin{aligned} \tilde{\mathbf{I}}_i(k) &= -\frac{1}{2} ln\left[(2\pi e)^n \mid \tilde{\mathbf{P}}_i(k \mid k) \mid\right], && \text{for the information filter,} \\ &= \sum_b p(X_{ib} \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k))\ ln\ \left[p(X_{ib} \mid \{\mathbf{Z}^{k-1}\} \cup \mathbf{z}_i(k))\right], && \text{for classification.} \end{aligned}$$

- **Observation information.** The observation information is given by the information in the likelihood, (Equations 4.30 and 4.38), that is,

$$\mathbf{i}(k) = -\frac{1}{2}\ ln\ [(2\pi e)^n \mid \mathbf{R}[k, \mathbf{z}(k)] \mid], \qquad \text{for the information filter,}$$

$$= \sum_{b} \alpha\Lambda_{\mathbf{z}(k)}(X_b)\, ln\, \left[\alpha\Lambda_{\mathbf{z}(k)}(X_b)\right], \quad \text{for classification.}$$

For a sensor i in a decentralized system this metric is given by $\mathbf{i}_i(k)$ from Equations 4.34 and 4.41. Notice that for the information filter, the observation information is dependent on a covariance which is dependent on the observations themselves, as a result of our modelling of the Tracking Sonar in Chapter 5.

7.2 Quantities for Sensor Management

7.2.1 State Estimation Metrics

The information obtained by a sensor depends on its ability to take measurements and the uncertainty associated with these measurements. For the Tracking Sonar, the uncertainty model is modelled by the covariance of the measurement noise given by $\mathbf{R}[k, \mathbf{z}(k)]$. In Chapter 5, we showed that the variances in $\mathbf{R}[k, \mathbf{z}(k)]$, both depend on the measured quantities themselves as well as on the sensor operational parameters $\zeta = (\sigma_\delta^2, \sigma_u^2, \psi)$. The choice of these parameters depends on the tracking (fixating) performance required. For a fixed range and orientation, and known parameters ζ, the variances σ_r^2 and σ_θ^2 can be approximated based on the analysis of Chapter 5. However, it was also stated that these variances are modified by factors which depend on the values of r and θ themselves as modelled by $g_r(r)$ and $g_\theta(\theta)$. Hence the covariance $\mathbf{R}[k, \mathbf{z}(k)]$ can be considered as a *location covariance*, whose trace gives actual (modified) the variances of r and θ respectively. Therefore, for a given set of operating parameters and a known operation range, the variances used in Equation 5.22, that is

$$\mathbf{R}[k, \mathbf{z}(k)] = \begin{bmatrix} \sigma_r^2(\zeta) \times g_r(r) & 0 \\ 0 & \sigma_\theta^2(\zeta) \times g_\theta(\theta) \end{bmatrix},$$

can be approximated. In the results that follow, we make use of three parameter settings, which give the approximate trace variances of the location covariance as follows; (i) $\sigma_\theta^2 =$

$\sigma_r^2 = 0.0001$, (ii) $\sigma_\theta^2 = \sigma_r^2 = 0.001$ and (iii) $\sigma_\theta^2 = 0.005$ and $\sigma_r^2 = 0.01$. We show in the following results, how the information metrics $\mathbf{I}(k)$, $\mathbf{i}(k)$ vary under different conditions. It must be noted that, because entropy for continuous distributions is relative [39], the absolute values of the information measures $\mathcal{I}(\cdot) = -h(\cdot)$, are not significant only their relative values matter.

In the following results, the Tracking Sonar is functioning as a self-contained guidance sensor using the algorithm outlined in Chapter 6 to estimate its location in an environment.

Variation with Range. Firstly, we show in Figure 7.1, how the information quantities vary with range and the effect that different location covariance models have on the metrics and hence management decisions. The motion of the sensor is as shown in Figure 7.1(a). In (b), it can be seen that the posterior information $\mathbf{I}(k)$, decreases slightly as the range is increased. This is because the information contained in each observation $\mathbf{i}(k)$ shown in (c) decreases with range. In (d) the cumulative observation information $\sum_k \mathbf{i}(k)$ is plotted. This shows how information gained continues to grow but at a slightly reduced rate. The effect of the different models can be seen quite clearly. A model which reduces the variance of the measured quantities results in relatively higher quantities of posterior and observation information $\mathbf{I}(k)$ and $\mathbf{i}(k)$ respectively. Plotting the cumulative information gain is useful because it shows trends in the way the information is gained. The usefulness of this shall be shown in later results.

Variation with Field of View. In Figure 7.2, we show how the information quantities vary across the field-of-view by moving the sensor across in front of a tracked feature as shown in (a). It can be seen in (b) that the information $\mathbf{I}(k)$ increases from the start position as the sensor moves towards the feature and starts to decrease as the sensor moves away.

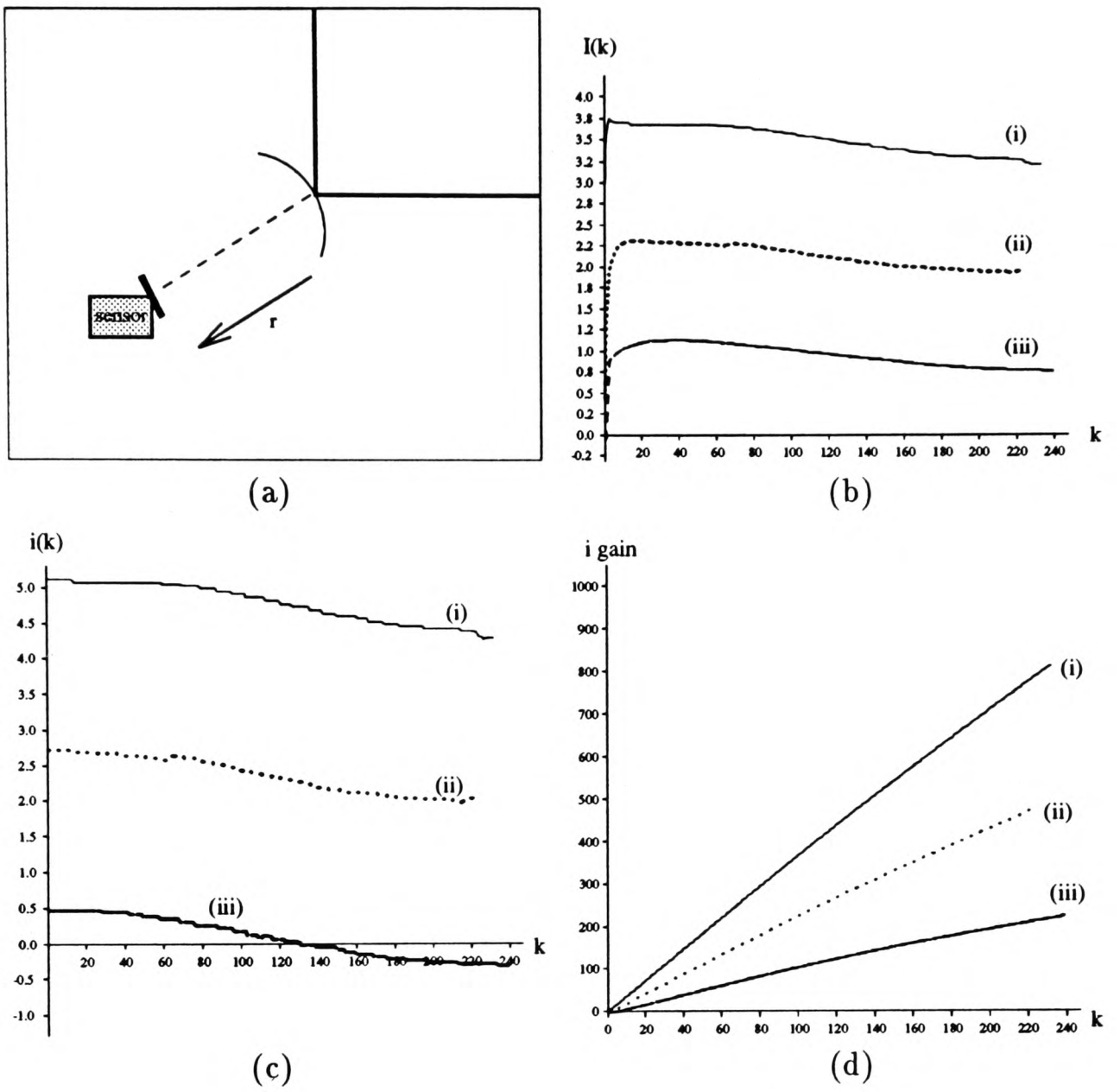


Figure 7.1: Variation of information quantities with range for various observation models. The graphs show in (b) global information $\mathbf{I}(k)$, in (c) relative observation information $\mathbf{i}(k)$ and (d) observation information gain. For (i) $\sigma_\theta^2 = \sigma_r^2 = 0.0001$ (ii) $\sigma_\theta^2 = \sigma_r^2 = 0.001$ and (iii) $\sigma_\theta^2 = 0.005$ $\sigma_r^2 = 0.01$.

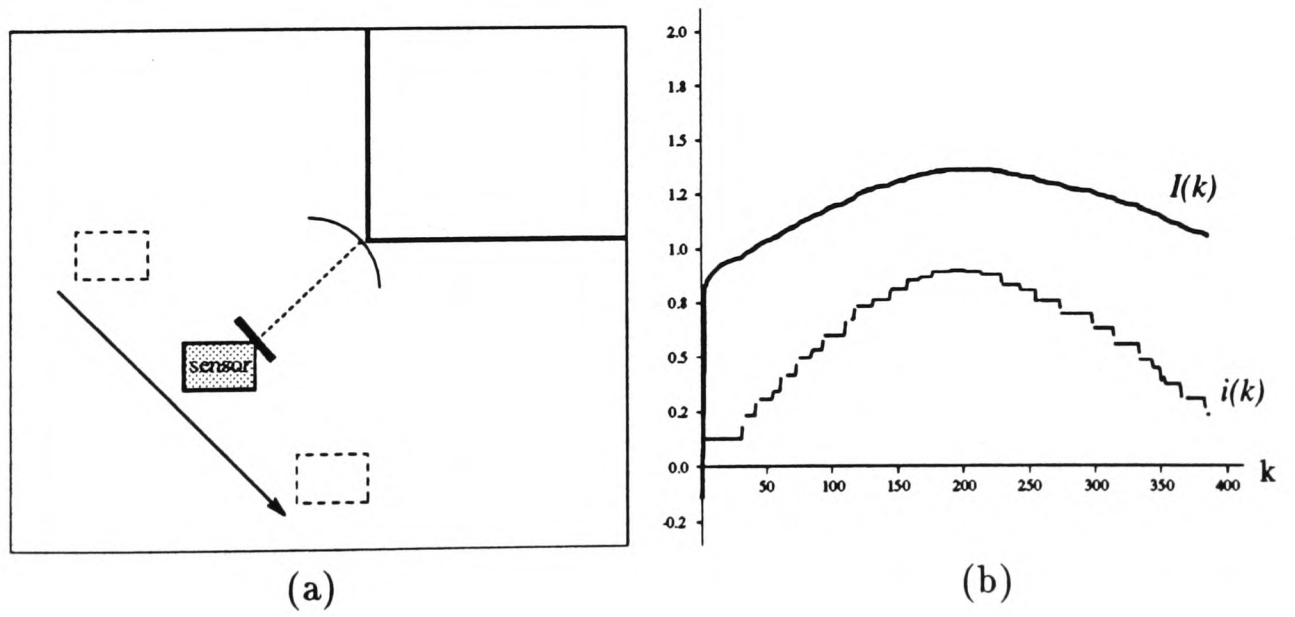


Figure 7.2: Information quantities for motion past a tracked feature. (a) shows the motion past the feature and (b) the global information $\mathbf{I}(k)$ and the observation information $\mathbf{i}(k)$ for $\sigma_\theta^2 = 0.005$ $\sigma_r^2 = 0.01$.

This is due to the observation information $\mathbf{i}(k)$ being greatest when the sensor is closest to the tracked feature as shown in Figure 7.2(c). As before, the relative levels depend on the location covariance.

The results of Figures 7.1 and 7.2 show that the information obtained by the sensors functioning as a self-contained guidance sensors, depends on the observation model which in turn is related to the sensor operational parameters and the relative spatial locations between the tracked feature and the sensor.

7.2.2 Classification Metrics.

We now consider the information quantities when classifying features using the algorithm of Chapter 3. The algorithm takes as input the observation model $p(\{M\} \mid \mathbf{x})$ and vehicle location information. Firstly, the parameter model $p(\{M\} \mid \mathbf{x})$ relates the parameters $\{M\}$ to the state and thus incorporates the uncertainty inherent in the geometrical interpretation of the observation model. Secondly, implemented as in Chapter 6, the classification algorithm makes use of location information from the localization algorithm, to measure displacement. In this regard we defined a displacement step ξ to be used as the minimum displacement for the update of the classification algorithm. Since the location information is not precise, the results of Chapter 6 showed that a larger rather than smaller step is desirable as it limits the effect of the location uncertainty. In the following results, we vary both these factors and examine their effect on management.

Variation with Parameter Model. In the following results we make use of the same parameter models as in Chapter 6 that is

$$\text{model 1}: \quad p(\{M\} \mid \mathbf{x}) = \begin{bmatrix} p(M_1 \mid X = line) & p(M_1 \mid X = point) \\ p(M_2 \mid X = line) & p(M_2 \mid X = point) \end{bmatrix} = \begin{bmatrix} 0.82 & 0.15 \\ 0.18 & 0.85 \end{bmatrix},$$

$$\text{model 2}: \; p(\{M\} \mid \mathbf{x}) = \begin{bmatrix} 0.52 & 0.45 \\ 0.48 & 0.55 \end{bmatrix} \text{ and model 3}: \; p(\{M\} \mid \mathbf{x}) = \begin{bmatrix} 0.65 & 0.31 \\ 0.35 & 0.69 \end{bmatrix} \quad (7.1)$$

where model 3 is added for illustrative purposes. In the results of Figure 7.3, we show the information quantities $\mathbf{I}(k)$ and $\mathbf{i}(k)$ obtained as the sensor moves linearly past the feature being classified in the same way as the motion in Figure 7.2(a). The results show that with increasing confidence in the parameter model $p(\{M\} \mid \mathbf{x})$, the posterior entropy $(-\mathbf{I}(k))$ is decreased more rapidly. With model 1 (highest confidence), the initial observations contain a lot of information as shown by $\mathbf{i}(k)$, but as uncertainty decreases, the information content of the observations decreases to negligible quantities at about $k = 175$ as shown in (b). Using model 2, useful albeit low information content observations continue to be made up to $k = 300$. This is also reflected in the information gain plots in (e) where using model 2, information gain continues to rise.

Variation with displacement step ξ. In Figure 7.5, we vary the minimum displacement step ξ, for a sensor moving linearly past the feature being classified. We make use of three typical values of ξ. It can be seen in (a) that entropy is more rapidly reduced with the larger value of ξ even though this means that updates occur less frequently compared to using a smaller step value. This is because the classification as implemented in Chapter 6, is better posed with a larger separation between the two observation positions used in the classification algorithm as shown in Figure 6.15. The information in the observations reflects this as shown in (b), (c) and (d). The information gain plots in (e) show that for a step value of $0.03m$, the sensor has little additional information to gain by continuing observations of the particular feature beyond the $150th$ time-step. In Figure 7.4, we make use of the same step values as above but the vehicle motion is arbitrary. With arbitrary motion, the location information is even less accurate, because sudden changes in direction or motion are not

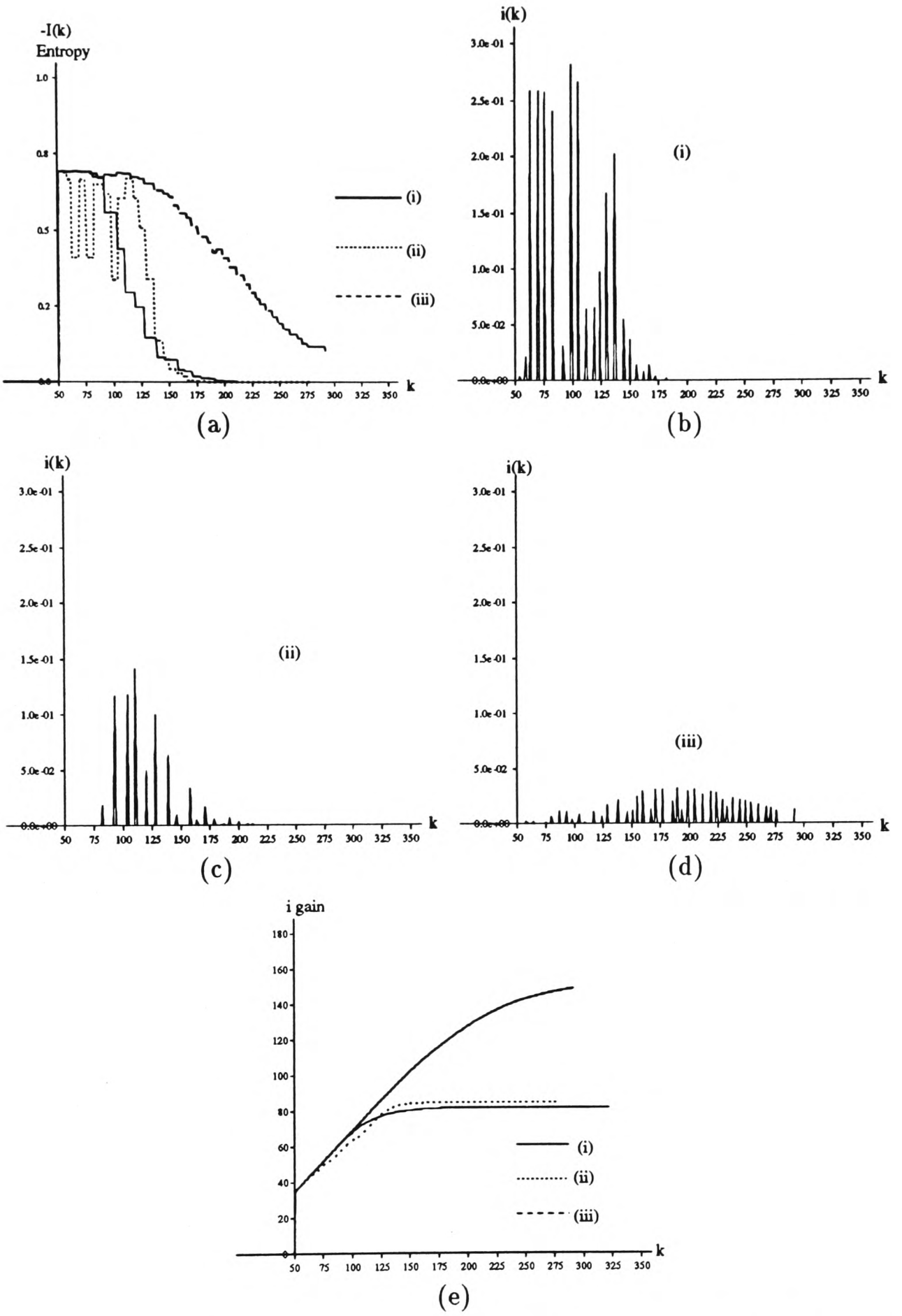


Figure 7.3: Variation of classification information with parameter model. Motion is linear past a tracked feature being identified. (a) Entropy (negative information), (b), (c) and (d) observation information. (e) observation information gain. (i) is model 1, (ii) is model 3, and (iii) is model 2.

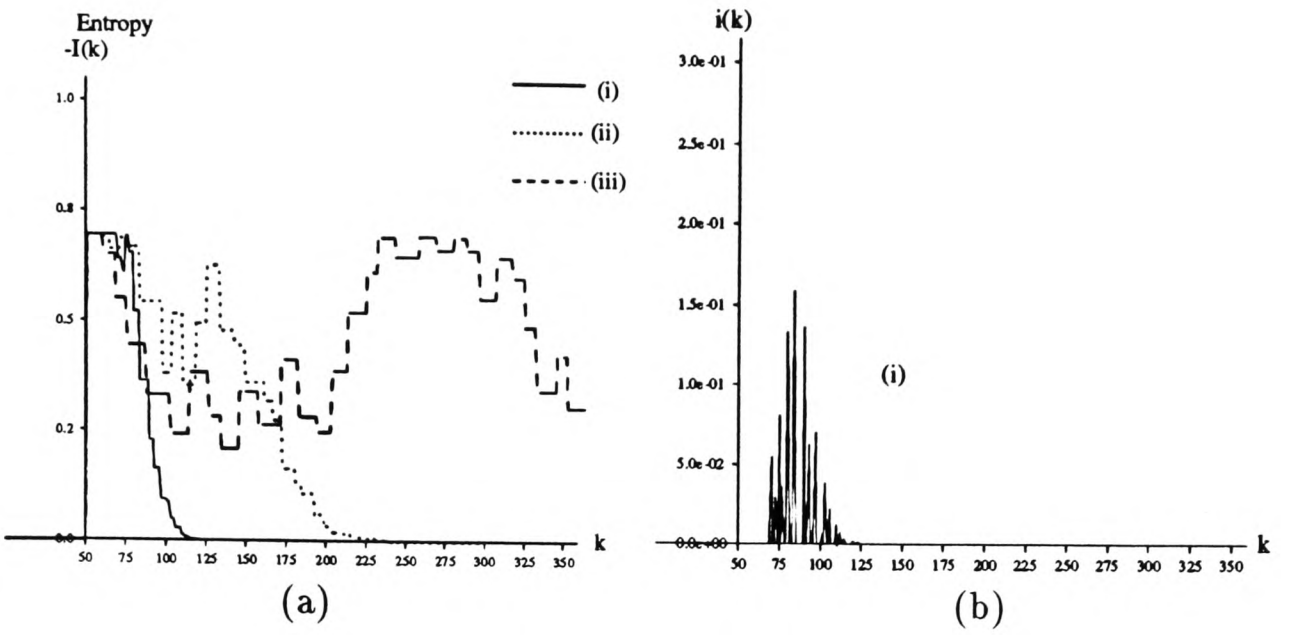


Figure 7.4: Variation of Classification information with accuracy of location information (that is variation in displacement step ξ). Motion is arbitrary in front of tracked feature. (a) Entropy (negative information) (b) (c) and (d) Observation information. (e) Observation information gain. For (i) step = $0.03m$ (ii) step = $0.02m$ and (iii) step = $0.01m$.

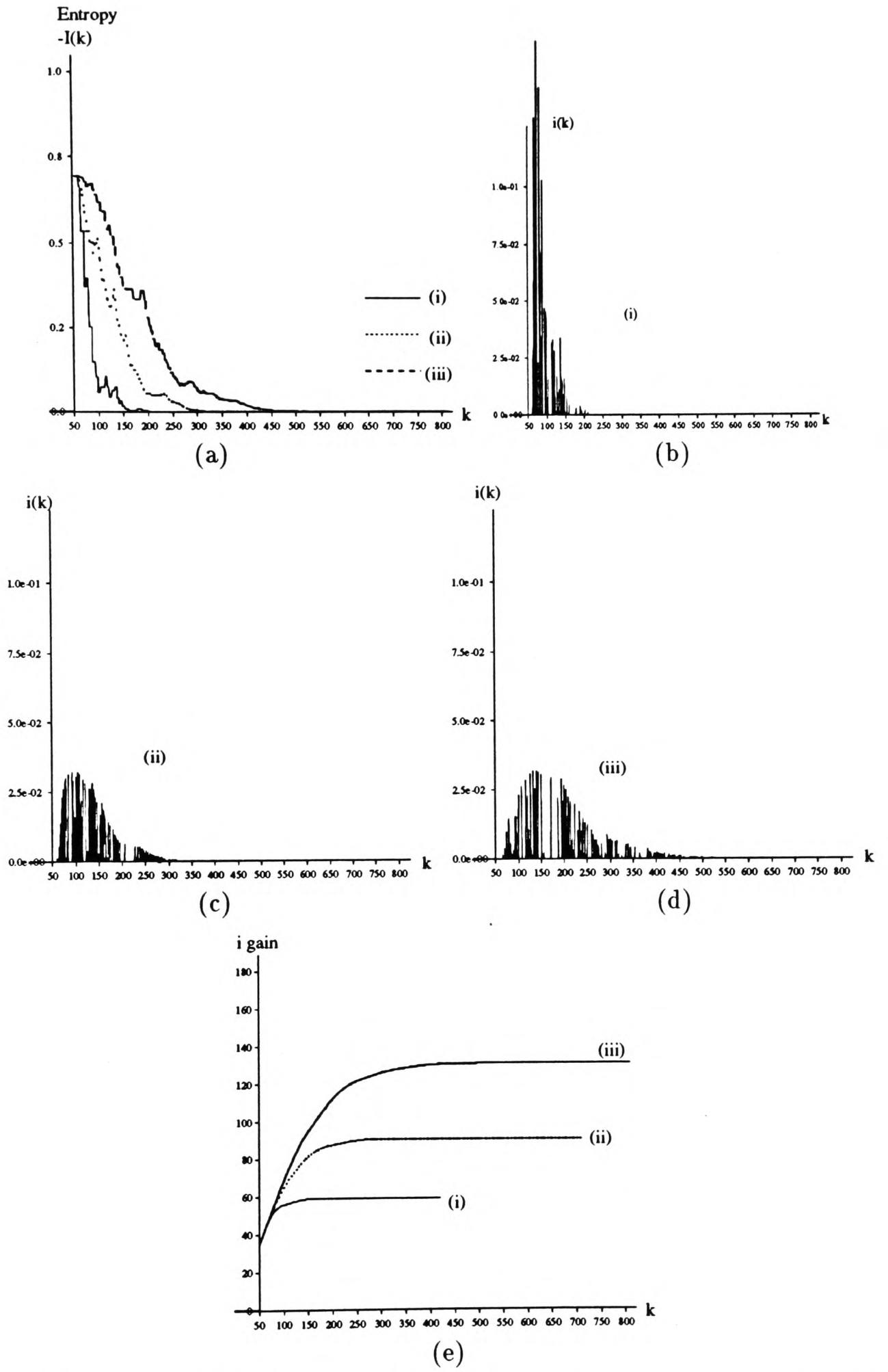


Figure 7.5: Variation of Classification information with accuracy of location information (that is variation in displacement step ξ). Motion is linear past a tracked feature being identified. (a) Entropy (negative information), (b), (c) and (d) Observation information. (e) Observation information gain. For (i) step = $0.03m$ (ii) step = $0.02m$ and (iii) step = $0.01m$.

always reflected immediately in the location estimates (due to the estimation lags which vary with $\mathbf{R}[k, \mathbf{z}(k)]$, see results in Chapter 6). With a step of $0.03m$ the rate of uncertainty reduction is not greatly affected. With a step of $0.01m$, entropy decreases and then increases due to ambiguities in the classification of the feature, this is also reflected in the observation information. An example of such ambiguities in classification was illustrated in the results shown in Figure 6.18(b).

7.2.3 Discussion

The results presented serve to illustrate the effect of various models and parameters on the information obtained and its effect in reducing uncertainty (entropy) and hence increasing information. This is important because of the subsequent use of such information quantities to manage sensor systems. The results also illustrate the effect of relative spatial locations on the information obtained. This justifies the fact that if several such sensors are used, as in the decentralized implementations of Chapter 6, some sensors will be better suited than others for certain sensing tasks by virtue of their spatial locations and their ability to make observations.

7.3 Feature-Sensor Assignments

In the algorithms developed in Chapter 4, decisions regarding sensor-feature assignments are made by choosing assignments which maximize the information obtained. Here we demonstrate how this is achieved in situations involving vehicle motion, the stationary case following naturally from these results.

7.3.1 Demonstration 1

Two sensors are in a position to observe two features for purposes of localization using the decentralized algorithm implemented in Chapter 6 as illustrated in Figure 7.6. The action set in this case is given by

$$\mathcal{A} = \left\{ \begin{array}{l} a_1 = (t_a \rightarrow \text{sensor } 2, t_b \rightarrow \text{sensor } 1), \\ a_2 = (t_a \rightarrow \text{sensor } 1, t_b \rightarrow \text{sensor } 2) \end{array} \right\}.$$

Figure 7.6 (a) shows assignment (i) corresponding to a_1 and (b) shows assignment (ii) corresponding to a_2. The vehicle is stationary up until time-step $k = 75$, after which it starts to rotate. Considering time prior to rotation, the following can be observed from (c) by considering partial information up to k;

$$(\tilde{\mathbf{I}}_1(k), a_2) > (\tilde{\mathbf{I}}_1(k), a_1) \quad \text{and} \quad (\tilde{\mathbf{I}}_2(k), a_2) > (\tilde{\mathbf{I}}_2(k), a_1).$$

By considering observation information at time k the following can be observed from Figure 7.6(d);

$$(\mathbf{i}_1(k), a_2) > (\mathbf{i}_1(k), a_1) \quad \text{and} \quad (\mathbf{i}_2(k), a_2) > (\mathbf{i}_2(k), a_1).$$

From Equation 4.60 the optimal action is clearly obtained as

$$\hat{a} = \arg\max_a \sum_j \tilde{\mathbf{I}}_j(k),$$

or by considering observation information

$$\hat{a} = \arg\max_a \sum_j \mathbf{i}_j(k).$$

In both cases $\hat{a} = a_2$. It can also be noticed that with both a_1 and a_2, sensor 1 consistently obtains more information than sensor 2 while the vehicle is stationary. At time step $k = 75$ the vehicle starts to rotate and with both assignments, the information starts to increase.

However, with assignment (i), sensor 1 soon loses its feature as it moves out of view and the tracking process is terminated. With assignment (ii) we can continue tracking. At time $k = 150$, the trend changes and sensor 1 experiences a drop in the information content of its observations as shown in (d). On the other hand sensor 2 continues to obtain increased information in its observations. At time $k = 250$, sensor 2 starts to obtain more information from its observations than sensor 1. This is reflected in the partial information content $\tilde{\mathbf{I}}_i(k)$ in (c) where that of sensor 1 falls below that of sensor 2. Because observation information is assimilated in the data fusion process, this results in *both* the partial information content metrics falling but with that of sensor 1 falling faster. It can also be seen that sensor 2 has the higher utility as compared to the time prior to $k = 225$ when sensor 1 had the higher utility. Therefore for this example the optimal sensor-feature assignment starts as and remains assignment (ii) (that is, action a_2).

7.3.2 Demonstration 2

In this demonstration, the vehicle moves in a straight line as illustrated in Figure 7.7 (a) while each of the sensors tracks one of two features. The action set considered in this case, is given by

$$\mathcal{A} = \left\{ \begin{array}{l} a_1 = (corner \rightarrow \text{sensor } 2, cylinder \rightarrow \text{sensor } 1) \\ a_2 = (corner \rightarrow \text{sensor } 1, cylinder \rightarrow \text{sensor } 2) \end{array} \right\},$$

where in Figure 7.7, a_1 and a_2 are assignments (i) and (ii) respectively. The two possible assignments give rise to the results in (b) and (c). It can be seen that assignment (ii) results in higher information content. Although in assignment (ii) the sensors collectively obtain more information in their observations, sensor 2 in assignment (i) obtains increasingly higher information from its observations $\mathbf{i}_2(k)$ which start to raise the information content $\tilde{\mathbf{I}}_1(k)$ and $\tilde{\mathbf{I}}_2(k)$ of assignment (i).

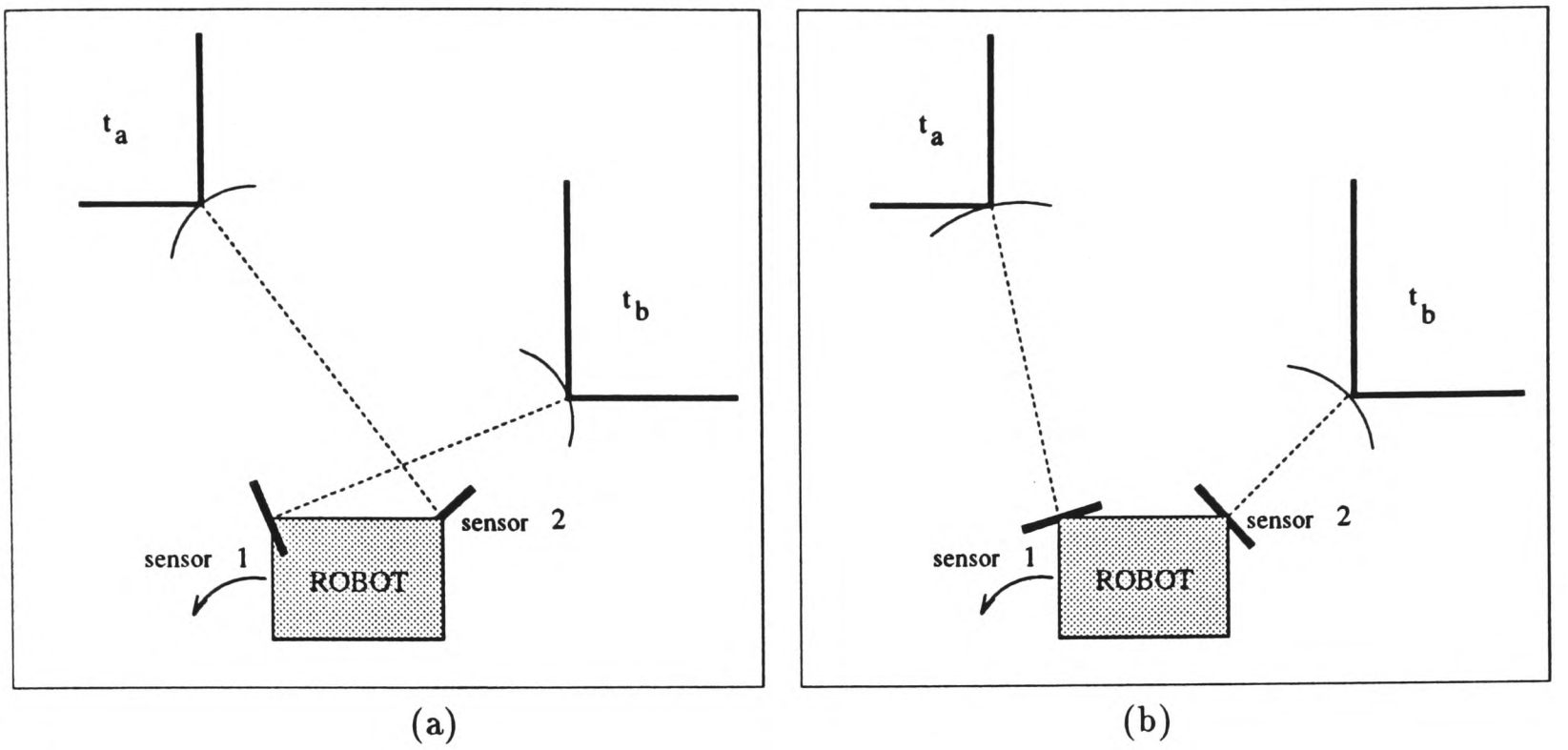
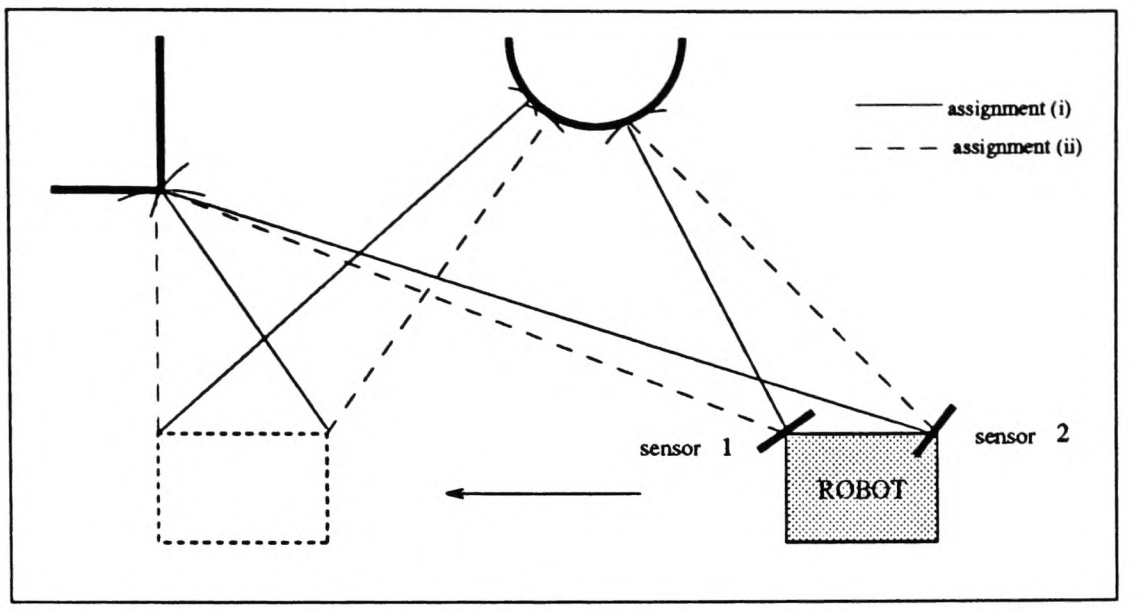
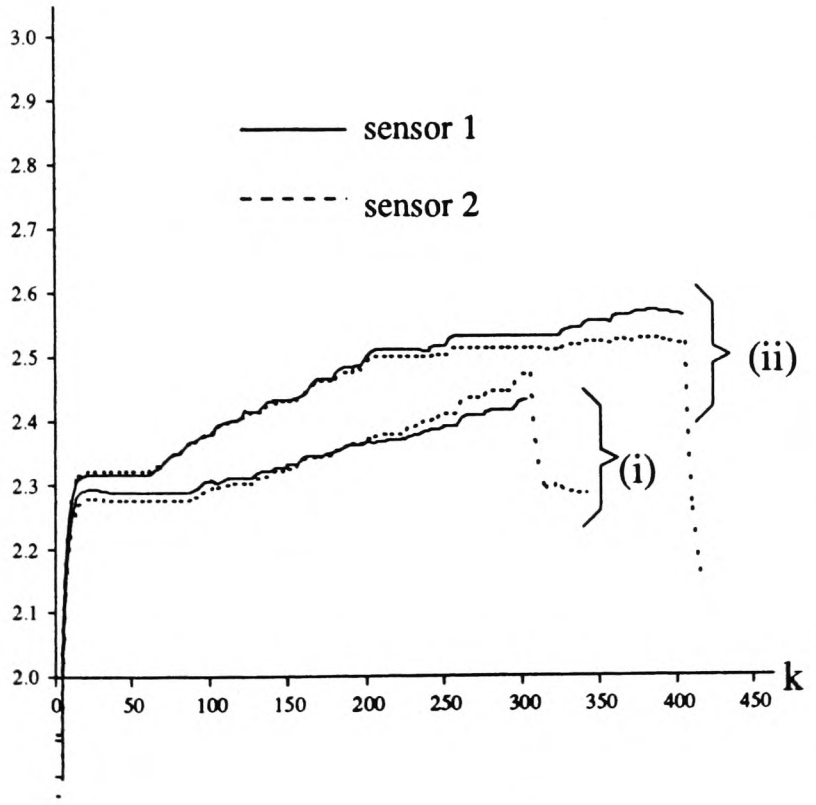


Figure 7.6: Variation of information quantities with two feature-target assignments while the vehicle is rotating counter-clockwise. (a) illustrates assignment (i) and (b) assignment (ii). Shown in (c) is the partial posterior information $\tilde{\mathbf{I}}_i(k)$ for each of the sensors corresponding to each assignment. Shown in (d) is the observation information $\mathbf{i}_i(k)$ for each sensor corresponding to each assignment. The run in assignment (ii) is shorter because at approximately $k = 150$ sensor 1 loses its feature as it moves out of view.

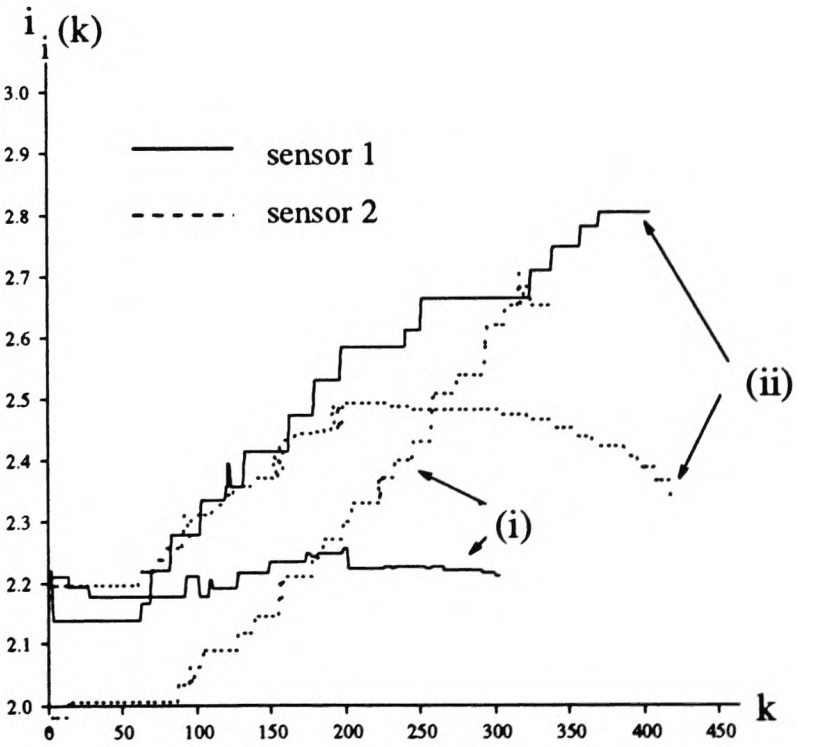


(a)

$\tilde{I}_i(k)$



(b)



(c)

Figure 7.7: Variation of information quantities with two feature-target assignments while the vehicle moves as indicated in (a). Illustrated in (a) are the assignments (i) and (ii). Shown in (b) is the partial posterior information $\tilde{\mathbf{I}}_i(k)$ for each of the sensors corresponding to each assignment. Shown in (c) is the observation information $\mathbf{i}_i(k)$ for each sensor corresponding to each assignment.

7.3.3 Demonstration 3

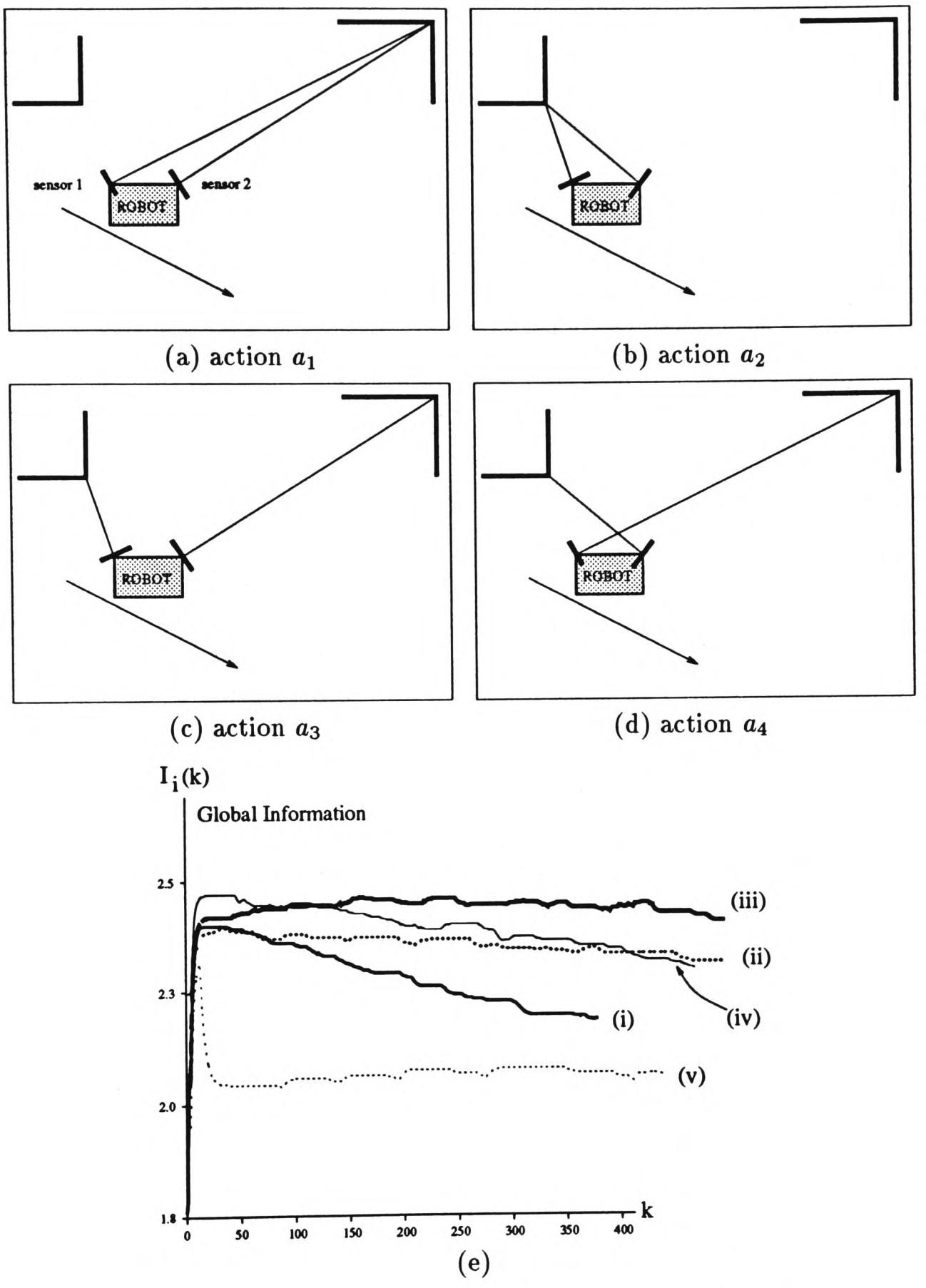
An optimal sensor feature assignment in localization should allow the sensor system to obtain the most information possible about the location of the vehicle. To demonstrate this, we illustrate how the global posterior information $\mathbf{I}_i(k)$ is maximized through management decisions. In this demonstration, there are two features which can be used for localization by a vehicle with two Tracking Sonars. The action set under consideration is given by

$$\mathcal{A} = \left\{ \begin{array}{llll} a_1 = (\text{corner} & \rightarrow \text{sensor 2}, & \text{corner} & \rightarrow \text{sensor 1}), \\ a_2 = (\text{edge} & \rightarrow \text{sensor 2}, & \text{edge} & \rightarrow \text{sensor 1}), \\ a_3 = (\text{corner} & \rightarrow \text{sensor 2}, & \text{edge} & \rightarrow \text{sensor 1}), \\ a_4 = (\text{edge} & \rightarrow \text{sensor 2}, & \text{corner} & \rightarrow \text{sensor 1}), \\ a_5 = (\text{edge} & \rightarrow \text{sensor 1}) & & \end{array} \right\}.$$

The environment is illustrated in Figure 7.8 and shows the different sensing strategies given in the action set. In Figure 7.8(e) we plot the global information from the information filter for each of the different strategies. It can be seen that the strategy corresponding to action (iii) results in the most information being obtained about vehicle location. Therefore this action is chosen as the most optimal one. The maximization to obtain this optimal action is based on the algorithm communicating expected utilities in the form of partial information $\tilde{\mathbf{I}}_i(k)$ as discussed in Chapter 4.

7.4 Sensor Hand-off and Cueing

In the context of vehicle navigation using Tracking Sonars, hand-off and cueing refers to the notion of a sensor currently observing a particular feature handing over observation of it to another sensor that has a higher utility for tracking this feature. This amounts to making decisions about which sensor is best suited to a given tracking task at any given point in time and consequently allowing that sensor to perform the tracking task. We demonstrate this for both the localization and the feature classification algorithms.



(a) action a_1 (b) action a_2

(c) action a_3 (d) action a_4

(e)

Figure 7.8: Various sensing strategies for a vehicle with two sensors and able to track two features. (a) assignment (i) (b) assignment (ii) (c) assignment (iii) and (d) assignment (iv). (e) shows the global posterior information after assimilation for each of the sensing actions. Plot (v) corresponds to only using sensor 1 to track the edge.

7.4.1 Demonstration 4

Consider two sensors on a vehicle, tracking the same feature for purposes of localization. In the first instance, the vehicle moves in straight line past the feature. We now demonstrate the sensor hand-off. In the results of Figure 7.9, we show the information quantities using two location covariance models; model 1: $\sigma_\theta^2 = \sigma_r^2 = 0.0001$ and model 2: $\sigma_\theta^2 = 0.001$, $\sigma_r^2 = 0.005$. From the results in (a), it can be seen that with model 2, the hand-off point is difficult to distinguish if only partial information $\tilde{\mathbf{I}}_i(k)$ is considered. However, if observation information $\mathbf{i}_i(k)$ is considered as well, the hand-off point is easily distinguishable with both models. The significance of this is that, if metrics such described in Chapter 4 Section 4.7.1 are being used, based on a fixed threshold level, the information to be gained from management may fall below the threshold level and hence is not implemented. (See Equation 4.77.). For this reason, *both* observation information $\mathbf{i}_i(k)$ and partial information $\tilde{\mathbf{I}}_i(k)$ must be considered for hand-off and cueing.

Figure 7.10 shows hand-off for a vehicle moving back and forth while two sensors track the same feature. Again the observation information $\mathbf{i}_i(k)$ in (b) shows the hand-off points quite distinctly, whereas the partial information $\tilde{\mathbf{I}}_i(k)$ in (a) does not. In (b) it can be seen that from the start to $k = 200$, sensor 1 has the higher utility and between $k = 200$ and $k = 400$ sensor 2 has the higher utility. After $k = 400$ sensor 1 again has the higher utility.

7.4.2 Demonstration 5

Similar hand-off results can are obtained when two sensors are classifying the same feature as the results in Figures 7.11 and 7.12 show. The difference between the localization and classification results is that for the classification, the partial entropy ($-\tilde{\mathbf{I}}_i(k)$) is more useful than the observation information $\mathbf{i}_i(k)$. This is because the observation information is

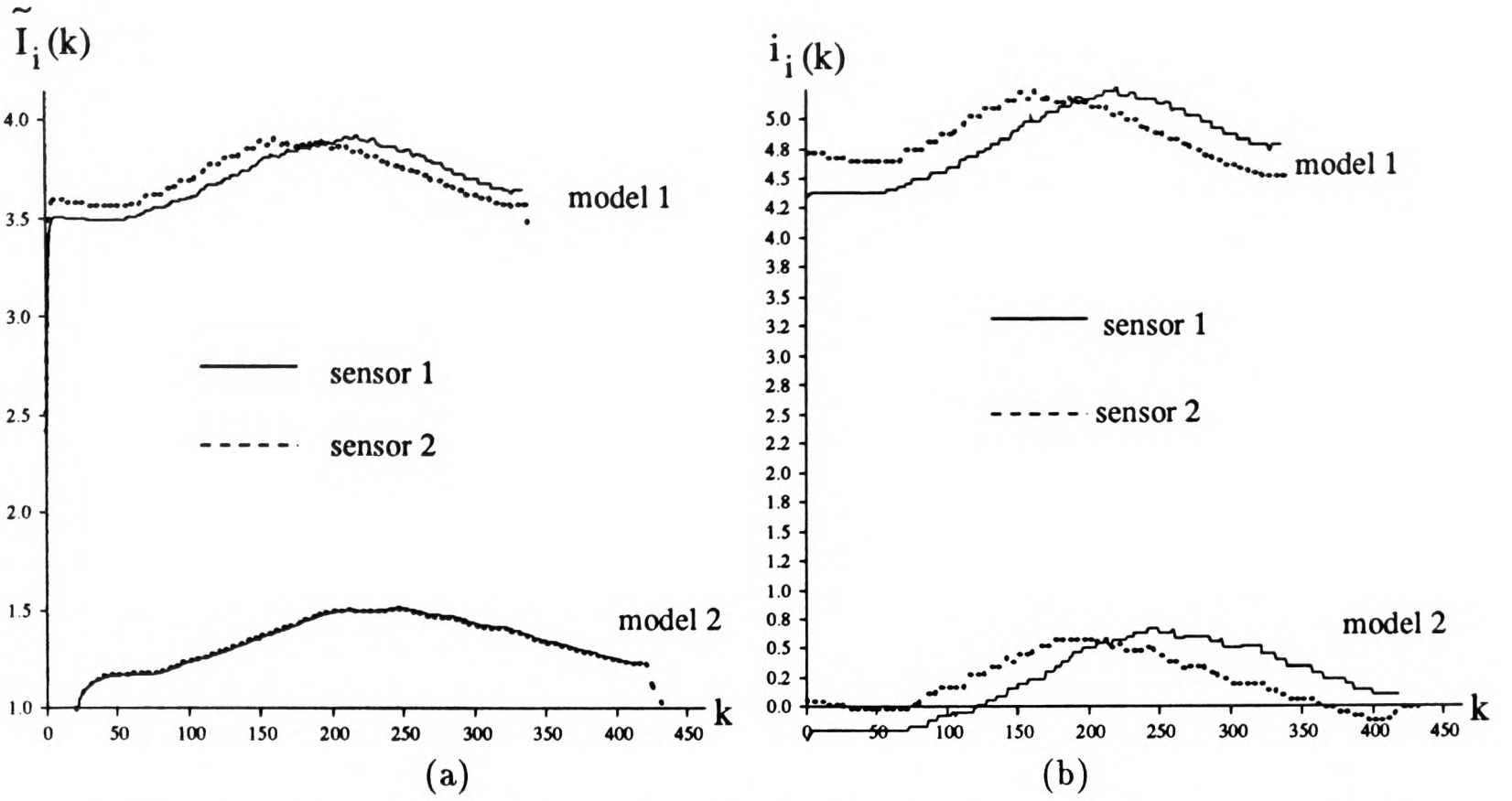


Figure 7.9: Illustrating partial and observation information used for hand-off while vehicle motion is in a straight line. Different parameters resulting in different location covariance models are used, that is, model-1 is $\sigma_\theta^2 = \sigma_r^2 = 0.0001$ and model-2 is $\sigma_\theta^2 = 0.01, \sigma_r^2 = 0.005$. Hand-off is at $k = 210$.

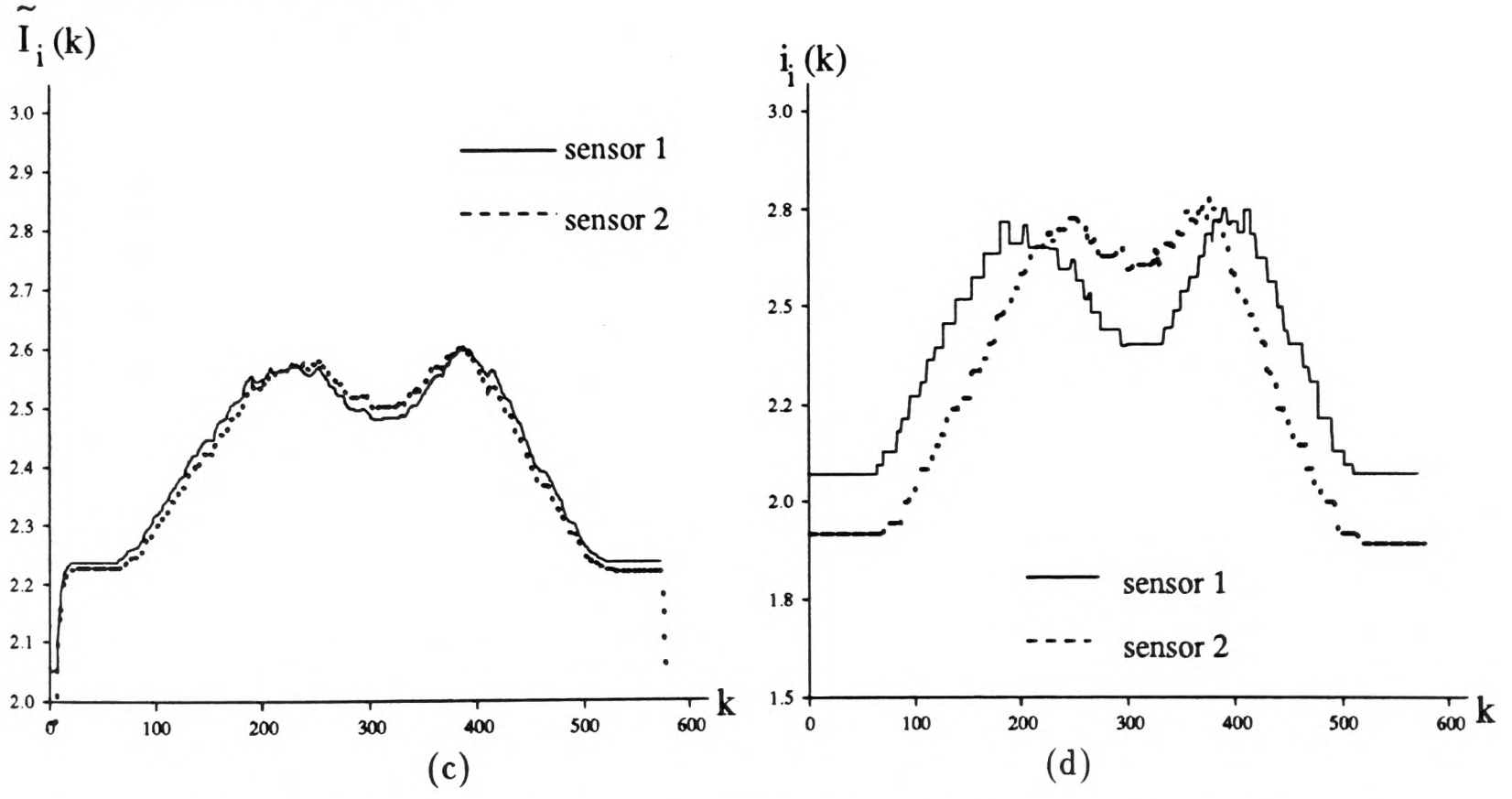
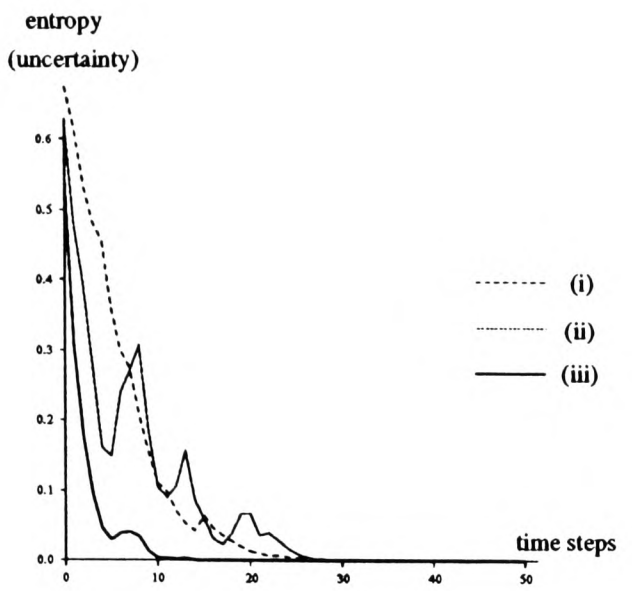
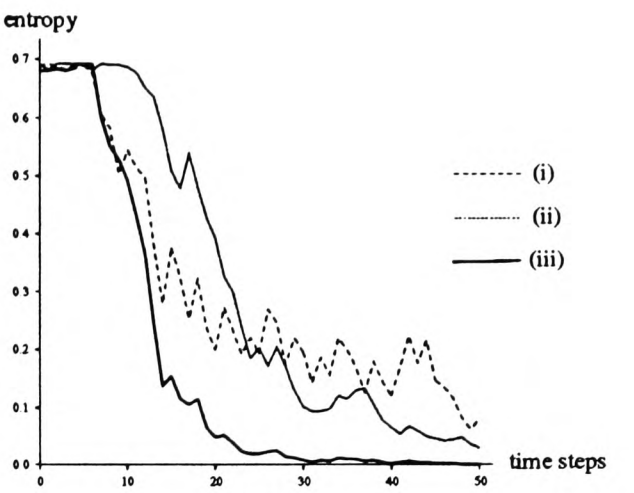


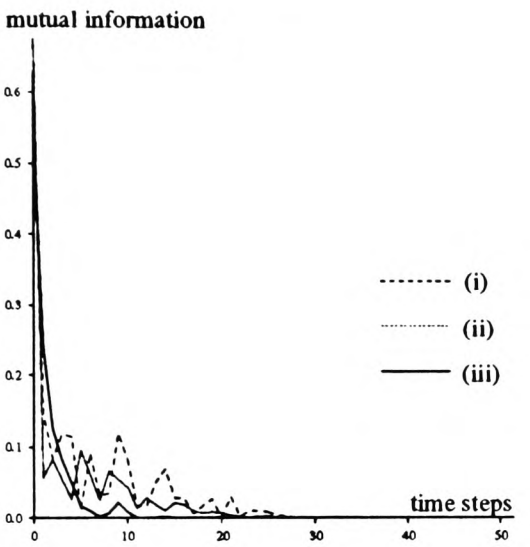
Figure 7.10: Illustrating partial and observation information used for hand-off as vehicle moves back and forth. Hand-off times are $k = 200$ and $k = 400$.



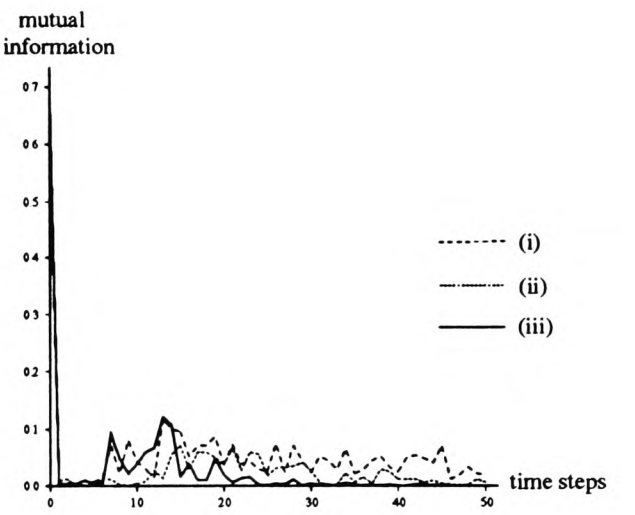
(a)



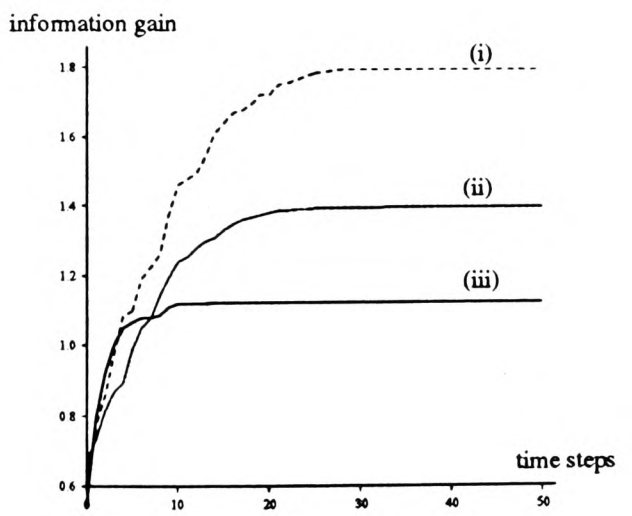
(b)



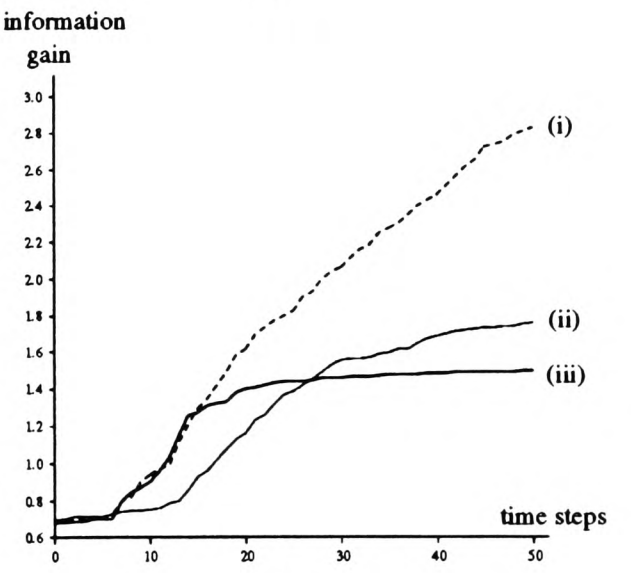
(c)



(d)



(e)



(f)

Figure 7.11: The information values for two sensors running the Decentralized Classification algorithm while tracking the same feature . (a) and (b) show the entropy ($-\tilde{\mathbf{I}}_i(k)$) (c) and (d) the mutual (observation) information $\mathbf{i}_i(k)$ and (e) and (f) the observation information gain. In these (i) is sensor 1 information (ii) is sensor 2 and (iii) is the assimilated (fused) information. Two classification observation models are used, model-1 in (a)(c)(e) and model-2 in (b)(d)(f).

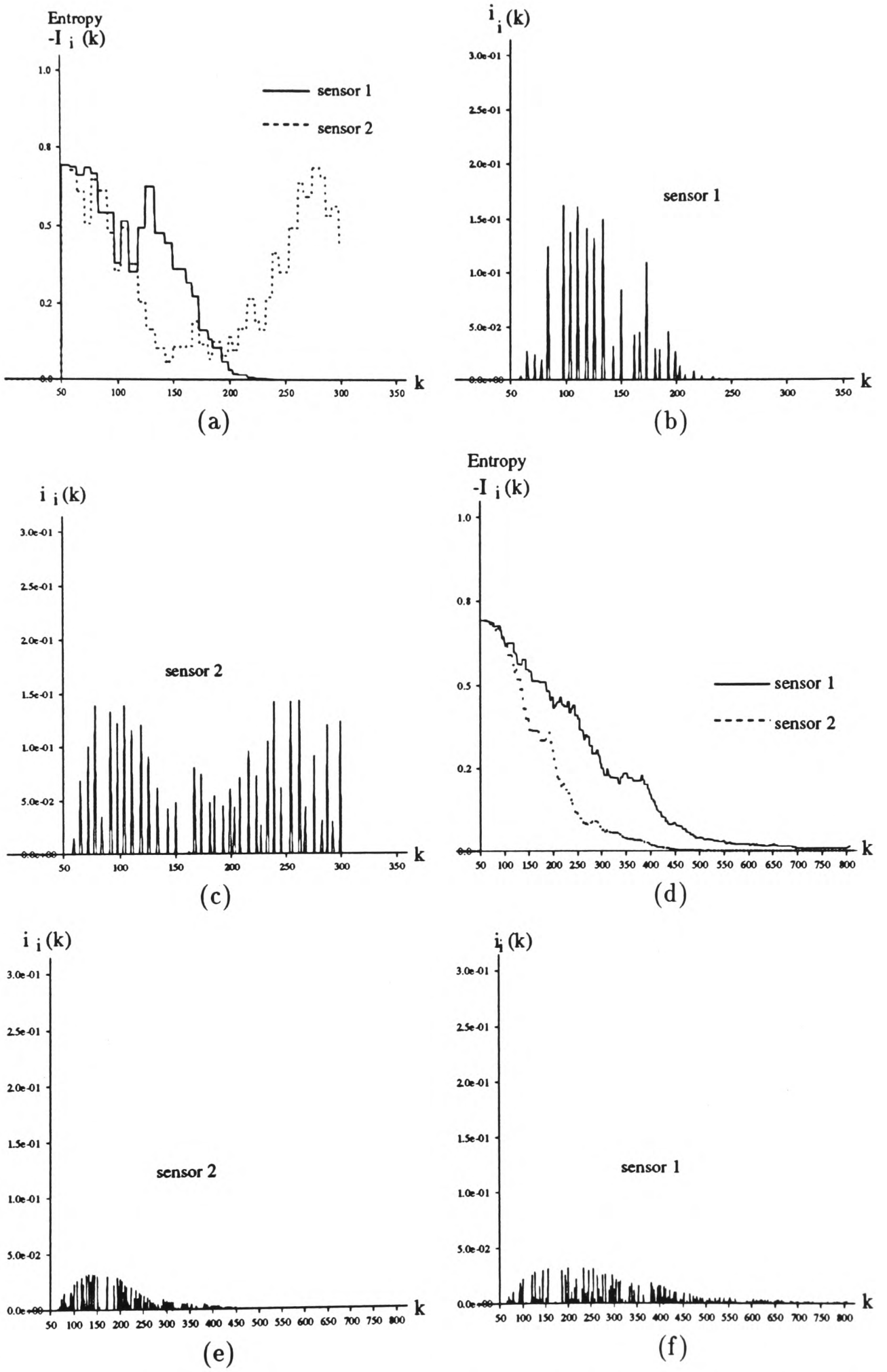


Figure 7.12: Partial information at each sensor for two sensors classifying the same feature. Classifying a corner, (a) the partial entropy at each sensor (b) observation (mutual) information at sensor 1 (c) observation information at sensor 2. Classifying a plane (d) partial entropy at each sensor (e) and (f) observation information at each sensor.

somewhat unstable and varies so rapidly that it would result in unwarranted and frequent hand-off and cueing. This is evident in the results of Figure 7.11(c) and (d), and Figure 7.12 (b), (c), (e) and (f). Cost metrics such as those based on the value of information and a threshold (see Chapter 4, Section 4.7.1), can be used to damp out the effect on decisions due to the rapid fluctuations in the observations by introducing a dead-band. On the other hand the entropy is stable and distinct enough to be used for hand-off as shown in Figure 7.11 (a) and (b) and Figure 7.12 (a) and (d).

7.5 Discussion

Due to the small number of sensors involved in our implementation, the time taken to obtain the rational action is negligible. Therefore the bargaining algorithm of Chapter 4, can be run to completion at no significant computational cost. However, there are several practical problems associated with effecting management decisions such as the occasional failure to validate a newly initiated target following a hand-off decision. However, these are largely a result of the hardware limitations of the Tracking Sonars and are eliminated or at least significantly reduced with improved hardware (See Appendix C) and track validation.

While the results presented have been for a "stationary-feature-moving-vehicle" arrangement, similar results are obtained when the Tracking Sonars are on a stationary platform and the features themselves are moving. In such a case, if the Tracking Sonars are replaced by CCD cameras this becomes a surveillance system where management decisions can be used to make target assignments in a similar way. Such management implementations can be extended to systems with a large number of sensors and action sets. And since performing the maximizations required can be computationally significant, the bargaining algorithm of Figure 4.8 can be employed.

Chapter 8

Conclusions

This thesis has addressed the development of a single consistent framework for data fusion and sensor management in multi-sensor systems in general and decentralized systems in particular. We have also demonstrated the application of such a frame-work and the methods derived from it, to mobile robotics. In conclusion, we discuss the contributions made by the work presented in this thesis as well as its limitations and suggest directions for future research.

8.1 Contributions

A Unified and Consistent Approach to Data Fusion.

The development of a unified approach to data fusion and sensor management has in the past not been crucial to the development and implementation of multi-sensor systems. This is because in centralized and hierarchical systems, *ad hoc* methods can be made to perform well by taking advantage of the fact that decisions are referred to the central processor thus guaranteeing consistency. With the advent of distributed and decentralized systems, the consistency and coherence of data fusion methods have become necessary due to the presence of several decision-makers. In this respect, based on justifiable axiomatic considerations we have presented a single comprehensive model in which consistency and coherence are implicit. Using this model we have shown the following:

- The model provides a homogeneous representation of sensor **Information** and **Uncertainty** and allows the probabilistic information from sensor measurements to be fused in a consistent manner.

- **Architectures** and continuous and discrete **Algorithms** for data fusion have been derived in a mutually consistent manner starting from the information update.

- The model provides facilities for **Decision-theoretic management** based on utility theory; from this we have developed methods for decentralized sensor management.

However, the effectiveness of this model is dependent on an ability to model the information from sensors and the outcomes of decisions probabilistically.

Methods for Decentralized Sensor Management.

Sensor management in decentralized systems reduces ultimately to a problem of decentralized decision-theory for which the existence and nature of the solution depends on the constraints placed on the decision-makers and the decision process itself. In this respect, the work on sensor management that has been presented in this thesis is an important contribution in that it presents, for the first time, a normative method for sensor management in decentralized systems. The management methods are based on the assumption that the gain of information for the tasks of inference and perception forms the management imperative. To our knowledge, normative methods for coordinating and managing sensor resources have only been described for centralized and hierarchical systems. The methods that we have presented are applicable to a variety of situations where autonomous sensors in a decentralized system are required to make non-conflicting choices concerning sensing actions in a manner that is optimal and rational. However, the formulation as we have presented, it is dependent on an innate ability to compare and evaluate the utilities of all

the sensors in the system. We justified such comparisons by basing the utility structure on information in an entropy or Fisher sense. This amounts to evaluating alternate sensing actions for their information value with respect to the state being inferred. While this is "good and proper" for a system whose primary goal is perception, it places limitations on the evaluation of actions not directly concerned with the perception task at hand. It could however, be argued that any action by a sensor can always be abstracted to its informational value. What becomes a challenge, is assigning value in the form of expected utilities to such actions so as to facilitate decision making. In addition, assigning such value requires a probabilistic model of the outcomes of such actions. This places limitations on the extensibility and maintainability of such a system because any additions or changes in the functionality of the system require a "proper" probabilistic interpretation. Notwithstanding such limitations, our approach to sensor management has been validated by the results and demonstrations that we have presented.

A Novel Sonar Model.

In the first instance, the sonar model that has been presented highlights the importance of understanding and correctly modelling information from a sensor, these being both a requirement of normative methods and also necessary for the uncertainty models used in the data fusion algorithms. More importantly however; the Tracking Sonar is, in its novelty, a contribution of considerable significance to sensing for mobile robotics and represents an important new development in sonar data processing. This sensor model extends the concept of "focus of attention" largely exploited in vision systems, to standard sonar at a cost several orders of magnitude less than equivalent vision systems. However, the practical implementation of the Tracking Sonar is very much at the prototype stage leaving room for

further improvements and applications.

A New Approach to Navigation.

Making use of the Tracking Sonar and the algorithms for data fusion and sensor management, we have presented new concepts and techniques for realizing autonomous sonar navigation systems. The speed of localization is several orders of magnitude faster than previously published work using standard Polaroid sonar[90]. Each Tracking Sonar operating in a decentralized system is able to generate location estimates at rates of the order of $25Hz$. The amount of data processing is also vastly reduced by removing from the localization cycle the data intensive and computationally significant process of "scanning and matching" every processing cycle[90]. And so through management, this thesis demonstrates "true" directed sonar sensing by directing the Tracking Sonars to sense only that which is necessary for the task at hand. This ability to manage sensors presents exciting new possibilities for navigation.

However, the implementations that we have presented are very much at the prototype stage and more work is required to produce robust and fully functional navigation systems. Our aim has been to introduce new approaches and to demonstrate what can be achieved.

8.2 Directions for Future Research

The contributions outlined above provide the basis and methods for further research in several areas. Because of the relatively short history of research in decentralized sensor management many of the concepts we have presented are new, hence throughout this thesis we have endeavoured to highlight areas that can be improved and explored further. However, here we summarize the main issues for further research:

Towards a General Normative Management Strategy

As already discussed, the actions which can be evaluated using our approach are those directly related to sensing alternatives whose value with respect to the inference process can be quantified consistently as an expected utility. The ability to express value formally in terms of expected utility requires a probabilistic modelling of the outcome of a sensing operation. When actions and their outcomes cannot be subjected to such rigorous modelling our approach, as it stands, becomes ineffectual. In this respect, more work needs to be done in order to develop methods for modelling outcomes for a generalized class of actions.

In the algorithms of Chapter 4, there is a need to communicate observations in order for each sensor to evaluate its own utility for tasks currently being carried out by another sensor. This can be avoided through a less rigorous method in which utilities are based on segmenting the environment into grids in such a way that each grid has an associated utility[1]. While the discussion in Chapter 4 considered the computational and temporal costs of blindly pursuing a purely decision-theoretic solution, the cost of implementing actions was not considered. Clearly such costs in a real system, ought to have bearing on the choice of optimal action. A starting point in this direction could be the use of *cost functions* such those presented by Pearl [113]. Cost functions can also be used to model outcomes for a general class of actions. While the use of such techniques may be disconcerting to Bayesian purists who might call to question the normative status of the solution obtained, it can be argued that such cost functions can be developed axiomatically as has been done with utility functions. This is indeed the approach taken in the very recent work by Blackman for the management of agile radar [20]. Such cost functions provide a formal method for dealing with what may otherwise be an intractable problem in a strictly normative sense.

[1]This could be similar to the occupancy grids of Elfes [51] used in a different context.

Our entire presentation of decentralized sensor management has been for fully connected systems. However, as described in the work of Grime [62], for large scale systems, full interconnection becomes impractical and hence the development of non-fully connected decentralized systems aimed at reducing the connectivity of the system while yielding results which approach those of fully connected systems. The methods that we have presented need to be extended to such systems. This could be done either by assuming the communication solutions presented by Grime [62] or by adopting an alternate approach based on results from Team Decision theory [73]. Ho [72] gives a general discussion on the organisation of multi-sensor systems based on Team theory. However, an approach which is closer to the work presented in this thesis, is the use of *sub-sampling* techniques amongst large groups of Bayesian decision makers as discussed in the theoretical work by Weerahandi and Zidek [134] and Zidek [138] and taking advantage of the resulting communication simplifications. In essence, sub-sampling means that from a large group of decision-makers, a smaller group with the most to gain takes charge of the decision process, and so the decision problem amongst this smaller group becomes similar to what we have presented.

The methods that we have presented also need to be applied to systems with large action sets and a large number of sensors and the performance evaluated. Our application has been such that the number of sensor involved is relatively small and hence management performance in this respect cannot be fully evaluated.

Fully Functional Robust Decentralized Navigation

Application of the methods and tools presented in this thesis to navigation needs to be explored further and implemented as a fully functional system. The first step is to improve the prototype design of the Tracking Sonar in order to make it more robust; firstly by

improving the hardware and secondly by developing a more stringent validation scheme for maintaining focus of attention. A more detailed quantitative analysis of performance is also required.

The decentralized localization scheme needs to be more rigorously evaluated and tested for robustness and reliability. This would entail considering various criteria such as; performance over extended periods, absolute accuracy of location information, and quantitative comparison with other methods. Ultimately, we envisage a fully functional vehicle making use of managed Tracking Sonars in a decentralized configuration, based on the one we have demonstrated, navigating and building maps in a robust manner. Such work has already started on the OxNav vehicle where the Tracking Sonars are being integrated into a sensor suite incorporating; inertial navigation sensors, a gyroscope, infra-red and encoders. This provides an ideal platform for the application of the work in this thesis to a wider range of sensing technologies in what is intended to be a fully functional system.

8.3 Summary

What has been presented in this thesis has been a consistent methodology for addressing data fusion and sensor management decentrally. Much more needs to be done to generalize these methods and make systems using them fully functional and robust. The implementation and results obtained have served to demonstrate practically that these methods can indeed be applied to current research problems in robotics and implemented in actual systems yielding exciting new results. Along the way, we have also developed a new sonar sensor, and improved the speed of sonar based localization by several orders of magnitude.

Appendix A

Entropy and Information

A.1 Entropy of a vector distribution given covariance

For an n-dimensional vector $\mathbf{x}$ with a normal distribution, the PDF is given by

$$p(\mathbf{x}) \;=\; \mathbf{N}(\bar{\mathbf{x}},\mathbf{P}) = |2\pi\mathbf{P}|^{-1/2}\, exp\left\{\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right\}, \tag{A.1}$$

where $\bar{\mathbf{x}}$ is the mean of the distribution and $\mathbf{P}$ the covariance matrix. The entropy for this normal distribution is obtained as follows

$$\begin{aligned}
E\{ln\, p(\mathbf{x})\} &= -\frac{1}{2}E\left\{(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) + ln[(2\pi)^n|\mathbf{P}|]\right\} \\
&= -\frac{1}{2}E\left\{\sum_{ij}(\mathbf{x}_i-\bar{\mathbf{x}}_i)(\mathbf{P}^{-1})_{ij}(\mathbf{x}_j-\bar{\mathbf{x}}_j)\right\} - \frac{1}{2}ln[(2\pi)^n|\mathbf{P}|] \\
&= -\frac{1}{2}\sum_{ij}E\{(\mathbf{x}_j-\bar{\mathbf{x}}_j)(\mathbf{x}_i-\bar{\mathbf{x}}_i)\}(\mathbf{P})_{ij} - \frac{1}{2}ln[(2\pi)^n|\mathbf{P}|] \\
&= -\frac{1}{2}\sum_{j}\sum_{i}\mathbf{P}_{ji}(\mathbf{P}^{-1})_{ij} - \frac{1}{2}ln[(2\pi)^n|\mathbf{P}|] \\
&= -\frac{1}{2}\sum_{j}(\mathbf{P}\mathbf{P}^{-1})_{jj} - \frac{1}{2}ln[(2\pi)^n|\mathbf{P}|] \\
&= \sum_{j}\mathbf{I}_{jj} - \frac{1}{2}ln[(2\pi)^n|\mathbf{P}|] \\
&= -\frac{n}{2} - \frac{1}{2}ln[(2\pi)^n|\mathbf{P}|] \\
&= -\frac{1}{2}ln[(2\pi e)^n|\mathbf{P}|].
\end{aligned} \tag{A.2}$$

This results shows that all that is required to give the entropy of a normal vector distribution is its length n and its covariance $\mathbf{P}$.

A.2 Non-informative priors and the Maximum Entropy Principle

Non-informative priors

A non-informative prior is a PDF that contains no information about the state $\mathbf{x}$. Finding such a distribution of the state $\mathbf{x}$ satisfying this condition is difficult when no information concerning the space $\mathcal{X}$ is known. A much used heuristic method is that due to Jeffreys[78] based on Fisher information whereupon the prior is set to

$$p(\mathbf{x}) = \mid \mathbf{J} \mid^{1/2}, \tag{A.3}$$

where $\mathbf{J}$ is the expected Fisher information given by the definition in Equation 2.48 or Equation 2.49. The main attraction of this approach is that it is not affected in any way by restrictions on the state space $\mathcal{X}$ [18].

An approach commonly used when some information is available concerning the state,is, ignoring philosophical inconsistencies, to assume an equivalence between "non-informativeness" and "least informativeness", so that the least informative distribution for $\mathbf{x}$ becomes the non-informative prior. The *Maximum entropy principle* states that the distribution, which for a given state gives the maximum entropy, can be used as the non-informative prior, so equating imprecision with ignorance. For a discrete distribution $\mathbf{x} = [X_1, X_2, \ldots, X_n]$, the maximum entropy distribution is obtained by assigning $p(X_i) = \frac{1}{n}, \quad \forall i$. The entropy for such a distribution is given by

$$h(p(\mathbf{x})) = -\sum \frac{1}{n} ln\left(\frac{1}{n}\right) = ln(n).$$

Since it can be shown that $h(p(\mathbf{x})) \leq ln(n)$ for all proper distributions $p(\mathbf{x})$ then indeed assigning equi-probabilities gives the maximum entropy distribution.

For continuous states, entropy is not bounded in this way and hence in general has no maximum. However, if information is known about the space $\mathcal{X}$ in the form of moments, these can be used as constraints based on which a distribution maximizing entropy can be derived as we now outline:

Maximum entropy priors for continuous distributions. Moment constraints about the distribution $p(\mathbf{x})$ are usually given as follows

$$\int_{\mathcal{X}} f_i(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \alpha_i. \tag{A.4}$$

What is then required is to find a maximum entropy distribution satisfying these moments constraints. The approach is as follows: For most continuous distribution only the first two moments are usually known, that is

$$\int \mathbf{x}p(\mathbf{x})\, d\mathbf{x} = \bar{\mathbf{x}} \quad \text{and} \quad \int \mathbf{x}^2 p(\mathbf{x})\, d\mathbf{x} = \mathbf{P} \tag{A.5}$$

The following are two approaches for obtaining the distribution maximizing entropy given these moment constraints:

- **Calculus Approach.** The Lagrangian [122] for these constraints can be written

$$L \triangleq -\int p(\mathbf{x})\, ln[p(\mathbf{x})]d\mathbf{x} + \lambda_0 \int p(\mathbf{x})d\mathbf{x} + \sum \lambda_i \int f_i(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \tag{A.6}$$

  where the λ_is are Lagrangian multipliers. This can be maximized by differentiating and equating to zero

$$\partial L/\partial \mathbf{x} = -ln[p(\mathbf{x})] - 1 + \lambda_0 + \sum \lambda_i f_i(\mathbf{x}) = 0. \tag{A.7}$$

  From this the form of the distribution which maximizes the entropy is obtained as

$$p(\mathbf{x}) = e^{\lambda_0 - 1 + \sum \lambda_i f_i(\mathbf{x})}, \tag{A.8}$$

where λ_i is chosen to maximize the constraints. Considering only the mean and covariance, Equation A.8 is written

$$p(\mathbf{x}) = e^{\lambda_0 + \lambda_1 \mathbf{x} + \lambda_2 \mathbf{x}^2}. \tag{A.9}$$

The result of this [29] shows that given the constraints, the normal distribution with the same first and second moments as given in the constraints, maximizes entropy.

- **Using information inequalities.** We assume that $p(\mathbf{x})$ is the distribution for which we require a maximum entropy prior which we denote $f(\mathbf{x})$. Given that the mean of $p(\mathbf{x})$ is $\bar{\mathbf{x}}$ and the covariance is $\mathbf{P}$, by the constraints (Equation A.5), the mean and covariance of $f(\mathbf{x})$ are the same. The following derivation is based on that presented by Cover [39]. From the definition of relative entropy [39] between two distribution, we can write the following information inequality

$$\int p(\mathbf{x}) ln \left[\frac{p(\mathbf{x})}{f(\mathbf{x})}\right] d\mathbf{x} \geq 0. \tag{A.10}$$

Simplifying and rearranging gives

$$0 \leq \int (p(\mathbf{x})\, lnp(\mathbf{x}) - p(\mathbf{x})\, lnf(\mathbf{x}))\, d\mathbf{x}, \tag{A.11}$$

and using the definition of entropy we obtain

$$h(p(\mathbf{x})) \leq -\int p(\mathbf{x})\, lnf(\mathbf{x}) d\mathbf{x}, \tag{A.12}$$

but $\int p\, lnf = \int f\, lnf$, since they yield the same moments (from the constraints), so

$$\begin{aligned} h(p(\mathbf{x})) &\leq h(f(\mathbf{x})) \\ &\leq \frac{1}{2} ln\left[(2\pi e)^n |\mathbf{P}|\right], \end{aligned} \tag{A.13}$$

using the result from Appendix A.1. Hence the normal maximizes the entropy for the given constraints.

A.3 On the relationship between Fisher information and Entropy

Entropy and Fisher information are related by the log-likelihood function. Fisher information results in a MMSE estimate of the state $\mathbf{x}$ in that it provides a lower bound on the error in the MMSE estimate

$$\text{MSE} \geq tr(\mathbf{J}^{-1}). \tag{A.14}$$

This can be seen from the fact that each diagonal element of the estimation error covariance P_{ii} is the mean-squared error of the estimate of x_i, that is; $P_{ii} = E\{(\hat{x}_i - x_i)^2\} \geq (\mathbf{J}^{-1})_{ii}$, by the Cramer-Rao bound. Scharf [121] shows that minimizing the estimated entropy is equivalent to an MMSE estimate. Another argument further strengthening the direct relation between entropy and Fisher information comes from consideration of the Asymptotic Equipartition Property (AEP)[39][1]. The AEP gives rise to the definition of the *typical set* A_ϵ^n. Formally, the typical set A_ϵ^n is the smallest volume set with probability $\geq 1 - \epsilon$ to the first order in the exponent. Informally, the typical set is the volume containing most of the probability. It can be shown that

$$Vol(A_\epsilon^n) \leq 2^{n(h+\epsilon)},$$

where h is the entropy associated with the corresponding PDF. From this we can write that $Vol(A_\epsilon^n) \approx 2^{nh}$. Whereas entropy can be seen to be related to the volume of the typical set, Fisher information can be shown [39] to be inversely related to the surface area of the typical set. And so maximization of one is equivalent to minimization of the other. Defined as in Chapter 2, entropy is always a scalar and Fisher information always a matrix for a vector $\mathbf{x}$.

[1] Cover [39] gives a detailed development of the AEP for discrete and continuous r.v.'s, we make use of it here for illustrative purposes only.

Appendix B

Differential Sonar Model Details

B.1 Sonar Physical Model

Range r to a feature (target) is obtained from

$$r = ct/2, \tag{B.1}$$

where c is the speed of sound in the transmission medium[1] and t is the measured time of flight. The receiving system detects the first echo which exceeds a threshold setting of the receiving circuit and ignores subsequent echoes. The principle is illustrated in Figure B.1 (a) which shows the observed time waveform. The impulse response can be analysed by separating the transmitter (T) and the receiver (R) and making the *far-field* approximation so that at the receiving element the spherical wavefront can be approximated by a plane wave. It can be shown using Huygens principle [85] that if angular inclination to the direction of propagation is α, the radius of the receiving aperture a and the range r, the time function of the impulse response for a receiving only (R) aperture is given by

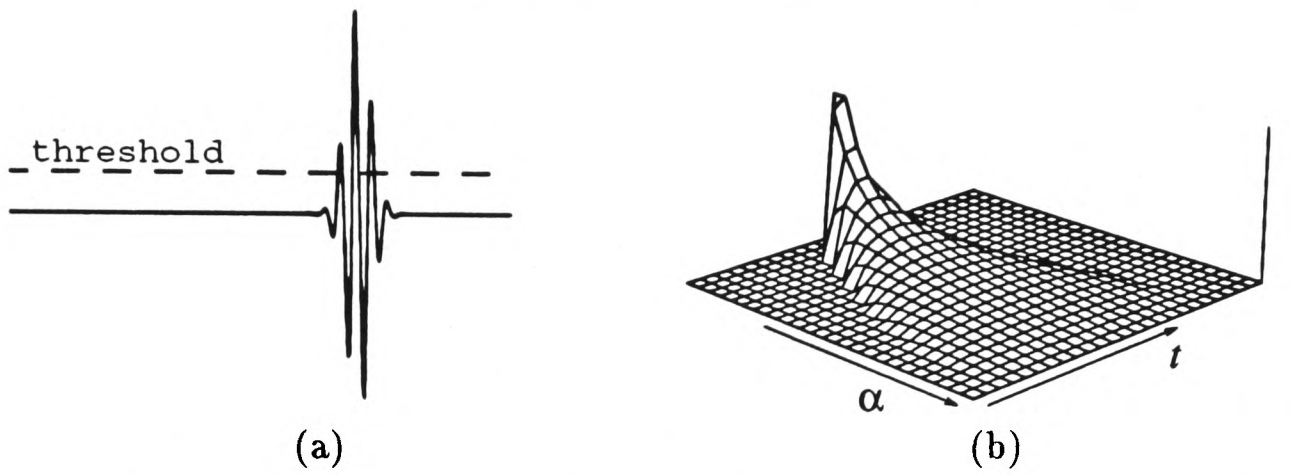
$$h_R(t, r, a, \alpha) = \begin{cases} k_1 \left(1 - \frac{c^2(t-2r/c)^2}{a^2 \sin^2 \alpha}\right)^{1/2}, & \text{for } \frac{2r - a \sin \alpha}{c} \leq t \leq \frac{2r + a \sin \alpha}{c} \\ & \text{and } 0 \leq \mid \alpha \mid \leq \pi/2 \\ \delta(t - 2r/c), & \text{for } \alpha = 0 \end{cases} \tag{B.2}$$

where

$$k_1 = \frac{2c \cos \alpha}{\pi \alpha \sin \alpha},$$

[1] In air this is 343.5 m/s at room temperature. This is also given by $\sqrt{\gamma RT}$

and δ is the Dirac Delta function [109]. The impulse response of Equation B.2 is illustrated



(a) (b)

Figure B.1: (a) The observed time waveform (b) The impulse response of the receiver. The vertical axis shows the amplitude.

in Figure B.1(b). The impulse response of the combination of a receiver and transmitter (T/R) is found to be a self convolution of the impulse response of the receiver h_R[85]

$$h_{T/R}(t, r, a, \alpha) = \int_{2r-a\ \sin\alpha/c}^{2r+a\ \sin\alpha/c} h_R(\tau, r, a, \alpha) * h_R(t-\tau, r, a, \alpha)\, d\tau. \qquad \text{(B.3)}$$

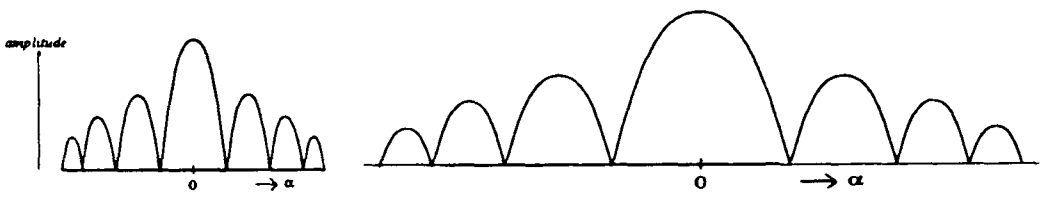
The amplitude of the response decreases and its duration increases with α as can be deduced from Figure B.1(b). Therefore a target will continue to be visible as α is increased up to the point where the amplitude of the response becomes less than the threshold of the receiving circuit. The maximum value of α permissible for detection of the echo α_{max} is dependent on the physical properties of the T/R device and the propagated wave. For edges and cylinders whose radius is very small compared to the range, the diffracted signal is more significant and so a factor $\sqrt{1/r}$ indicating the diminishing waveform due to diffraction with range r is included in the impulse response relationship. The beam has two distinct regions, the *Fresnel* (near) region and the *Fraunhofer* (far) region. The far region is defined by ranges $r > a^2/\lambda$. The beam emanating from the transducer can be described as shown by Morse and Ingard [104] in a rather complex analytical form. A simpler approximation

is a Gaussian envelope[15]

$$p(\theta) = p_{max}\, exp\left[\frac{-2\theta^2}{\theta_0^2}\right], \quad \text{where} \quad \theta_0 = \sin^{-1}\left(\frac{0.61c}{fa}\right), \tag{B.4}$$

for the -30dB level.

The effect of frequency on the angle α_{max} can be seen by considering the radiation characteristic of the beam pattern [88] emanating from the transducer given in Equation B.4. Figure B.2 shows the characteristic of Equation B.4 including the side-lobes. The figure illustrates the effect of decreasing the frequency of the transmitted signal. It can be seen that for a fixed threshold, the maximum α for reception is larger with a lower frequency f_2 than with a higher frequency f_1.



(a) f_1 (b) f_2

Figure B.2: The effect of decreasing frequency. In (a) and (b) $f_1 > f_2$.

B.2 Differential Feature Model

Plane Model. Consider the differential sonar facing in the first instance a plane as shown in Figure 5.5(a). In the figure α is the same as $\alpha_{T/R}$ in Figure 5.4. In the arrangement of Figure 5.5(a), a is the T/R device and b is the R device. For device a, a_t is the transmitter and a_r the receiver, for b we only have b_r. The differential is a consequence of the different path lengths. For reception at the T/R that is, at a_r the path length is given by $2p_1$. By considering the geometry of Figure 5.5(a) we have

$$2p_1 = 2\left(r - \frac{d}{2}\sin(\alpha)\right). \tag{B.5}$$

For reception at b_r, the path length is given by $p_2 + p_3$. From Figure 5.5 we have

$$p2 = \sqrt{\left(r - \frac{d}{2}\sin(\alpha)\right)^2 + \left(\frac{d}{2}\cos(\alpha)\right)^2} \tag{B.6}$$

$$p3 = \sqrt{\left(r + \frac{d}{2}\sin(\alpha)\right)^2 + \left(\frac{d}{2}\cos(\alpha)\right)^2}. \tag{B.7}$$

The differential for the plane, δ_p is the difference in the path lengths for the primary and secondary reception that is; $\delta_p = (p_2 + p_3) - 2p_1$. Simplifying the above expressions this can be written

$$\delta_p = \left(\sqrt{r^2 + \frac{d^2}{4} - r\,d\,\sin(\alpha)} + \sqrt{r^2 + \frac{d^2}{4} + r\,d\,\sin(\alpha)}\right) - 2\left(r - \frac{d}{2}\sin(\alpha)\right). \tag{B.8}$$

Corner Model. For the corner in Figure 5.5(b), notice the reversal of a_r and b_r in comparison with the case for the plane in Figure 5.5(a). For the reception by the T/R device, that is, at a_r the path length is given by $p_2 + p_3$ where

$$p_2 = p_3 = \sqrt{\left(r - \frac{d}{2}\sin(\alpha)\right)^2 + \left(\frac{d}{2}\cos(\alpha)\right)}. \tag{B.9}$$

For reception at R the path length is given by

$$p_0 + p_1 = \left(r - \frac{d}{2}\sin(\alpha)\right) + \left(r + \frac{d}{2}\sin(\alpha)\right) = 2r. \tag{B.10}$$

Thus the differential is given by the path difference as follows

$$\delta_c = (p_0 + p_1) - (p_2 + p_3) = 2\left[r - \sqrt{r^2 + \frac{d^2}{4} - rd\,\sin(\alpha)}\right]. \tag{B.11}$$

Edge Model. For an edge target the received signal is largely due to diffraction. We assume that the edge behaves as a rigid knife-edge[85] as shown in Figure 5.5(c). The virtual reflectors a_r and b_r coincide at the point of intersection as shown in the figure. For

reception by the T/R device, that is, at a_r the path is given by $2p_1$ and that at R by $p_1 + p_3$. From Figure 5.5(c) we have that

$$p_1 = \sqrt{\left(r - \frac{d}{2}\sin(\alpha)\right)^2 + \left(\frac{d}{2}\cos(\alpha)\right)^2}. \tag{B.12}$$

$$p_3 = \sqrt{\left(r + \frac{d}{2}\sin(\alpha)\right)^2 + \left(\frac{d}{2}\cos(\alpha)\right)^2}. \tag{B.13}$$

The differential is given by the difference in the path lengths $\delta_e = p_3 - p_1$. Simplifying this using Equation B.12 and Equation B.13 gives

$$\delta_e = \sqrt{r^2 + \frac{d^2}{4} - rd\sin(\alpha)} - \sqrt{r^2 + \frac{d^2}{4} + rd\sin(\alpha)}. \tag{B.14}$$

Cylinder Model. The path length for reception at the T/R, that is, at a_r is given by $2p_1$. And that at R is given by $p_2 + p_3$. From the geometry of Figure 5.5(d), path p_1 is given by

$$p_1 = \sqrt{\left(\frac{d}{2}\cos(\alpha)\right)^2 + \left(r - \frac{d}{2}\sin(\alpha) + p_r\right)^2} - p_r \tag{B.15}$$

where p_r is the radius of the cylinder. Also from Figure 5.5(d), paths p_2 and p_3 are given by

$$p_2 = \sqrt{\left(r - \frac{d}{2}\sin(\alpha)\right)^2 + \left(\frac{d}{2}\cos(\alpha)\right)^2}, \tag{B.16}$$

$$p_3 = \sqrt{\left(r - \frac{d}{2}\sin(\alpha)\right)^2 + \left(\frac{d}{2}\cos(\alpha)\right)^2}, \tag{B.17}$$

respectively. Thus the path difference is given by

$$\begin{aligned}\delta_{cyl} = (p_2 + p_3) - 2p_1 = &\left(\sqrt{r^2 + \frac{d^2}{4} - r\,d\sin(\alpha)} + \sqrt{r^2 + \frac{d^2}{4} + r\,d\sin(\alpha)}\right) \\ &-2\left(\sqrt{\left(\frac{d}{2}\cos(\alpha)\right)^2 + \left(r - \frac{d}{2}sin(\alpha) + p_r\right)^2} - p_r\right)\end{aligned} \tag{B.18}$$

The differential for the cylinder approximates to that for the plane as p_r goes to infinity. This can be appreciated from a consideration of p_1 in Figure 5.5(d) as given by Equation B.15 that

$$p_1 \overset{p_r \to \infty}{\longrightarrow} \left(r - \frac{d}{2} \sin(\alpha) \right), \tag{B.19}$$

which is the p_1 for a plane (From Equation B.5). Since p_2 and p_3 are not affected by changes in p_r, Equation B.18 and hence the differential, approximates to that for the plane. When the radius becomes small, p_1 as given by Equation B.15 becomes

$$p_1 \overset{p_r \to 0}{\longrightarrow} \sqrt{\left(\frac{d}{2} \cos(\alpha) \right)^2 + \left(r - \frac{d}{2} \sin(\alpha) \right)^2}, \tag{B.20}$$

which is the same as p_1 for the edge (Equation B.12). Hence as the radius of the cylinder goes to zero the differential of Equation B.18 approximates to that for an edge. Thus for any cylinder the true differential lies between that for the edge and that for the plane.

Variations in the theoretical differential

In the following comparisons of the theoretical differential, the only region of concern is that within the bounds of an RCD. Figure B.3 shows that for the plane, increasing the base-line d increases the gradient of the differential and also introduces an offset from the *true* centre of the RCD. Figure B.3(a) shows that decreasing the range with a fixed base-line increases the offset away from the *true* RCD centre in the *negative* direction. The offset with a range of $0.45m$ is approximately $-0.017rad.$ The offset can be explained by a consideration of Figure 5.4 where for an angle $\alpha = 0$, there is still a difference in the path lengths due to the asymmetry of the differential arrangement. For the corner, increasing the base-line also increases the gradient but the offset is in the *positive* direction as shown in Figure B.3(c). The phase reversal due to the double reflection accounts for the direction of the offset.

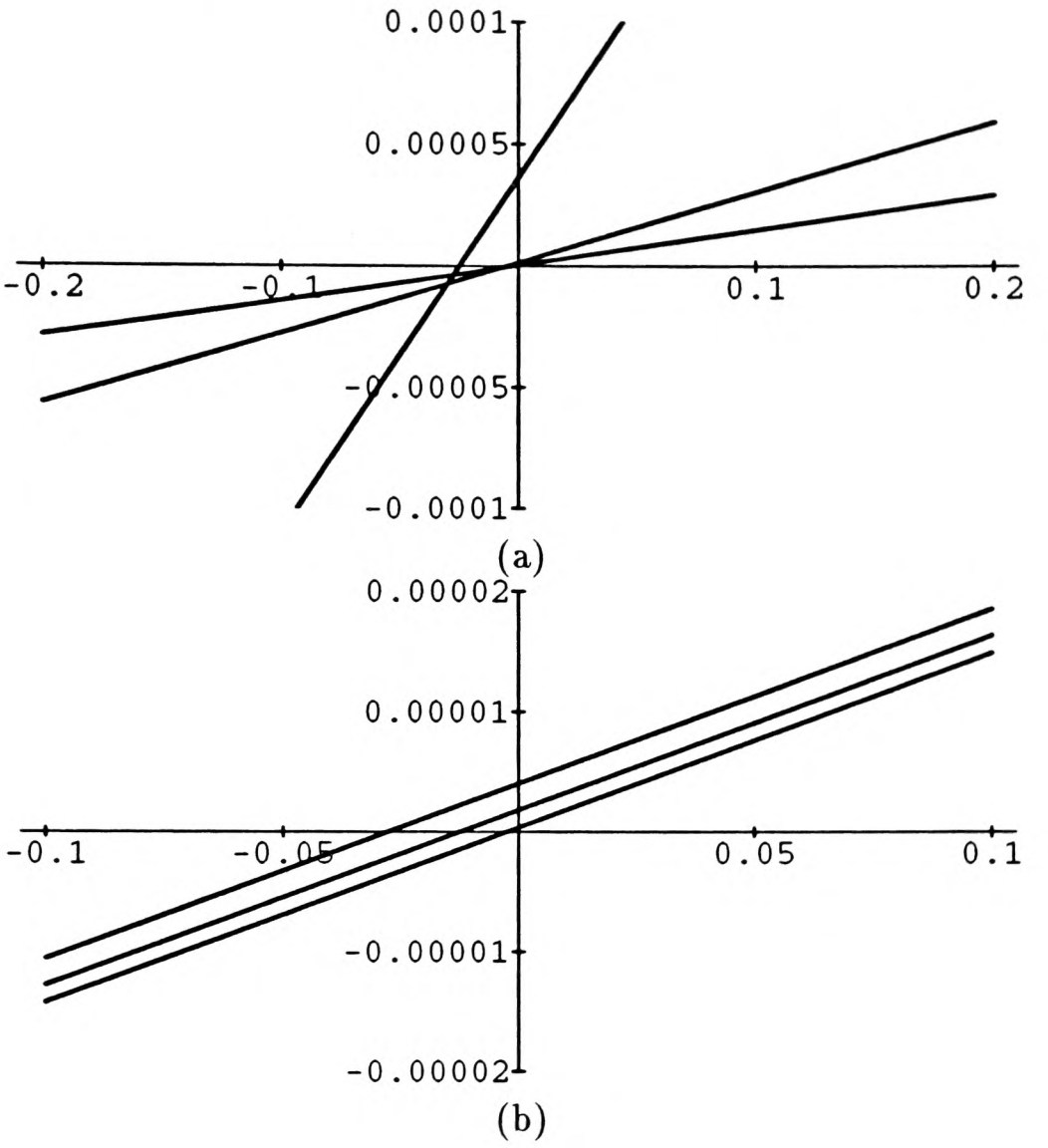


Figure B.3: Plot shows the differential plotted against angle for a given plane RCD. (a) The plot illustrates variations of Δ_p with d for values of d = 0.05m, 0.1m, 0.5m, r is fixed at 2m. The plot shows that increasing d increases the gradient of the differential. (b) The plot illustrates variations of Plot for Δ_p showing variations with r for values of $r = 0.45m$, $1m$, $5m$, d is fixed at $0.05m$. Decreasing the range r increases the offset from the true centre of the RCD.

The differential for the edge varies with base-line in a similar way to that for the plane and the corner but has no offset irrespective of the values of r and d. Because the cylindrical feature has a differential which varies between that of the plane and that of the edge, variations on the differential are similar to those of the feature which the cylinder by virtue of its dimensions, closely approximates.

Choice of Base-line. Other than physical compactness resulting from a small d, there

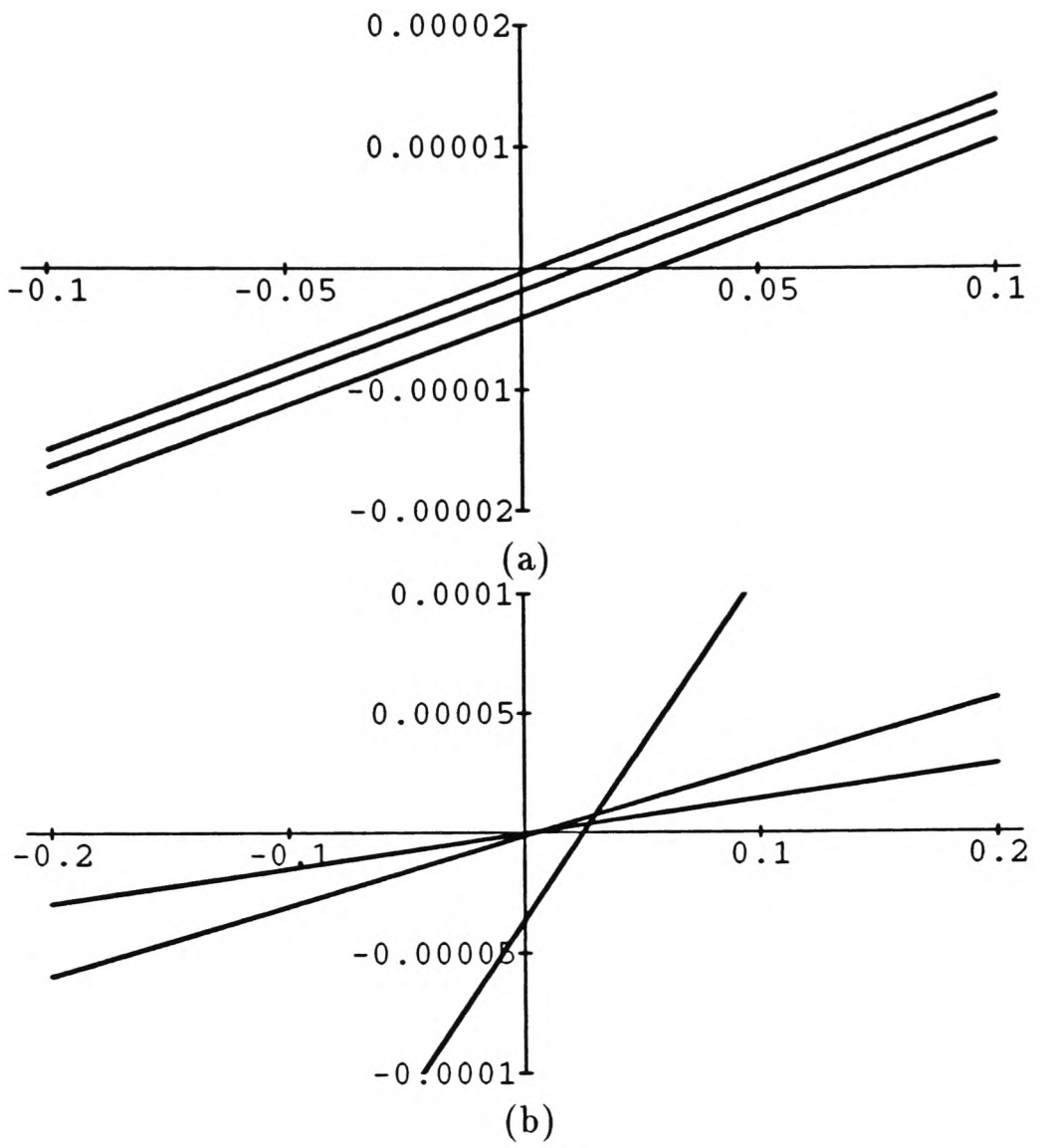


Figure B.4: Plot shows the differential plotted against angle for a given corner RCD. (a)(c)Plot for Δ_c showing variations with r for values of $r = 0.45m$, $1m$, $5m$, d is fixed at $0.05m$. Decreasing the range r increases the offset from the true centre of the RCD. (b) Plot for Δ_c showing variations with d. Values of $d = 0.05m$, $0.1m$, $0.5m$, r is fixed at $2m$. The plot shows that increasing d increases the gradient of the differential.

are other considerations in the choice of base-line. A very small d results in a very small gradient for the differential thus we need to reconcile by way of compromise, the range of maximum differential TOF to be measured, our hardware timing capabilities and the resulting timing discrimination. Consider Table B.1 for Δ_p in the range $0.45m - 3m$. It can be seen from that a small base-line is desirable because it gives a minimum offset for the minimum range and hence more accurately gives the true centre over the working range.

base-line d (m)	range r (m)	max differential TOF (μs)	centre offset (rad)
0.50	0.45	500	0.244
	3.00	67	0.041
0.30	0.45	209	0.157
	3.00	54	0.024
0.20	0.45	115	0.105
	3.00	41	0.017
0.10	0.45	43	0.054
	3.00	23	0.009
0.075	0.45	27	0.042
	3.00	17	0.007
0.05	0.45	16	0.028
	3.00	12	0.003

Table B.1: Pertinent values in choosing a base-line d for a range of $0.45 - 3.00m$. The table is for an RCD of width $0.17rad$.

Variation of range and orientation variances while tracking. Within the working range of $0.045 - 4.00m$, for a given set of sensor parameters ζ, the variances $\sigma_r^2$and σ_θ^2vary with the observations themselves. The modification of the nominal variances determined by ζ can be modelled by functions which depend on the r and θ. These factors can be determined empirically by obtaining several results such as those of Table 5.1 at various values of r and θ and finding functions which approximately model the variations observed. With this approach we approximate the dependence of the nominal values on observations using the functions

$$g_r(r) = 1 - 2(\sqrt{r})e^{-r}, \quad g_\theta(\theta) = (1 - e^{\frac{-(\theta-2)^2}{5}}). \tag{B.21}$$

for the $\sigma_r^2$and σ_θ^2respectively. These are then used in Equation 5.22. In this way the covariance $\mathbf{R}[k, \mathbf{z}(k)]$ becomes a *location covariance.*

Appendix C

Sensor Hardware and the OxNav Vehicle

C.1 Monopulse Sonar

The monopulse sonar unit provides the following; [r, θ, $\mid \Delta \mid$, $sign(\Delta)$]. These values are obtained as follows:

- **Range r.** The range of obtained by measuring the TOF as usual. The range can be resolved to an accuracy of approximately 1%.

- **Orientation θ.** The orientation is obtained from the direction in which the servo points the two Polaroids. With our present hardware the positioning of the servo is very crude and smallest angle resolved is 2.7^o.

- **Differential $\mid \Delta \mid$.** The differential TOF is obtained by measuring the elapsed time between the primary reception and secondary reception. The differential TOF can be resolved to an accuracy of $3.2\mu s$.

- $Sign(\Delta)$. The sign of the differential TOF is given by a bit which notes the order of reception.

The above specification is clearly very crude and this is has been to the detriment of the results obtained. An improved design was completed at the time of writing, which offered improvements of at least an order of magnitude in all the above.

C.2 Decentralized Sensor Node

We have developed a generic architecture for sensor nodes in a decentralized architecture. Each sensor node is built around a Transputer with all the requisite communication links and interface logic necessary. Figure C.1 shows the architecture for such a sensor node. The architecture features:

- TRAM2/T800 Transputer module, with full floating-point ALU.
- 4 External Communication links, with data rates of 10/20 MBits per second.
- Input Output handled by C011 serial to parallel interface chips.

The generic nature of this architecture is evidenced by its use in the other work done at Oxford reported in [14][63].

C.3 OxNav Vehicle (JTR)

The vehicle used in the work described in this thesis is the OxNav Vehicle JTR purpose built at Oxford. The objectives of this vehicle project can be summarized as follows:

- Development of a sensor suite for navigation.
- Development of a suitable modular vehicle for indoor environments.
- Development of robust navigation algorithms.

The vehicle is shown in Figure C.2. The vehicle in addition to decentralized multi-sensor fusion, features decentralized kinematic control. The vehicle was designed primarily by Burke [26]. The complete sensor suite includes; Tracking Sonars, infra-red device, a complete inertial navigation system.

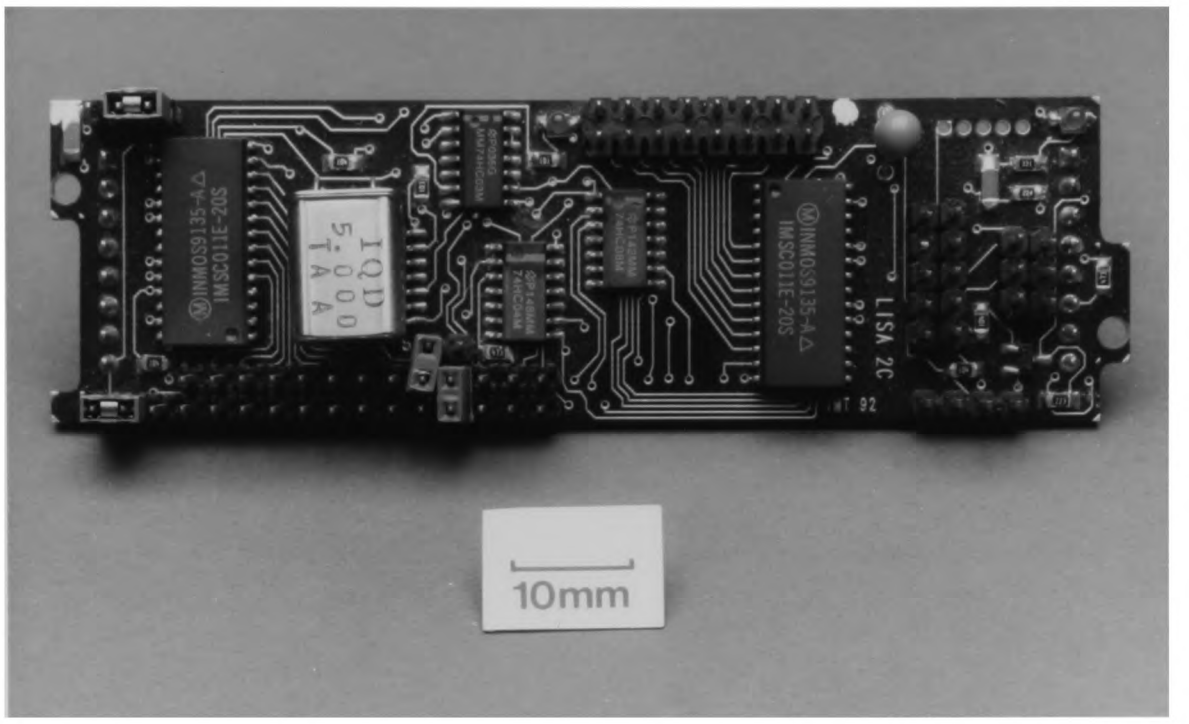


Figure C.1: Generic Transputer-based decentralized sensor node hardware.

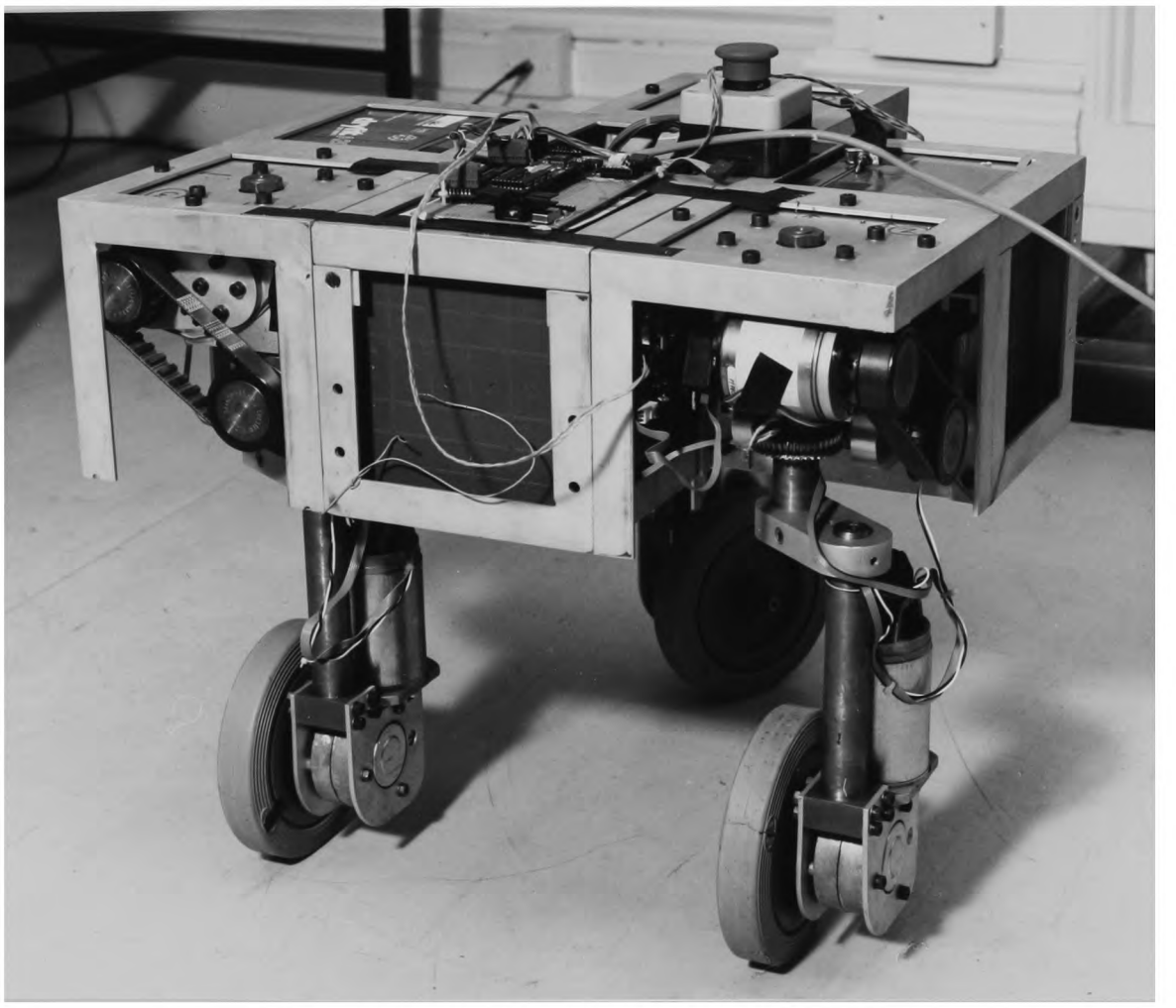


Figure C.2: The OxNav Vehicle JTR.

Bibliography

[1] Abidi and Gonzalez. Introduction. In *Data Fusion In Robotics And Machine Intelligence*, pages 1–6. Academic Press, 1992.

[2] J. Alexander and J. Maddocks. On the kinematics of wheeled mobile robots. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Velag, 1990.

[3] P. Allen and R. Bajcsy. Two sensors are better than one: Example of vision and touch. In *Third Int. Symp. Robotics Research*, pages 48–55, Gouvieux, France, 1986. MIT Press.

[4] A. T. Alouani. Nonlinear data fusion. In *Proc. 29th CDC*, pages 569–572, Tampa, December 1989.

[5] R. L. Andersson. *A Robot Ping-Pong Player*. MIT Press, 1988.

[6] F.J. Anscombe and R.J. Aumann. A definition of subjective probability. *Ann. Math. Statist*, 34:199–205, 1963.

[7] K.J. Arrow. *Social Choice and Individual Values 2nd Ed.* Wiley, New York, 1966.

[8] M. Bacharach. Group decisions in the face of differences of opinion. *Management Science*, 22(2):182–191, 1975.

[9] M. Bacharach. Normal bayesian dialogues. *J. American Statistical Soc.*, 74:837–846, 1979.

[10] J.G. Balchen and F. Dessen. Structural solution of highly redundant sensing in robotic systems. In *Highly Redundant Sensing in Robotic Systems*, pages 264–275. Springer-Velag, Berlin, 1990.

[11] Y. Bar-Shalom. *Multi-Target Multi-Sensor Tracking*. Artec House, 1990.

[12] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[13] B. Barshan and H.F. Durrant-Whyte. Evaluation of a solid-state gyroscope for robotics applications. In *Submitted to IEEE Int. Conf. Robotics and Automation, Atlanta, U.S.A.*, 1993.

[14] B. Barshan and H.F. Durrant-Whyte. Inertial navigation system for a mobile robot. In *To appear 1st IFAC International Workshop on Intelligent Autonomous Vehicles, Southampton, U.K.*, 1993.

[15] B. Barshan and R. Kuc. Differentiating sonar refelctions from corners and planes by employing an intelligent sensor. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(6):560–569, 1990.

[16] O. Basir and H. Shen. A new approach for aggregating multisensory data. Technical report, Dept of Systems Design Eng. University of Waterloo, Ontario, 1992.

[17] T. Berg and H.F. Durrant-Whyte. Model distribution in decentralized multi-sensor fusion. In *Proc. American Control Conference*, pages 2292–2294, 1991.

[18] J.O. Berger. *Statistical Decisions (second edition).* Springer Verlag, Berlin, GDR, 1985.

[19] S.S. Blackman. *Multiple Target Tracking with Applications to Radar.* Artec House, 1986.

[20] S.S. Blackman. Multitarget tracking with agile beam radar. In Y. Bar-Shalom, editor, *Multi-Target Multi-Sensor Tracking*, pages 237–268. Artech House, 1992.

[21] David Blackwell and M. A. Girshick. *Theory of Games and Statistical Decisions.* Dover Publications, 1954.

[22] J. Borenstein and Y. Koren. Obstacle avoidance with ultrasonic sensors. *IEEE J. Robotics and Automation*, 4:213–218, 1988.

[23] J.M. Brady, S. Cameron, H. Durrant-Whyte, M. Fleck, D. Forsyth, A. Noble, and I. Page. Progress towards a system that can acquire pallets and clean warehouses. In *Fourth Int. Symp. Robotics Research*, Santa Cruz, August 1987.

[24] C.M. Brown, H. Durrant-Whyte, J. Leonard, and B. Rao. Centralized and decentralized kalman filter techniques for tracking, navigation and control. In *Proc. DARPA Image Understanding Workshop*, pages 651–676. Morgan Kaufmann Inc., 1989.

[25] R.G. Brown and P. Hwang. *Introduction to Random Signals and Aplied Kalman Filtering (2nd Ed).* Wiley, 1992.

[26] T. Burke and H.F. Durrant-Whyte. Modular mobile robot design. In *1st IFAC International Workshop on Intelligent Autonomous Vehicles, Southampton, U.K.*, 1993.

[27] A. Cameron and H. Durrant-Whyte. Optimal sensor placement. *Int. J. Robotics Research*, 9(3), 1990.

[28] D.A. Castanon and D. Teneketzis. Distributed estimation algorithms for nonlinear systems. *IEEE Trans. Automatic Control*, 30(5):418–425, 1985.

[29] D. Catlin. *Estimation, Control, and the Discrete Kalman Filter.* Springer-Verlag, 1989.

[30] Z. Chair and P.K. Varshney. Optimal data fusion in multiple sensor detection systems. *IEEE Trans. Aerospace and Electronic Systems*, 22:98–101, 1986.

[31] Z. Chair and P.K. Varshney. Distributed bayesian hypothesis testing with distributed data fusion. *IEEE Trans. Systems Man and Cybernetics*, 18(5):695–699, 1988.

[32] K.C. Chang, C.Y. Chong, and Y. Bar-Shalom. Joint probabalistic data association in distributed sensor networks. *IEEE Trans. Automatic Control*, 31(10):889–897, 1986.

[33] H. Chernoff and L. Moses. *Elementary Decision Theory.* Dover, 1959.

[34] C. Chong. Hierarchical estimation. In *2nd MIT/ONR CCC Workshop*, Monterey CA, 1979.

[35] C. Chong, S. Mori, and K. Chan. Distributed mutitarget multisensor tracking. In Y. Bar-Shalom, editor, *Multitarget Multisensor Tracking.* Artech House, 1990.

[36] A.G. Cohn. Approaches to qualitatitive reasoning. *Artificial Intelligence Review*, 3:177–232, 1989.

[37] J. Colly, M. Devy, and M. Ghallab. Sensory control for 3D environment modelling. In *Proc. IARP Conf. on Multi-Sensor Fusion*, 1989.

[38] Polaroid Corporation and Commercial Battery Division. *Ultrasonic ranging system.* 1984.

[39] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications, 1991.

[40] I.J. Cox. Blanche: An autonomous robot vehicle for structured environments. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 978–982, 1988.

[41] J. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 674–681, 1989.

[42] K.P.Dunn D. Willner, C.B.Chang. Kalman filter algorithm for a multi-sensor system. In *15th IEEE Conf. Decision Control*, Clearwater, Florida, 1976.

[43] M.H. DeGroot. *Optimal Statistical Decisions.* McGraw-Hill, New York, 1974.

[44] M.H. DeGroot. Reaching a consensus. *J. of the American Statistical Assoc.*, 64(345):118–121, 1974.

[45] M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(2):325–332, 1987.

[46] H.F. Durrant-Whyte. *Integration, Coordination, and Control of Multi-Sensor Robot Systems.* Kluwer Academic Press, Boston, MA., 1987.

[47] H.F. Durrant-Whyte. Sensor models and multi-sensor integration. *Int. J. Robotics Research*, 7(6):97–113, 1988.

[48] H.F. Durrant-Whyte. Multi-sensor data fusion for (semi-) autonomous robots. *unpublished*, 1990.

[49] H.F. Durrant-Whyte, S. Grime, and H. Hu. A modular, Transputer-based architecture for multi-sensor data fusion. In *Proc. Second Int. Conf. Applications of Transputers*, pages 71–77, 1990.

[50] H.F. Durrant-Whyte, B.Y. Rao, and H. Hu. Toward a fully decentralized architecture for multi-sensor data-fusion. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1331–1336, 1990.

[51] A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 727, 1988.

[52] O. Faugeras and N. Ayache. Building visual maps by combining noisy stereo measurements. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1433–1438, San Fransico, U.S.A., 1986.

[53] O. D. Faugeras, M. Hebert, and E. Pauchon. Segmentation of range data into planar and quadratic patches. In *Int. Conf. Computer Vision and Pattern Recognition*, pages 8–13, 1983.

[54] T.S. Ferguson. *Mathematical Statistics: ADecision Theoretic Approach.* New York, Academic Press, 1967.

[55] P.C. Fishburn. Subjective expected utility: a review of normative theories. *Theory and Decision*, (13):139–199, 1981.

[56] W. Fleskes and G. Van Keuk. Adaptive control and tracking with elra phased array radar experimental system. *IEEE Radar-80 Conf. Rec. June*, 1980.

[57] A.M. Flynn. Combining ultra-sonic and infra-red sensors for mobile robot navigation. *Int. J. Robotics Research*, 7(5):5–14, 1988.

[58] T.D. Garvey, J.D. Lawrence, and M.A. Fischler. An inference technique for intergrating knowledge from disparate sources. In *7th Int. Joint Conf. Artificial Intelligence, Vancouver*, pages 319–325, 1981.

[59] M. Gelb. *Applied Optimal Estimation.* MIT press, 1974.

[60] G. Giralt. Research trends in decisional and multi-sensory aspects of third generational robots. In *Second Int. Symp. Robotics Research*, Koyoto, Japan, 1984. MIT Press.

[61] I.J. Good. Twenty-seven principles of rationality. In V.P. Godambe and D.A. Sprott, editors, *Foundations of statistical inference.* Holt, Rhinehart, Winston, 1971.

[62] S. Grime. *Communication in Decentralized Sensing Architectures.* PhD thesis, Oxford University, U.K., 1992.

[63] S. Grime, H.F. Durrant-Whyte, and P. Ho. Communication in decentralized sensing. In *Proc. American Control Conference*, 1992.

[64] G. Hager. *Active Reduction of Uncertainty in Multi-Sensor Systems.* PhD thesis, University of Pensylvania, 1988.

[65] G. Hager. *Task Directed Sensor Fusion and Planning.* Kluwer Academic, Boston MA, 1990.

[66] G. Hager and H.F. Durrant-Whyte. Information and multi-sensor coordination. In *Uncertainty in Artificial Intelligence 2*, pages 381–394. North Holland, 1988.

[67] G. Hager and M. Mintz. Computational methods for task-directed sensor data fusion and sensor planning. *Int. J. Robotics Research*, 10(4):285–313, 1991.

[68] E. Hanle. Control of a phased array radar for position finding of targets. *IEEE Radar-75 Conf. Rec*, 1975.

[69] H.R. Hashemipour, S. Roy, and A.J. Laub. Decentralized structures for parallel kalman filtering. *IEEE Trans. Automatic Control*, 33(1):88–93, 1988.

[70] T. Henderson and C. Hansen. The specification of distributed sensing and control. *J. of Robotic Systems*, 2(4):387–396, 1985.

[71] W.D. Hillis. A high-resolution imaging touch sensor. *Int. J. Robotics Research*, 1(2):33–44, 1982.

[72] P. Ho. Organization in distributed sensing. *First year report. Robotics Research Group. Oxford University*, 1991.

[73] Y.C. Ho. Team decision theory and information structures. *Proceedings of the IEEE*, 68:644, 1980.

[74] S.A. Hovanessian. *Introduction to Sensor Systems.* Artech House, 1988.

[75] H. Hu and P.J. Probert. Distributed architectures for sensing and control in obstacle avoidance for autonomous vehicles. In *IARP Int. Conf. Multi-Sensor Data Fusion*, 1989.

[76] K. Ikeuchi and T. Kanade. Automatic generation of object recognition programs. *IEEE Proc*, 76(8):1016–1035, 1988.

[77] S. Iyenger, R. Kashyap, and R. Madan. Distributed sensor networks. *IEEE Trans. Systems Man and Cybernetics*, 21(5), 1991.

[78] H. Jeffreys. *Theory of Probability(3rd Ed).* Oxford University Press, UK, 1961.

[79] G.M. Jenkins and D.G. Watts. *Spectral Analysis and its Applications.* Holden-Day Inc: San Francisco, 1968.

[80] W.B. Davenport Jr. *Probability and Random Processes.* McGraw-Hill, New York, 1970.

[81] E. Kalai. Nonsymmetric nash solutions and replications of 2-person bargaining. *Int.J.Game.Theory*, 6:129–133, 1977.

[82] R.L. Keeny and H. Raiffa. *Decisions with multiple objectives.* Wiley, New York, 1976.

[83] T. Kerr. Decentralised filtering and redundancy management for multisensor navigation. *IEEE Trans. Aerospace and Electronic Systems*, 23(1):83–119, 1987.

[84] R. Kuc and B. Barshan. Navigating vehicles through an unstructured environment. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1422–1426, 1989.

[85] R. Kuc and M. W. Siegel. Physically based simulation model for acoustic sensor robot navigation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(6):766–778, 1987.

[86] R.M. Kuczewski. Neural network approaches to multi-target tracking. In *IEEE First International Conference of Neural Networks*, 1987.

[87] E.L. Lehmann. *Testing Statistical Hypothesis 2nd Ed (1st Ed 1959).* Wiley, New York, 1985.

[88] J.J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation.* PhD thesis, University of Oxford, 1991.

[89] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE Int. Conf. on Intelligent Robot Systems (IROS)*, 1991.

[90] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Navigation.* Kluwer Academic Press, 1992.

[91] R. D. Luce and H. Raiffa. *Games and Decisions.* Wiley, 1957.

[92] R. Luo. Data fusion and sensor integration: State-of-the-art 1990s. In *Data Fusion In Robotics And Machine Intelligence*, pages 7–136. Academic Press, 1992.

[93] R.C. Luo and M.G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Trans. Systems Man and Cybernetics*, 19(5):901–931, 1989.

[94] J. Manyika and H. Durrant-Whyte. Information as a basis for management and control in decentralized fusion architectures. In *Proc. IEEE Conf. Decision and Control, Tuscon*, 1992.

[95] J. Manyika, S. Grime, and H. Durrant-Whyte. A formaly specified decentralized architecture for multi-sensor data fusion. In *Transputing '91*, pages 609–628. IOS press, 1991.

[96] J.M. Manyika and H.F. Durrant-Whyte. An information-theoretic approach to management in decentralized data fusion. In *In Proc. Spie 92 Conference. Vol.1828*, 1992.

[97] J.M. Manyika and H.F. Durrant-Whyte. Navigation using tracking sonars. Technical Report 1956/92, Oxford U. Robotics Research Group, 1992.

[98] J.M. Manyika and H.F. Durrant-Whyte. A tracking sonar for vehicle guidance. In *In Proc. IEEE Robotics and Automation*, 1993.

[99] P.S. Maybeck. *Stochastic Models, Estimaton and Control, Vol. I.* Academic Press, 1979.

[100] R. McKendall and M. Mintz. Data fusion techniques using robust statistics. In *Data Fusion In Robotics And Machine Intelligence*, pages 211–244. Academic Press, 1992.

[101] R.J. McKenzie and D.G. Mullens. Expert system control for airborne radar surveillance. *American Institute of Aeronautics and Astronautics*, pages 87–2854, 1987.

[102] A. Mitche and J.K. Aggarwal. Multiple sensor integration through image processing: A review. *Optical Engineering*, 23(2):380, 1986.

[103] H. Moravec. Sensor fusion in certainty grids for mobile robots. In *Sensor Devices and Systems for Robotics*, pages 253–276. Springer-Velag. Nato ASI Series, 1989.

[104] P.M. Morse and K.U. Ingard. *Theoretical Acoustics.* New York: McGraw-Hill, 1968.

[105] N. Nandhakumar and J. Aggarwal. Integrating information from thermal and visual images for scene analysis. In *Proc. SPIE Conf. Applications of Artificial Intelligence*, pages 132–142, 1986.

[106] J.F. Nash. The bargaining problem. *Econometrica*, page 155, 1950.

[107] J.M. Nash. Optimal allocation of tracking resources. In *IEEE Int. Conf. Decision and Control*, pages 1177–1180, 1977.

[108] J. Von Neumann and O. Morgenstein. *Theory of Games and Economic Behaviour.* Princeton University Press. 2nd Ed, 1947.

[109] A. Oppenheim and R. Schafer. *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

[110] N.E. Orlando. An intelligent robotics control scheme. In *American Control Conference*, page 204, 1984.

[111] N. Pal and S. Pal. Entropy, a new definition and its applications. *IEEE Trans. Systems Man and Cybernetics*, 21(5):1260–1270, 1991.

[112] A. Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill International Editions, 2nd Ed, 1984.

[113] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausable Inference.* Morgan Kaufmann Publishers Inc., 1988.

[114] R. Popoli. The sensor management imperative. In Y. Bar-Shalom, editor, *Multi-Target Multi-Sensor Tracking*, pages 325–392. Artech House, 1992.

[115] B.S.Y Rao. *Data Fusion Methods for Decentralized Sensing Systems.* PhD thesis, Oxford University, U.K., 1991.

[116] B.S.Y. Rao and H.F. Durrant-Whyte. A fully decentralized algorithm for multi-sensor kalman filtering. *IEE Transactions Schedule D*, 138(5):413–420, 1991.

[117] B.Y.S. Rao, J.M. Manyika, and H.F. Durrant-Whyte. Decentralized algorithms and architecture for tracking and identification. In *In. Vol 2. IEEE Conference on Intelligent Robots and Systems*, 1991.

[118] J.M. Richardson and K.A. Marsh. Fusion of multi-sensor data. *Int. J. Robotics Research*, 7(6):78–96, 1988.

[119] S. Russell and E. Wefald. *Do the right thing.* MIT Press, 1991.

[120] L.J. Savage. *The Foundations of Statistics.* Wiley, New York, 1954.

[121] L.L. Scharf. *Statistical Signal Processing.* Addison Wesley, 1991.

[122] M.J. Sewell. *Maximum and minimum principles.* Cambridge University Press, 1987.

[123] C. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, (27):379–423, 1948.

[124] H. Simon. Theories of bounded rationality. In C. McGuire and R. Radner, editors, *Decision and Organization*, pages 161–176. North-Holland Publishing Co. Amsterdam, 1972.

[125] R.A. Singer and A.J. Kanyuck. Computer control of multiple site correlation. *Automatica*, 7(6):455–463, 1971.

[126] M. Skolnik. *Introduction to Radar Systems.* McGraw-Hill International Editions, 1981.

[127] R.C. Smith and P. Cheesman. On the representation of spatial uncertainty. *Int. J. Robotics Research*, 5(4):56–68, 1987.

[128] J.L. Speyer. Communication and transmissio requirments for a decentralized linear-quadratic-gaussian control problem. *IEEE Trans. Automatic Control*, 24(2):266–269, 1979.

[129] C. Stirling and D. Morrell. Convex bayes decision theory. *IEEE Trans. Systems Man and Cybernetics*, pages 173–182, 1991.

[130] M. Stone. The opinion pool. *The Annals of Statistics*, 32:1339–1342, 1961.

[131] D. Terzopolous. Integrating visual information from multiple sources. In A.P. Pentland, editor, *From Pixels to Predicates*. Ablex Press, 1986.

[132] J.N. Tsitsiklis and M. Athans. On the complexity of decentralised decision-making and detection problems. *IEEE Trans. Automatic Control*, 30(5):440–446, 1985.

[133] E.L. Waltz and J. Llinas. *Multi-Sensor Data Fusion*. Artec House, 1991.

[134] S. Weerahandi and J.V. Zidek. Elements of multi-bayesian decision theory. *The Annals of Statistics*, 11(4):1032–1046, 1983.

[135] L. Weinberg. Scheduling multifunction radar systems. *IEEE Eastcon 77 Record*, 1977.

[136] A.S. Willsky, M.G. Bello, and D.A. Caston. Combining and updating of local estimates and regional maps along sets of one-dimensional tracks. *IEEE Trans. Automatic Control*, 27(4):799–812, 1982.

[137] L.A. Zadeh. A simple view of the dempster-shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(3):85–90, 1986.

[138] J. Zidek. Multi-bayesianity. Technical Report No.05, Dept. of Statistics, University of British Columbia, 1984.