UNIVERSITY OF SOUTHAMPTON

Multiple-Objective Sensor Management and Optimisation

by

Scott F. Page

A thesis submitted in partial fulfilment of the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

June 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Scott F. Page

One of the key challenges associated with exploiting modern Autonomous Vehicle technology for military surveillance tasks is the development of Sensor Management strategies which maximise the performance of the on-board Data-Fusion systems. The focus of this thesis is the development of Sensor Management algorithms which aim to optimise target tracking processes. Three principal theoretical and analytical contributions are presented which are related to the manner in which such problems are formulated and subsequently solved.

Firstly, the trade-offs between optimising target tracking and other system-level objectives relating to expected operating lifetime are explored in an autonomous ground sensor scenario. This is achieved by modelling the observer trajectory control design as a probabilistic, information–theoretic, multiple-objective optimisation problem. This novel approach explores the relationships between the changes in sensor-target geometry that are induced by tracking performance measures and those relating to power consumption. This culminates in a novel observer trajectory control algorithm based on the minimax approach.

The second contribution is an analysis of the propagation of error through a limited-lookahead sensor control feedback loop. In the last decade, it has been shown that the use of such non-myopic (multiple-step) planning strategies can lead to superior performance in many Sensor Management scenarios. However, relatively little is known about the performance of strategies which use different horizon lengths. It is shown that, in the general case, planning performance is a function of the length of the horizon over which the optimisation is performed. While increasing the horizon maximises the chances of achieving global optimality, by revealing information about the substructure of the decision space, it also increases the impact of any prediction error, approximations, or unforeseen risk present within the scenario. These competing mechanisms are demonstrated using an example tracking problem. This provides the motivation for a novel sensor control methodology that employs an adaptive length optimisation horizon. A route to selecting the optimal horizon size is proposed, based on a new non-myopic risk equilibrium which identifies the point where the two competing mechanisms are balanced.

The third area of contribution concerns the development of a number of novel optimisation algorithms aimed at solving the resulting sequential decision making problems. These problems are typically solved using stochastic search methods such as Genetic Algorithms or Simulated Annealing. The techniques presented in this thesis are extensions of the recently proposed Repeated Weighted Boosting Search algorithm. In its original form, it is only applicable to continuous, single-objective, optimisation problems. The extensions facilitate application to mixed search spaces and Pareto multiple-objective problems. The resulting algorithms have performance comparable with Genetic Algorithm variants, and offer a number of advantages such as ease of implementation and limited tuning requirements.

Contents

List of Figures

Declaration of Authorship

I, Scott F. Page, declare that this thesis entitled 'Multiple-Objective Sensor Management and Optimisation' and the work presented in this thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at the University of Southampton;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given, and with the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- and, parts of this work have been published, as described in Section 1.4.

Signed:.....

Date:.....

Glossary of Terms

ACFR	Australian Centre for Field Robotics
ADP	Adaptive Dynamic Programming
APM	Alternative Performance Measure
AV	Autonomous Vehicle
CRLB	Crámer-Rao Lower Bound
DARPA	Defense Advanced Research Projects Agency
DEIF	Decentralised Extended Kalman Filter
DF	Data-Fusion
DIF	Decentralised Information Filter
DP	Dynamic Programming
EHRWBS	Embedded Hull Repeated Weighted Boosting Search
EIF	Extended Information Filter
EKF	Extended Kalman Filter
EO	Electro-Optical
FI	Fisher Information
FIM	Fisher Information Matrix
FISST	Finite-Set-Statistics
FSPM	Fusion System Performance Measure
FST	Fusion System Task
GA	Genetic Algorithm
GOF	Geometric Objective Function
IF	Information Filter
JDL	Joint Directors of Laboratories
JPDAF	Joint Probabilistic Data-Association Filter
KF	Kalman Filter
Laser	Light Amplification by Stimulated Emission of Radiation
MAP	Maximum A Posteriori
MDP	Markov Decision Process
MC	Monte-Carlo
MHT	Multiple Hypothesis Tracker
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error

MOOSM	Multiple-Objective Optimisation based Sensor Management
MPC	Model Predictive Control
NCV	Nearly Constant Velocity
NNRWBS	Nearest-Neighbour Repeated Weighted Boosting Search
OAF	Output Augmented Fusion
OLFC	Open-Loop Feedback-Control
OWF	Optimal Weighted Fusion
PCRLB	Posterior Crámer-Rao Lower Bound
PDF	Probability Density Function
PF	Particle Filter
POLFC	Partial Open-Loop Feedback-Control
POMDP	Partially-Observed Markov Decision Process
QOI	Quality of Information
Radar	Radio Detection and Ranging
ROI	Region of Interest
RMSE	Root Mean Squared Error
RWBS	Repeated Weighted Boosting Search
SAR	Synthetic Aperture Radar
SIS	Sequential Importance Sampling
SLAM	Simultaneous Localisation and Mapping
SM	Sensor Management
SMC	Sequential Monte-Carlo
SMOF	Sensor Management Objective Function
SRKF	Square-Root Kalman Filter
SVAF	State Vector Assimilation Fusion
TRL	Technology Readiness Level
TTF	Track-to-Track Fusion
UAV	Uninhabited Aerial Vehicle
UGV	Uninhabited Ground Vehicle
UKF	Unscented Kalman Filter

Nomenclature

General Notation

$[(\cdot)]$	Vector whose elements are $(\cdot)$
$(\cdot)_k$	$(\cdot)$ at discrete time k
$a\|b$	Variable a conditioned on b
$(\cdot)^T$	Matrix transpose
$(\cdot)^{-1}$	Matrix inverse
$\|(\cdot)\|$	Matrix determinant
$\{(\cdot)\}$	Set or interval whose elements are $(\cdot)$
$E_x\{(\cdot)\}$	Expected value of $(\cdot)$ with respect to x
$\forall(\cdot)$	For all $(\cdot)$
$a \triangleq b$	a is defined as b
$p(\cdot)$	Probability distribution of $(\cdot)$
$\hat{(\cdot)}$	Estimate of $(\cdot)$
σ	Standard deviation
σ^2	Variance
$a \in b$	a is a member of the set b
$\|(\cdot)\|_a$	Absolute value of $(\cdot)$
$\|\|(\cdot)\|\|$	Euclidean norm of $(\cdot)$
$\Delta(\cdot)$	Change in $(\cdot)$
δ	Kronecker delta operator
$\Re$	Set of real numbers
$\approx$	Approximately equal to
$a \propto B$	Variable a varies proportionally to b
I^m	Identity matrix of dimension m
iff	If, and only if
$\mathcal{N}(a, b)$	Normal distribution with mean a and variance b

Sensor and Target Notation

dt	Sampling time interval
X^t	Target state vector

$X^{s,i}$	State vector for sensor i
x^t	x-position of target in Cartesian coordinates
y^t	y-position of target in Cartesian coordinates
$x^{s,i}$	x-position of sensor i in Cartesian coordinates
$y^{s,i}$	y-position of sensor i in Cartesian coordinates
$\dot{x}^{s,i}$	Velocity of sensor i in x-dimension
$\dot{y}^{s,i}$	Velocity of sensor i in y-dimension
$u^{s,i}$	Control input vector for sensor i
ψ^i	Heading for sensor i
z^i	Observation vector associated with sensor i
Z	Set of observation vectors
r^i	Range of sensor i to target
θ^i	Bearing of sensor i to target
v^t	Target process noise vector
$v^{s,i}$	Process noise vector for sensor i
w^i	Observation noise vector associated with z^i
f^t	Target state evolution function
$f^{s,i}$	Sensor state evolution function for sensor i
h^i	Observation function associated with sensor i
A	Linear state transition matrix
H	Linear observation matrix
Q	Process noise covariance matrix
R^i	Observation noise covariance matrix for sensor i
$\mathcal{X}$	Set of all possible state vectors
$\mathcal{Z}$	Set of all possible observation vectors

Sensor Control Notation

$\mathcal{J}$	Fisher Information Matrix
$\mathcal{H}(a)$	Entropy of a
$\mathcal{I}(a,b)$	Mutual Information of a and b
Ω	Optimisation constraint set
a	Sensor control action vector
a^i	Control action vector for sensor i
$a^\star$	Optimal sensor control action vector
$\mathcal{A}$	Set of all possible sensor actions
$\mathcal{U}(a,X)$	Utility associated with taking action a in state X
β	Requirement for trace of estimator covariance matrix

Global Optimisation Notation

U	Feasible set of candidate solutions
$\mathbf{u}$	Population of candidate solutions
J	Cost function
δ_i	Weight for population member i

Acknowledgements

Firstly I would like to offer my sincere thanks to my supervisors, Emeritus Professor Chris J. Harris and Professor Neil M. White. Not only have they offered me excellent guidance and support throughout my studies, they have also been friendly, approachable, and a pleasure to study with.

I am thankful to Dr. Alexander Dolia for help and advice and for always being on hand for useful (and often entertaining) discussion. A thank you is also offered to my internal examiner, Professor Mark S. Nixon. Thanks also to fellow project members Dr. Dimosthenis Karatzas (especially for the programming tips), Dr. Peter Boltryk and Dr. Arsenia Chorti. I gratefully acknowledge the UK MOD Data and Information Fusion Defence Technology Centre, for funding.

During my studies I was lucky enough to be housed within the ESD group of ECS, a great collection of characters who have made the last few years very enjoyable indeed. Special note goes to Cheryl Metcalf and Daryl Cotton, with whom I spent most of my time in the office, and who shared the ups and downs of PhD life. Thanks also to Dr. Mathew Swabey (the walking Wikipedia), as well as Andrew Frood, Geoff Merrett, Dr. Andy Cranny, Dr. Neil Grabham, Dr. John Tudor, Dr. Steve Beeby and many more. A sincere thank you is also extended to Lesley Courtney for general help and support with all things related to admin and Shaun Ford and ECS Stores for their help with ordering equipment. I would like to praise all the members of ESD and everyone else in the ECS department for their brave approach to dealing with the Mountbatten fire in 2005. The months following the fire were disorientating for many people, and would have been ever more so if it were not for the support of others.

Special note goes to the great people who were residents of the house on Highfield Lane between 2003 and 2007 - Prav, Dan, Rob, Jon, and Macka. A huge thank you is due to my family for unwavering encouragement and financial support during my undergraduate and postgraduate studies. Thanks also goes to Christine, Peter, and Ray for high quality proof-reading. Finally, I would like to thank my wonderful girlfriend, Olivia, who has supported me through the good times and the bad, and I would not be writing this acknowledgement without her. I very much look forward to spending more time with her.

To Olivia

Chapter 1

Introduction

Over the last few decades, military sensor resources have become increasingly cheap and now offer higher levels of performance than ever before in terms of signal quality, resolution, flexibility, and data rates. It is now, therefore, common to adopt multiple-sensor systems for military and civilian surveillance and reconnaissance activities, such as detecting, tracking, and identifying objects of interest (targets). Often such systems involve multiple sensor platforms such as aircraft, ground-sensors, and in recent times, Autonomous Vehicles (AVs). In order to utilise these resources in an efficient manner, much research has been carried out in the field of Data-Fusion (DF). Many different definitions of Data-Fusion can be found in the literature [1], but the common underlying concept is of integrating data from multiple, disparate sources, in order to derive higher levels of performance in terms of accuracy and robustness than could be achieved with the individual sources.

The performance of any multiple-sensor fusion system is inherently constrained by the data which it collects; a choice must therefore be made on how to optimise the configuration of the sensor resources, prior to Data-Fusion, in order to ensure that the data obtained leads to optimal system performance. Traditionally, this process is either conducted *a priori* in the design stage, for example when an engineer plans the physical placement of Radars in multi-static systems, or *online* by a human operator, for example a solider controlling an Uninhabited Aerial Vehicle (UAV). The capability of modern sensor systems now means that the complexities involved in efficiently optimising multiple-platform, multiple-sensor configuration is far beyond the limits of human capability. This fact, coupled with the trend towards fully autonomous systems, which is motivated by the desire to reduce the chance of human casualty, has driven research into the field of autonomous *Sensor Management* (SM). Autonomous SM considers methods by which sensors and sensor platforms can automatically self-configure in order to collect and process the most optimal data. A pertinent example is a team of autonomous UAVs collaborating to search for a target lost at sea [2]. The UAVs must cooperatively select their trajectories such that the chances of finding the target as quickly as possible are

maximised. This is just one example of how SM can be used to exploit sensor resources, and the underlying theory has broad application in both military and civilian domains.

This thesis presents a number of contributions to the SM field, relating to the manner in which such control problems are formulated and solved.

1.1 Overview of Sensor Management

The SM processes considered in this work are *online* problems (also known as active or dynamic SM), in that the configuration of the sensor resources is controlled over time. This is in contrast to the offline SM process, examples of which include the Radar placement problem described in the previous section, and other problems such as camera placement. This class of problem, therefore, implies the use of sensor resources that can be re-configured, for instance into different modes, or physically moved, for example Pan-Tilt-Zoom cameras or AVs. The generic online SM process aims to actively reconfigure the sensor suite to achieve improved overall system performance. The following is offered as an explicit definition of SM for the purposes of this research:

Definition 1.1. Active Sensor Management

> *'On-line, adaptive management and configuration of constrained sensor resources to optimise the result of various perceptive and non-perceptive tasks in accordance with multiple, possibly conflicting, performance measures'*

The typical SM system may be responsible for a number of activities associated with various perceptive tasks such as:

- deciding which known targets to observe;
- deciding which areas to search in for new targets;
- selecting which sensors to use for the observation tasks;
- spatial, frequency and time domain sensor and platform control;
- and, controlling new resource deployment.

The ultimate *output* of the SM process is therefore a control time-line for the sensor resources that are available to the system. A number of issues complicate this process, for example:

- multiple, possibly conflicting objectives;

- constraints on sensor resources;
- random sensor faults and failures;
- varied sensor capability and performance;
- and, the presence of uncertainty.

The list and definition presented above both point out the important fact that real sensor systems are constrained in various ways. Example constraints include limited physical motion, communication bandwidth and computational resources. A real system is also likely to have significant objectives other than those associated with observation tasks, such as maintaining operation, or fulfilling other non-perceptive goals (e.g. reaching way-points). These issues are key factors in one of the major contributions of this thesis, which considers how such constraints can be incorporated into the control process.

The presence of uncertainty also poses a serious challenge for the SM process. The usual military observation tasks of detecting, tracking, and classifying targets are usually solved through the use of probabilistic estimation methods, which naturally provide a way to handle the uncertainty. The SM process must also be capable of handling this uncertainty, and compensating for it in some way. This is the focus of the second area of contribution in this thesis.

Sensor Management problems commonly resolve into complex optimisation problems due to the uncertainty involved in the underlying observation tasks. Such problems are typically solved using stochastic search techniques such as Genetic Algorithms (GAs) and the SM community therefore relies on advances in the optimisation fields. This is the focus of the third area of contribution in this thesis.

1.1.1 Origins and Applications

Sensor Management research can be traced back to the late 1960s and early 1970s following the development of the Kalman Filter (KF), which provided the foundations for modern target tracking theory. Initial SM research was focused on controlling Radar systems to optimise the detection and tracking of targets. Sensor Management has enjoyed a higher profile since the 1990s, primarily due to improvements in computational resources, which have permitted the development of more sophisticated probabilistic estimation techniques, but also due to the increased availability of high performance sensor technology.

The majority of SM research programmes to date have been military funded, and have been approached from a number of contexts, from target tracking in tactical aircraft and AVs [1],[3], to submarine tracking [4]. An example civilian context is computer vision

[5]. Good introductions to the military fusion-based SM problem can be found in [6] and [7].

The term 'Sensor Management' itself has reportedly been around for over a decade and is beginning to come under some scrutiny; the authors of [8] suggest that 'Perception Management' is perhaps a less restrictive label for such research as it does not focus purely on sensors, and therefore could accommodate other sources of information such as human observation (this is particularly relevant as many military systems have humans in the loop). In comparison, other authors have described SM as an under-researched area within the larger science of DF itself [3]. 'Resource Management' is another typical term that can often be found in the literature. For the moment, however, Sensor Management is an established term in the literature, especially in the defence community.

It is not always clear where the separation of DF and SM lies in application. The authors of [9] describe SM as the 'process refinement' of a DF system; 'partially inside and partially outside the Data-Fusion process'. Sensor Management is also often portrayed as the 'feedback-control' of sensor resources [6]. An illustration of this concept is shown in Figure 1.1, which shows the flow of information in the system model assumed in this research. The SM process can be divided conceptually into performance measurement, optimisation, and decision stages. It should be noted that this division does not necessarily imply any algorithmic structure, but is purely a tool for conceptualising the process.

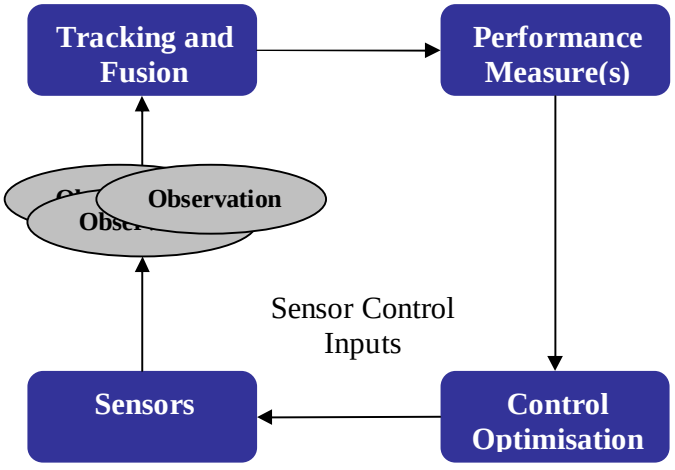

FIGURE 1.1: Sensor Management feedback-control paradigm.

This information flow architecture is also reflected in the well known JDL (USA Joint Directors of Laboratories) fusion model [3], [9]. Following tradition in military research, the model is split by levels of functionality [3], and in this case SM is analogous to JDL level four, otherwise known as process refinement. The 1999 revised JDL model is illustrated in Figure 1.2 [10] for convenience.

The JDL model also highlights the tendency to use a multiple-level architecture [11]. Further models, some of which enforce a top-down decision making structure within SM, are widespread [6], [8], [12]. Additionally, it is important to note that in application, some feedback is likely to occur between all levels of the JDL system architecture.

Optimising sensor configuration implies that the performance of the fusion process must be assessed against some kind of metric or performance measure. Unfortunately there is no generic measure; commonly metrics such as tracking accuracy are used to assess the performance of a sensor manager but mechanisms for evaluating it against other objectives are also beginning to appear [13]. This reflects the evolving maturity of the DF and SM fields. The SM problem can be described as analogous to a multiple-objective optimisation. As stated in [3], the ideal measure of performance would 'transcend the diversity of sensor tasks, be analytically tractable, robust in the face of operational constraints and account for the temporal evolution of the operational environment'. It should be highlighted that selected performance measures (and therefore associated optimisation criteria) will be highly mission and scenario dependent. However, it is clear that there is some scope for generic performance measures which are suitable for many scenarios, particularly in the military domain where tasks such as target tracking and classification are commonplace and share common theoretical underpinnings. As a result of the wide scope of SM research, it is difficult to compile a concise review of the literature without a reference basis, and the context of fusion performance measures provides a convenient platform from which to review the state of the art.

The SM field draws on influences from a wide range of research areas. Aside from the obvious links with estimation, DF, and target tracking, the research detailed in this work is closely related to areas of control theory, decision theory, and optimisation, among many others. As a result there are likely to be many important results in the literature associated with these fields which are highly relevant to SM research. Sensor Management can therefore be considered a branch of applied decision or control theory,

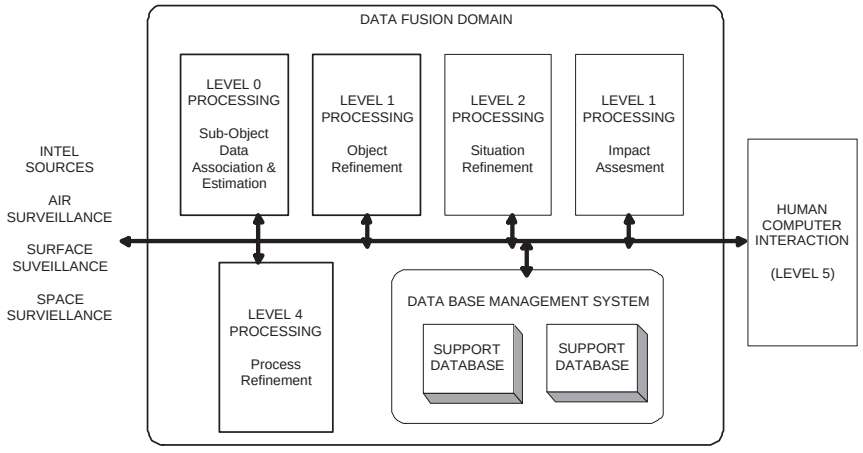

FIGURE 1.2: Revised JDL model (1999).

and a large number of research programmes involve significant aspects of cross-discipline technology integration.

1.1.2 Current Trends in Sensor Management

As mentioned in the previous section, a large proportion of international programmes is now focused on the application of SM to autonomous military reconnaissance and surveillance tasks involving the use of AVs. Until recently, much of this work was at a low Technology Readiness Level (TRL), i.e. it was based purely on simulation and theoretical analysis. Research is now aiming at pushing SM systems towards real-world demonstrators and test-beds. For example, a team from the Australian Centre for Field Robotics (ACFR) recently demonstrated, for the first time, successful autonomous and cooperative control of multiple UAVs for localising the position of a number of ground features [14][1].

FIGURE 1.3: DARPA Grand Challenge finalists.

Similarly, in 2005, an autonomous Uninhabited Ground Vehicle (UGV) nicknamed 'Stanley', built by Stanford University, successfully completed the Defense Advanced Research Projects Agency (DARPA) Grand Challenge, whereby the vehicle autonomously traversed a 132 mile off-road course with no human intervention[2]. Three of the finalists are depicted in Figure 1.3. In November 2007, following this programme, a further competition was held named the DARPA Urban Challenge, where the vehicles where tasked with navigating a 60 mile urban course which included obeying California State traffic laws. A total of six teams finished the course in under six hours. These programmes demonstrate state of the art AV technology and represent autonomy that is not yet reflected by in-service military platforms. A number of UAVs with some level of autonomy are in service in the West, for example the Northrop Grumman GlobalHawk (Figure 1.4) and the General Atomics Predator UAV.

[1] The ACFR pioneered Sensor Management research in the 1990s.

[2] The course was completed in a time of just under seven hours and a total of five vehicles finished.

FIGURE 1.4: GlobalHawk Uninhabited Aerial Vehicle.

Commercial and government security programmes are also developing autonomous UAV and UGV technology for surveillance and security purposes (e.g. for monitoring large events such as the Olympics and protecting other critical infrastructure). An example UGV platform is illustrated in Figure 1.5.

Many of such systems still require significant manual involvement in the command and control process and thus the level of autonomy remains relatively low. For example, the Predator UAV must be directed to different locations manually, and operates mainly as a data *collection* platform rather than an autonomous surveillance agent (see Figure 1.6). Ultimately, the aim is to achieve full autonomy so that platforms can manage their own behaviour in an intelligent fashion, thus removing entirely the need for human involvement. This is motivated by the desire to reduce the risk of human casualty and

FIGURE 1.5: Autonomous ground-based surveillance vehicle.

to allow operation in very harsh and hostile environments. It is also motivated by the potential performance that could be achieved through the use of autonomy.

FIGURE 1.6: Predator UAV control room.

Future UAVs and UGVs will therefore require the capability to autonomously control their own trajectories, the configuration of on-board sensors and to cooperate within teams of other platforms in order to achieve certain goals. There are numerous complex theoretical problems associated with developing such technology and a vast array of different sub-systems that must be integrated to achieve this aim. Whilst many aspects of these sub-systems are beginning to become well established (at least in theory and simulation), there are a number of open problems which face the SM community. These include the continued development of high performance planning strategies which can be implemented in decentralised networks, the integration of practical aspects of mission management with planning strategies, and methods to solve the increasingly complex optimisation problems that arise.

1.2 Thesis Structure

The research in this thesis is focussed on the development of *planning* algorithms which are aimed at optimising the information gathering process associated with target tracking[3]. These algorithms are required in order to optimise the trajectory of sensor platforms, and the configuration of on-board sensors. Broadly, this thesis examines three areas relating to such algorithms:

- how to balance information gathering with other important objectives;

[3]Aspects of identification and classification are beyond the scope of this work but can be considered using theoretical frameworks similar to those presented herein.

- how to exploit predictive planning algorithms to improve performance;
- and, how to solve the resulting optimisation problems.

The thesis is structured as follows. Firstly, background material regarding multiple-sensor target tracking systems and DF systems is presented in Chapter 2. This material provides the necessary technical foundations for the optimal control theory developed in later chapters. Chapter 3 presents background material relating to performance measures and sensor control. This includes a review of the SM literature based on an exposition of the associated performance measures or performance metrics.

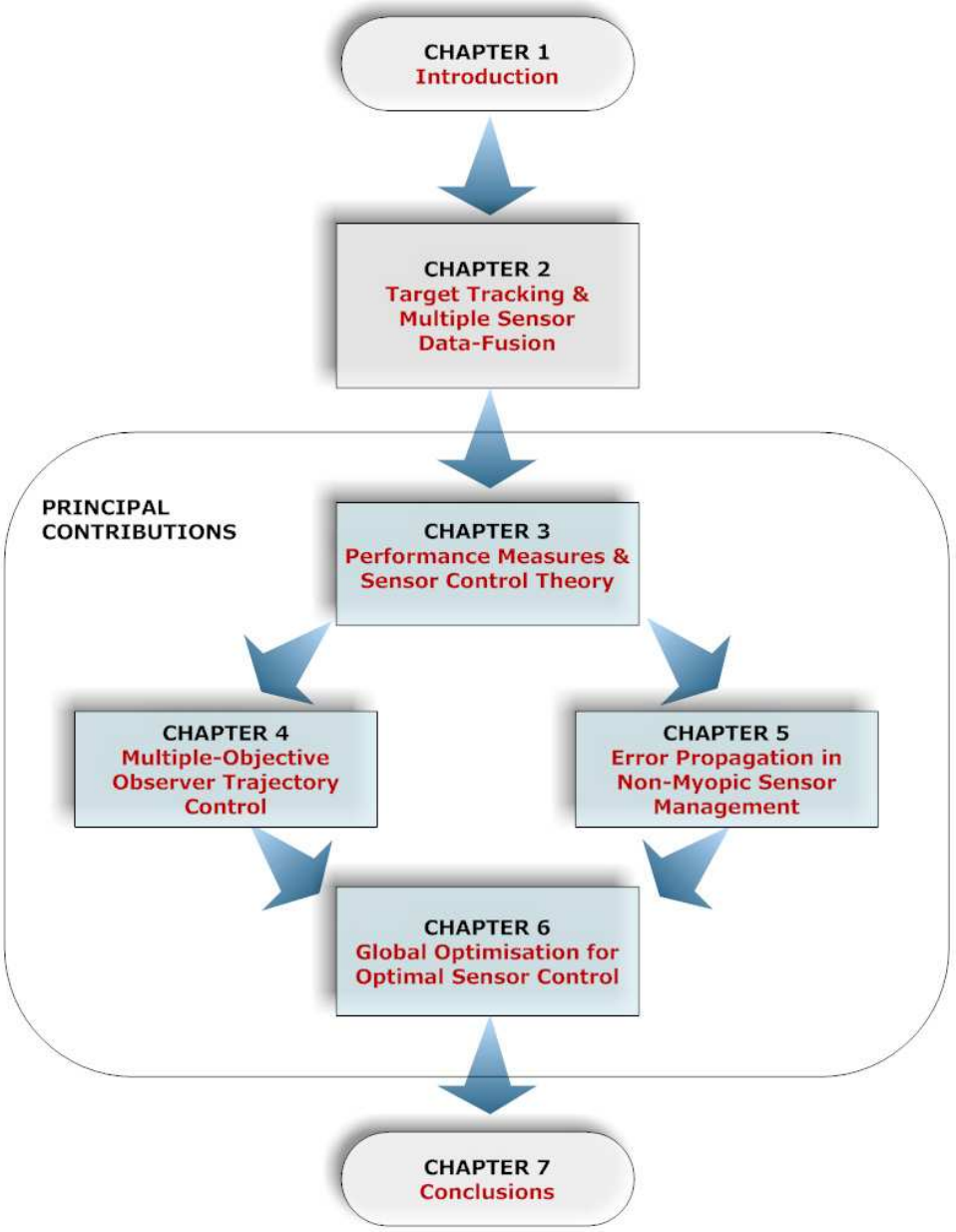

FIGURE 1.7: Illustration of thesis structure.

The primary contributions of this thesis are contained in Chapters 4-6. Chapter 4 presents a pragmatic analysis of the ground-based observer trajectory control problem from a multiple-objective perspective. Chapter 5 presents an analysis of the efficiency of multiple-step planning strategies. In Chapter 6, a number of novel stochastic search optimisation algorithms based on Repeated Weighted Boosting Search (RWBS) are presented.

Conclusions and proposals for future work are offered in Chapter 7. The thesis structure is highlighted in Figure 1.7.

1.3 Thesis Contributions

The primary contributions presented in this thesis are summarised below with reference to the associated chapter:

Chapter 4: Multiple-Objective Observer Trajectory Control

This chapter considers the UGV trajectory optimisation problem from a pragmatic viewpoint. In addition to the information gathering tasks associated with target tracking, other important objectives relating to platform survivability need to be accounted for. This problem is formulated as a multiple-objective optimisation problem and different objective functions and preference structures are analysed. The primary contributions of this research are:

- an analysis of the relationship between the changes in sensor-target geometry induced by information gathering objectives and secondary objectives relating to sensor survivability;
- an analysis of the use of different multiple-objective optimisation formulations for UGV observer trajectory optimisation;
- and, a novel UGV observer control algorithm based on the minimax approach which permits the optimisation of secondary system objectives whilst simultaneously maintaining a predefined tracking performance level.

Chapter 5: Adaptive Predictive Sensor Management

This chapter considers the non-myopic sensor control approach which exploits predictions of future target states to further optimise platform trajectories. Due to the computational requirements associated with such algorithms, they are typically solved using approximate methods such as receding-horizon control or other approximating techniques to yield tractable algorithms. Chapter 5 explores the efficiency of such approaches by examining the propagation of error through the control feedback-loop. It is shown

that there are two fundamental competing mechanisms which need to be balanced to achieve maximum planning performance and a strategy for identifying this balance is presented. This results in a novel control methodology based on an adaptive horizon control scheme. The primary outputs of this research are:

- an analysis of the propagation of uncertainty in a predictive SM feedback-loop;
- the identification of a multitude of error sources which can contribute to the propagation of error, and thus planning performance;
- and, an approach to developing a novel adaptive horizon control strategy which improves on the performance of classical limited-lookahead control by improving robustness to uncertainty.

Chapter 6: Global Optimisation for Sensor Management

The planning problems described above typically yield complex optimisation problems which cannot be solved using traditional derivative-based methods. Such problems are, therefore, typically solved using a stochastic search approach such as a GA. This chapter develops a number of extensions to the RWBS algorithm, which was recently proposed as an alternative to the GA. RWBS offers simple implementation and tuning procedures thus making it a potential candidate for autonomous application in evolving domains. The extensions presented in this thesis facilitate the algorithm's application to a wider class of problems, including those with discrete or mixed search spaces, and to Pareto-optimisation. The resulting algorithms maintain the attractive features of the original algorithm, namely ease of implementation and simplicity of tuning, while providing excellent convergence performance on standard test problems. The primary outputs of this research are:

- a quantitative convergence comparison of Repeated Weighted Boosting Search, multiple-start Quasi-Newton, random search, and a Genetic Algorithm;
- an analysis of the tuning parameter sensitivity of Repeated Weighted Boosting Search;
- a Repeated Weighted Boosting Search Optimisation algorithm that is capable of optimising over mixed search spaces;
- and, a Pareto multiple-objective optimisation algorithm based on Repeated Weighted Boosting Search.

1.4 Related Publications

Aspects of this research have been published in the open literature. The associated publications are listed below:

- S. F. Page, A. N. Dolia, C. J. Harris, N. M. White 'Multiple Objective Optimization for Active Sensor Management', Proceedings of the SPIE: Multisensor Multisource Information Fusion: Algorithms and Applications, Vol 5813, SPIE Defense and Security Symposium, Orlando, Florida, USA, March-April, 2005.
- S. F. Page, A. N. Dolia, C. J. Harris, N. M. White, 'Adaptive Horizon Model Predictive Control based Sensor Management for Multi-Target Tracking', Proceedings of the American Control Conference (ACC), Minneapolis, Minnesota, USA, June 2006.

The following publications contain, in part, contributions by the author of this thesis, but are not directly related to the research presented herein.

- A. N. Dolia, L. Mihaylova, S. F. Page, N. M. White, C. J. Harris, D. Bull, 'Multiple Objective D-optimal Sensor Management for Group Target Tracking', Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Orlando, Florida, USA, May, 2006.
- A. N. Dolia, S. F. Page, N. M. White, C. J. Harris, 'Passive Low SNR Tracking by Spatial-Temporal Fusion of Sliding-Window Radon Transforms', Proceedings of the Eighth International Conference on Information Fusion, Philadelphia, USA, July 25-29, 2005.
- A. N. Dolia, S. F. Page, N. M. White, C. J. Harris, 'D-Optimality for Minimum Volume Ellipsoid with Outliers', Proceedings of the Seventh International Conference on Signal/Image Processing and Pattern Recognition, (UkrOBRAZ), pp.73-76, Kiev, Ukraine, October 11-15, 2004.
- A. N. Dolia, S. F. Page, N. M. White, C. J. Harris, 'μ-MCD Approach to Novelty Detection', Machine Learning, Support Vector Machines, and Large Scale Optimization Workshop, Wissenschaftszentrum SchloB, Thurnau, Germany, March 16-18, 2005.

Chapter 2

Multiple-Sensor Data-Fusion for Target Tracking

2.1 Introduction

This chapter presents background material relating to multiple-sensor DF and state estimation methods, which form the basis of the target tracking process. In later chapters, the target tracking processes discussed here become the systems which are subject to control. It is necessary, therefore, to model and understand the operation of these processes, in order to identify optimal methods with which to control them. In the following discussion, particular focus is drawn to the *performance* of different state estimation methods, which is a key factor in the development presented in Chapters 4 and 5. The material in this chapter corresponds to the state of the art associated with the sub-systems highlighted in red in Figure 2.1

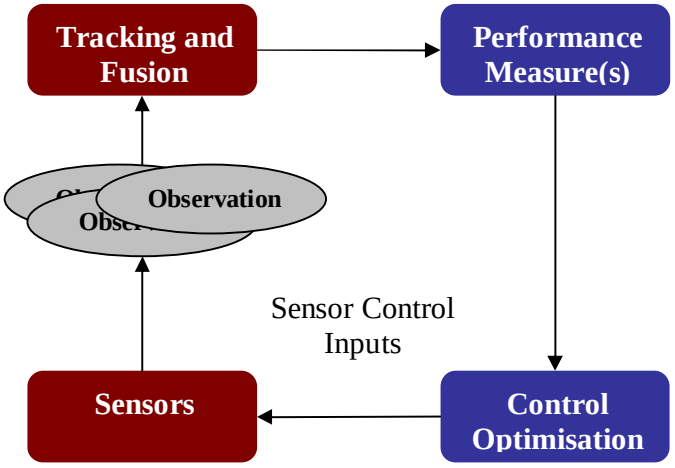

FIGURE 2.1: Data-Fusion and target tracking aspects of Sensor Management architecture.

It is noted that there are various other relevant processes which are generally pre-requisites for target tracking, for instance target detection[1], and data-association. These processes are not the subject of investigation in this thesis. Issues regarding the impact of the performance of such processes are discussed in Chapter 7.

This chapter begins with details of assumed sensor and target models, followed by the standard state estimation methods and multiple-sensor DF models relevant to target tracking. As discussed in Chapter 1, probabilistic methods of representing uncertainty in tracking methods are prolific in the DF and related fields, and this is reflected by the framework adopted in this thesis.

2.2 Modelling Sensors and Targets

Sensor and target models are of paramount importance in tracking and sensor control systems. The models provide the foundation for inference and subsequently the basis for identifying and quantifying the benefit of different sensor actions or control inputs. There is a plethora of different observation and dynamic models relating to various types of sensors and targets. In this work, focus is drawn to the use of simple sensor and target dynamics and high-level observational models. The use of such well-understood models reduces unnecessary complications in analysing processes which operate at a higher level. In addition, the use of generic observation models serves to maintain the flexibility and applicability of any derived control algorithms.

2.2.1 Sensor and Target Dynamics

In the following section, no differentiation is made between sensor and platform models (the terms are used interchangeably). It is assumed that, without loss of generality, each sensor has an associated platform on which it is mounted, and, therefore, any references to dynamic sensor models are equivalent to references to sensor platform models.

The generic state-space sensor dynamic model is defined formally as:

$$X_{k+1}^{s,i} = f^{s,i}\left(X_k^{s,i}, u_k^{s,i}, v_k^{s,i}\right) \tag{2.1}$$

where $X_k^{s,i}$ is the i-th sensor state vector at time step k, $f^{s,i}$ is the associated known (possibly non-linear) sensor state evolution function, $u_k^{s,i}$ is the sensor control input and $v_k^{s,i}$ is the sensor state noise vector. The superscript 's' is used to denote that the state vector refers to a sensor, and the superscript 'i' denotes the sensor index[2]. For the

[1] Typically targets are detected first, and then they are tracked. Recent 'track-before-detect' methods are the exception here.

[2] The reader is reminded that the sensor may be one of many.

purposes of this research, it is assumed that sensor dynamics are fully deterministic, and thus $v_k^{s,i} = 0, \forall k$. The impact of this assumption is discussed in Chapter 7.

The simplest sensor model considered in this thesis is a static sensor. This model represents a fixed sensor such as a ground-based Radar, Electro-Optical (EO) imaging device or an acoustic detector. The state of such a sensor, $X^{s,i}$, is defined here simply by its two–dimensional position, $x^{s,i}, y^{s,i}$, in a global Cartesian coordinate frame. The implicit assumption is that the sensor is not subject to any movement (or indeed vibration) and, therefore, there is no requirement for a time index:

$$X^{s,i} = \begin{bmatrix} x^{s,i} \\ y^{s,i} \end{bmatrix}. \tag{2.2}$$

Two dynamic sensor models are considered in this work. The first is an abstract model designed to represent the interaction of a platform with a sensor control algorithm. In this case, it is assumed that a SM algorithm requests a new sensor position, and some external process (e.g. an autonomous path planning and navigation system) actuates the request and moves the sensor. Under these assumptions, there is no need to model the velocity and position of the sensor, only the two–dimensional position. The dynamic model for such a sensor is outlined below (note the re-introduction of the time subscript, '$_k$'):

$$X_{k+1}^{s,i} = X_k^{s,i} + u_k^{s,i}, \tag{2.3}$$

where $u_k^{s,i}$ is the control input vector comprised of sensor movements (i.e. the change in two-dimensional position):

$$u_k^{s,i} = \begin{bmatrix} \Delta x^{s,i} \\ \Delta y^{s,i} \end{bmatrix}. \tag{2.4}$$

This model is generic and could be used to represent the motion of a range of different platforms[3]. The second model is more appropriate for airborne platforms, and is based on the assumption that the platform is moving at nearly constant velocity (NCV). This is a commonly-used model in the tracking and fusion fields. In this case, the sensors are modelled using their positions and velocities in two–dimensions and their heading with respect to some reference bearing. A further three variables are now introduced into the state vector:

$$X_k^{s,i} = \begin{bmatrix} x_k^{s,i} \\ y_k^{s,i} \\ \dot{x}_k^{s,i} \\ \dot{y}_k^{s,i} \\ \psi_k^i \end{bmatrix}, \tag{2.5}$$

[3]It is noted that the applicability of such a model is easier to justify for ground and naval platforms which do not necessarily require constant velocities to maintain motion, unlike some airborne platforms.

where $\dot{x}_k^{s,i}$, $\dot{y}_k^{s,i}$, and $\psi_k^{s,i}$ are the velocities and platform heading respectively. The sensor evolution equation is linear and is defined as:

$$X_{k+1}^{s,i} = A^s X_k^{s,i} + u_k^{s,i}, \tag{2.6}$$

where the constant state evolution matrix, A^s is given by:

$$A^s = \begin{bmatrix} dt & 0 & dt & 0 & 0 \\ 0 & dt & 0 & dt & 0 \\ 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.7}$$

where dt is the sampling interval, and $u_k^{s,i}$ represents the i-th sensor control input which is comprised of the new sensor heading, ψ_{k+1}^i:

$$u_k^{s,i} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \psi_{k+1}^i \end{bmatrix}. \tag{2.8}$$

In this model, the only control input is the platform heading, which may be subject to constraints relating to the sensor's maximum turn rate:

$$\psi_{k+1}^i - \psi_k^i \leq \psi_{max}, \tag{2.9}$$

Additional constraints on the platform's minimum and maximum velocities may also be imposed depending on the assumed platform type.

The exposition above has defined the manner in which sensor states and dynamics are modelled. Targets are modelled in an similar fashion. Static targets or 'features' may correspond to stationary targets or other points of interest such as buildings and geographical features. The state of such a target is defined again by its two-dimensional Cartesian position and does not require a time-script[4]:

$$X^t = \begin{bmatrix} x^t \\ y^t \end{bmatrix}. \tag{2.10}$$

Dynamic targets are modelled using the NCV approach outlined above for sensor dynamics, with the exception that in this case, the dynamics are not deterministic. The

[4]The reader is directed to the use of the superscript 't' to distinguish between target and sensor states, which are denoted using the superscript 's'.

associated target evolution equation is given by[5]:

$$X_{k+1}^t = A^t X_k^t + v_k^t, \tag{2.11}$$

where A^t is as defined in equation (2.7) and the target process noise vector, v_k^t, is defined as:

$$v_k^t = q \begin{bmatrix} \frac{dt^4}{3} & \frac{dt^2}{2} & 0 & 0 \\ \frac{dt^2}{2} & dt & 0 & 0 \\ 0 & 0 & \frac{dt^4}{3} & \frac{dt^2}{2} \\ 0 & 0 & \frac{dt^2}{2} & dt \end{bmatrix} \tag{2.12}$$

where q is the target process noise power spectral density. Various other dynamic models can be found in the DF literature, including constant-acceleration, constant-turn-rate, and various models for representing the motion of manoeuvering targets. Most of the models are designed to approximate the motion of military targets. In addition, a number of models have been developed for accounting for the motion of drifting bodies, for example objects lost at sea [2].

2.2.2 Sensor Observation Models

Sensor observation models describe the relationship between sensor observations and sensor and target states. Real targets are extended objects and thus high fidelity observation models are very complex. A common simplifying assumption is to model targets as point objects. While real targets are not point objects, this proves very useful by simplifying the observation model and also the resulting estimation process. There is also some practical justification for this assumption based on the fact that, as the observation distance increases, targets effectively reduce in size and thus approach point targets.

One of the most commonly used observation models is the 'range-bearing' model. This model can represent several real-world sensors including Synthetic Aperture Radar (SAR), Laser range finders and EO triangulation systems (see Figure 2.2). The following observation model is defined in this case:

$$z_k^i = \begin{bmatrix} r_k^i \\ \theta_k^i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k^{s,i} - x_k^t)^2 + (y_k^{s,i} - y_k^t)^2} \\ \tan^{-1}\left\{\frac{y_k^{s,i} - y_k^t}{x_k^{s,i} - x_k^t}\right\} - \psi_k^i \end{bmatrix} + w_k^i,$$

where z_k^i is the observation vector for sensor i, r_k^i is the range from sensor i to the target and θ_k^i is the bearing between sensor i and the target. The observation noise vector, w_k^i, is assumed to have a known variance-covariance matrix:

$$E\{w_k^i {w_l^i}^T\} = R^i \delta_{k,l} \quad \forall k, l, \tag{2.13}$$

[5]Note that it is assumed that there is no target input vector.

(a) SICK Laser range finder.

(b) Electro-Optical sensor pod.

FIGURE 2.2: Example range-bearing and bearings-only sensors.

where δ is the Kronecker delta operator, R^i is the observation noise variance-covariance matrix:

$$R^i = \begin{bmatrix} {\sigma_r^i}^2 & 0 \\ 0 & {\sigma_\theta^i}^2 \end{bmatrix}, \tag{2.14}$$

and, ${\sigma_r^i}^2$ and ${\sigma_\theta^i}^2$ are the range and bearing noise variances respectively. An illustration of the sensor-to-target geometry associated with a range-bearing sensor is provided in Figure 2.3. The diagram illustrates the two bearings that are important in this scenario.

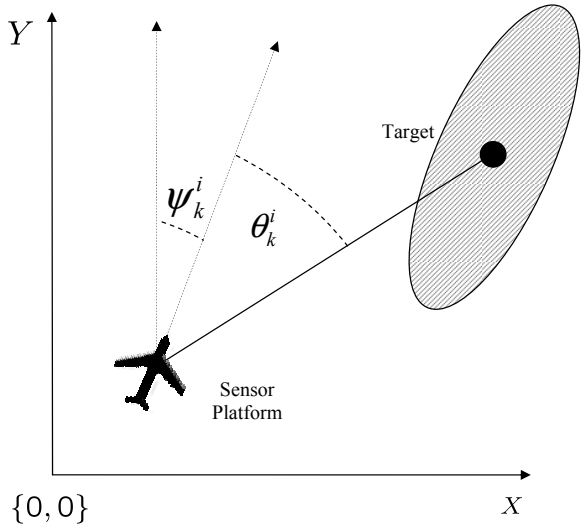

FIGURE 2.3: Range-bearing observation geometry.

The platform bearing, ψ_k^i, is the sensor/platform heading, and θ_k^i is the sensor-target bearing. In this work it is assumed that the sensors have a 360-degree Field-of-View (FOV). Bearings-only sensor models are often used to represent sensors such as single

EO imagers and acoustic sensors. This observation model is defined as:

$$z_k^i = \theta_k^i = \tan^{-1}\left\{\frac{y_k^{s,i} - y_k^t}{x_k^{s,i} - x_k^t}\right\} - \psi_k^i, \tag{2.15}$$

where the observation noise covariance is derived using:

$$E\{w_k^i {w_l^i}^T\} = R^i \delta_{k,l} \quad \forall k, l, \tag{2.16}$$

where

$$R^i = {\sigma_\theta^i}^2. \tag{2.17}$$

Range-only sensor observation models are defined in a similar fashion:

$$z_k^i = r_k^i = \sqrt{(x_k^{s,i} - x_k^t)^2 + (y_k^{s,i} - y_k^t)^2}, \tag{2.18}$$

with noise covariance matrix given in a similar way by:

$$R^i = {\sigma_r^i}^2. \tag{2.19}$$

2.3 Target Tracking Methods

Typically, information regarding the target state or process variables of interest is only available indirectly through noisy observations. The observations and *a priori* knowledge of the dynamics of the state, observation and noise processes can be used to infer estimates which are more accurate than the sensor observations themselves. This process is known as state estimation and is at the heart of target tracking techniques. The following describes the standard state estimation techniques, formulated within a Bayesian estimation framework.

2.3.1 Bayesian State Estimation

The target dynamic models outlined in Section 2.2 can be generalised to the following discrete-time stochastic target process model:

$$X_{k+1}^t = f_k^t\left(X_k^t, u_k^t, v_k^t\right), \tag{2.20}$$

where f_k^t is a known[6] function, X_k^t is the m-dimensional target state vector, u_k^t is the external control input and v_k^t is the process noise vector which accounts for both disturbances and process modelling errors. Measurements of the target state are obtained

[6]Note that there is a wide range of techniques that allow the process and measurement models to be generated using only data, see for instance [15].

through the generic observation process:

$$z_k^i = h_k^i \left(X_k^t, w_k^i \right), \tag{2.21}$$

where z_k^i is the n-dimensional measurement vector, h_k^i is a known function, and w_k^i is the observation noise vector. It is assumed that the functions f_k^t and h_k^i are constant, and the time index notation will be dropped from this point forward.

The state estimation problem can be considered from a Bayesian viewpoint as the recursive computation of the *belief* in the target state X_k^t at time k given a set of observation data vectors, $Z_{h:k}$, which can consist of the data received from multiple sensors[7]. For the purposes of this work, the problem is restricted to filtering (the case where the state estimate is conditioned on data up to and including the current time instant). The estimation problem in this context is equivalent to the computation of the posterior *Probability Density Function* (PDF), $p\left(X_k^t | Z_{h:k}\right)$. It is assumed that it is possible to construct the prior, initial state PDF, $p\left(X_{h-1}^t\right)$ which may derive from another estimation processes or other forms of *a priori* information. The recursive Bayesian filtering problem is then formulated as a two stage prediction-correction process as follows. Given a previous PDF $p\left(X_{k-1}^t | Z_{h:k-1}\right)$, which in the first time instant is the aforementioned initial PDF, the prediction state density is computed using the Chapman-Kolmogorov equation:

$$\underbrace{p(X_k^t | Z_{h:k-1})}_{\text{Prediction}} = \int \underbrace{p\left(X_k^t | X_{k-1}^t, Z_{h:k-1}\right)}_{\text{Transition}} \underbrace{p\left(X_{k-1}^t | Z_{h:k-1}\right)}_{\text{Prior}} dX_{k-1}^t, \tag{2.22}$$

where $p\left(X_k^t | X_{k-1}^t, Z_{h:k-1}\right)$ is the process evolution model represented in probabilistic form. If the system is first-order Markovian, the future state X_k^t is only dependent on the previous state and, therefore:

$$p\left(X_k^t | X_{k-1}^t, Z_{h:k-1}\right) = p\left(X_k^t | X_{k-1}^t\right). \tag{2.23}$$

The update or estimate stage now occurs as a new set of measurements, Z_k becomes available. Bayes' rule is used to update the prediction PDF using the new data:

$$\underbrace{p\left(X_k^t | Z_{h:k}\right)}_{\text{Posterior}} = \frac{\overbrace{p\left(Z_k | X_k^t\right)}^{\text{Likelihood}} \overbrace{p\left(X_k^t | Z_{h:k-1}\right)}^{\text{Prediction}}}{\underbrace{p\left(Z_k | Z_{h:k-1}\right)}_{\text{Normalising Constant}}}, \tag{2.24}$$

where

$$\underbrace{p\left(Z_k | Z_{h:k-1}\right)}_{\text{Normalising Constant}} = \int \underbrace{p\left(Z_k | X_k^t\right)}_{\text{Likelihood}} \underbrace{p\left(X_k^t | Z_{h:k-1}\right)}_{\text{Prediction}} dX_k^t, \tag{2.25}$$

[7] It is noted that the analysis in this work does not make use of joint multiple-target distributions which are sometimes used to deal with multiple targets.

is a normalising constant which depends on the likelihood density, $p\left(Z_k|X_k^t\right)$, which is constructed using the measurement model and the measurement noise statistics in a similar way to the transition density. The resulting posterior density is propagated through the update and prediction stages in (2.22) and (2.24) in order to find the Bayes solution. Any state estimate can be computed from the posterior PDF, for example the Minimum Mean Squared Error (MMSE) estimate, or the Maximum *A Posteriori* (MAP) estimate [16]. In addition to the state estimate, the posterior PDF can be used to determine a measure of the uncertainty associated with the estimate, such as covariance.

A state estimation algorithm is optimal in the probabilistic sense if it computes the posterior PDF exactly and completely [16]. Unfortunately, in general this PDF is equivalent to an infinite-dimensional vector and therefore the solution can be intractable. Except under certain conditions, the posterior density cannot be computed analytically or stored using sufficient statistics of finite-dimension. In this case sub-optimal algorithms must be used.

2.3.2 State Estimation for Linear Systems

Under certain assumptions, the Bayesian filtering problem can be solved analytically for linear systems. The following section details the standard optimal linear filter, the KF, and its dual representation, the Information Filter (IF). These filters are used in the development presented in later chapters.

2.3.2.1 The Kalman Filter

The KF [17, 18, 19, 20] has become the most commonly used state estimation algorithm since its conception in 1960. For linear, dynamic systems, the KF is a predictor-corrector algorithm that gives a linear, unbiased, minimum error variance estimate of an unknown state which is subject to additive Gaussian disturbed measurements. The KF is the analytical solution to the Bayesian estimation problem outlined in the previous section in this case.

It assumes that both the process noise and measurement noise are temporally-uncorrelated white Gaussian processes with zero mean. In addition, the two noise processes are assumed to be uncorrelated. The KF is optimal for Gaussian processes (it assumes the posterior PDF is Gaussian in nature and can therefore be completely represented by a mean and covariance) and optimal in the class of linear filters for non-Gaussian processes.

Consider the following linear state-space discrete time process and observation models, which represent (2.20) and (2.21) in the probabilistic framework outlined above:

$$X_k^t = A^t X_{k-1}^t + B_k^t u_k^t + v_k^t, \tag{2.26}$$

$$z_k^i = H^i X_k^t + w_k^i, \tag{2.27}$$

where X_k^t is an m-dimensional state vector:

$$X_k^t = [X_k^t(1)...X_k^t(m)]^T, \tag{2.28}$$

u_k^t is a g-dimensional input control vector:

$$u_k^t = [u_k^t(1)...u_k^t(g)]^T, \tag{2.29}$$

and z_k^i is an n-dimensional observation vector:

$$z_k^i = [z_k^i(1)...z_k^i(n)]^T. \tag{2.30}$$

A^t is the state transition matrix, B_k^t represents the control dynamics, and H_k^i is the linear observation model for sensor i. v_k^t and w_k^i represent additive Gaussian state and observation noise respectively. The noise processes are assumed to be independent and, therefore, satisfy the following conditions:

$$E\{v_k^t\} = E\{w_k^i\} = 0, \tag{2.31}$$

$$E\{v_k^t {v_l^t}^T\} = \delta_{k,l} Q \quad \forall k, l, \tag{2.32}$$

$$E\{w_k^i {w_l^i}^T\} = \delta_{k,l} R^i \quad \forall k, l, \tag{2.33}$$

$$E\{v_i^t {w_j^i}^T\} = 0. \tag{2.34}$$

The recursive Bayesian prediction and update stages given by (2.22) and (2.24) are implemented by the KF equations:

Prediction

$$\hat{X}_{k|k-1}^t = A^t \hat{X}_{k-1|k-1}^t + B_k^t u_k^t, \tag{2.35}$$

$$P_{k|k-1} = A^t P_{k-1|k-1} {A_k^t}^T + Q. \tag{2.36}$$

Update

$$K_k = P_{k|k-1} {H^i}^T \underbrace{[H^i P_{k|t-1} {H^i}^T + R^i]}_{\text{Innovation Covariance}}^{-1}, \tag{2.37}$$

$$\hat{X}_{k|k}^t = \hat{X}_{k|k-1}^t + K_k \underbrace{[z_k^i - H^i \hat{X}_{k|k-1}^t]}_{\text{Innovation}}, \tag{2.38}$$

$$P_{k|k} = [I^m - K_k H^i] P_{k|k-1}, \tag{2.39}$$

where $\hat{X}^t_{k|k}$ is the target state vector estimate, $P_{k|k}$ is the state vector estimate covariance, K_k is the Kalman gain matrix, and I^m is the m-dimensional identity matrix. The term $[z^i_k - H^i\hat{X}^t_{k|k-1}]$ in (2.38) and $[H^iP_{k|k-1}H^{iT} + R^i]$ in (2.37) represent the innovation and innovation covariance respectively. It is noted that the KF is equally applicable to cases where the matrices A,H,R, and Q are time-varying. For the purposes of this exposition, time subscripts are omitted as these matrices are assumed to be constant. Application of the KF is relatively straightforward, but some care must be taken in the initialisation stages of the filter. Covariance values for the noise processes must be calculated empirically if they are unknown. In addition, some tuning of the models and noise covariances may be required to achieve optimum performance. This is, of course, a reflection on the knowledge of the system under examination and not on the performance of the filter. A particular advantage of the KF is that the filter equations and gain equations are decoupled, allowing the user the ability to calculate the optimum Kalman gain and prediction covariance *off-line*, assuming noise covariances are constant. If this is not the case then the filter must stabilise *on-line*. The most significant computational cost of the filter hinges on the inversion of the innovation covariance matrix, which also implies that the matrix is invertible.

Note that in the linear Gaussian case, the conditional covariance definition coincides with the unconditional covariance definition. In other words, the covariance matrix is independent of the actual observation data which is integrated into the filter. This can be shown by taking the expectation of the conditional covariance [19]:

$$P_{k|k-1} = E_{z^i_{k-1}}\left\{[X^t_k - \hat{X}^t_{k|k-1}][X^t_k - \hat{X}^t_{k|k-1}]^T|z^i_{k-1}\right\}, \tag{2.40}$$

with respect to all possible sequences of observations, z^i_{k-1}, leading to:

$$P_{k|k-1} = E\left\{[X^t_k - \hat{X}^t_{k|k-1}][X^t_k - \hat{X}^t_{k|k-1}]^T\right\}. \tag{2.41}$$

The KF provides a way to keep track of the conditional probability density of the state. The mean of this density depends on the actual data which is observed, whereas the variance is unconditional as shown above.

2.3.2.2 Kalman Filter Variants

In practice, real world systems rarely satisfy the conditions required for estimation with the optimal linear filter. For instance, the noise processes may be non-Gaussian (and/or non-white) and cross-correlated. For linear systems with unknown or non-stationary noise processes, adaptive forms of the KF can be used for estimation [21]. Many of these types of filter estimate the noise process parameters by examining the statistical properties of the innovation sequence. As the innovation sequence is known to exhibit certain properties when the filter is acting optimally, it provides a means to identify

when and how the KF is performing sub-optimally. Limited-memory KFs can also be used to avoid the performance degradation associated with modelling errors [22]. In the case of coloured noise, shaping filters can be used to reduce the problem to one that the KF can address [20]. A variety of other alterations can be made to the KF structure to deal with other practical considerations.

Standard KF formulations have a number of numerical stability issues that can compromise filter performance. One major drawback is the marginal stability of the solution of the associated Riccati equation [23]. This means that in some cases, small rounding errors can propagate through the filter and cause performance degradation. In 1963, the Square-Root KF (SRKF) filter was introduced, which aims to combat this problem. The square-root filter factorises the covariance matrix into Cholesky factors; observation update is performed on these factors rather than on the original matrix. This results in improved numerical stability, but increases the computational requirements of the filter. Similar decomposition-based improvements and square-root implementations of the Unscented KF (UKF) and Particle Filter (PF) have also been proposed. For an introduction to the SRKF, see [24].

2.3.2.3 The Information Filter

An alternative representation of the KF is the IF or 'inverse-covariance filter'. It is algebraically and functionally identical to the KF and involves an alternate state representation, the 'information-state' vector, which is defined as

$$\hat{\mathrm{y}}^t_{k|l} \triangleq P^{-1}_{k|l}\hat{X}^t_{k|l}, \tag{2.42}$$

where $P^{-1}_{k|l}$ is the 'information matrix' (or inverse covariance), $\hat{X}^t_{k|l}$ is the target state estimate, and k and l are time indices. The prediction and update equations for the information filter are given by:

Prediction

$$\hat{\mathrm{y}}^t_{k|k-1} = P^{-1}_{k|k-1}A^t P_{k-1|k-1}\hat{\mathrm{y}}^t_{k-1|k-1}, \tag{2.43}$$

$$P_{k|k-1} = A^t P_{k-1|k-1}A^{tT} + Q. \tag{2.44}$$

Update

$$\hat{\mathrm{y}}^t_{k|k} = \hat{\mathrm{y}}^t_{k|k-1} + H^{jT}R^{j\,-1}z^j_k, \tag{2.45}$$

$$P^{-1}_{k|k} = P^{-1}_{k|k-1} + H^{jT}R^{j\,-1}H^j. \tag{2.46}$$

The estimated state is found by transforming back into standard representation:

$$\hat{X}^t_{k|k} = P_{k|k}\hat{\mathrm{y}}^t_{k|k}. \tag{2.47}$$

A common alternative is to represent the update stage of the IF for N_s sensors as

$$\hat{\mathrm{y}}^t_{k|k} = \hat{\mathrm{y}}^t_{k|k-1} + \mathrm{i}_k, \tag{2.48}$$

$$P^{-1}_{k|k} = P^{-1}_{k|k-1} + \mathrm{I}_k, \tag{2.49}$$

where

$$\mathrm{i}_k = \sum_{j=1}^{N_s} {H^j}^T {R^j}^{-1} z^j_k, \tag{2.50}$$

and

$$\mathrm{I}_k = \sum_{j=1}^{N_s} {H^j}^T {R^j}^{-1} H^j, \tag{2.51}$$

which highlights the additive and associative nature of the filter, a property which is often exploited for multiple-sensor DF and SM (this will be examined further in Chapter 3). Note again that the information matrix is unconditional for the time-invariant case.

The estimate phase of the IF is simpler than that of the KF as calculation of the term ${H^j}^T {R^j}^{-1}$ in (2.45) is less complex than that of the Kalman gain matrix in (2.37), especially as the innovation covariance is often high-dimensional and non-diagonal. Additionally, the IF avoids the problems associated with initialisation of the standard KF as the initial information matrix can simply be set to near–zero value. This is particularly important when comparing non-linear versions of the KF and IF (see section 2.3.3), where poor initialisation can lead to filter divergence and poor tracking performance. In the linear Gaussian case, poor filter initialisation will only lead to poor estimation performance in early stages, and not to filter divergence.

2.3.3 State Estimation for Non-Linear Systems

In cases where the process or observation models are non-linear, an analytic solution to the filtering problem does not generally exist[8]. In such cases, approximate (i.e. sub-optimal) filters must be employed. This section details the common state estimators which are employed for non-linear target tracking: the Extended Kalman Filter (EKF), the UKF, and the PF.

2.3.3.1 The Extended Kalman Filter

The EKF [18, 19] is a well-researched algorithm for the estimation of non-linear systems. It utilises a gradient-based linearisation about the current state (using a first-order Taylor expansion) to approximate the non-linear system. Appropriate use of the EKF assumes that the non-linearities are both smooth and differentiable.

[8]There are some cases where analytic solutions can be found for restricted classes of non-linear systems, see for example [16].

The EKF is a predictor-corrector algorithm that gives a linear approximate conditional mean estimate of an unknown state subject to Gaussian disturbed measurements. Like the KF, it assumes that both the process noise and measurement noise are temporally uncorrelated white Gaussian processes with zero mean. In addition, the two noise processes are assumed to be uncorrelated. In practical applications, the noise processes are used to account for both process noise and modelling error. The EKF is actually a *linear* filter - it uses the standard KF equations based on the approximation of the system dynamics. As such, the EKF is sub-optimal in the sense that there could exist a non-linear filter that offers superior performance, and, indeed, this is often the case. It is, however, optimal in the class of linear filters.

As with the KF, care must be taken in the initialisation stages of the EKF. However, unlike in the case of the KF, poor initialisation of the EKF can lead to divergence and highly degraded filter performance.

The filter usually requires the calculation of the associated Jacobian matrices of the non-linear system at *every* step of the iterative filtering process; the filter must, therefore, be used *on-line* as the Jacobians are not generally guaranteed to be constant [19]. The EKF also requires the matrix inverse operation found in the KF.

The EKF equations are essentially identical to the KF equations:

Prediction

$$\hat{X}^t_{k|k-1} = f^t\left(X^t_{k-1|k-1}, u^t_k, 0\right) \tag{2.52}$$

$$P_{k|k-1} = A^t_k P_{k-1|k-1} \tilde{A}^{tT}_k + Q. \tag{2.53}$$

Update

$$K_k = P_{k|k-1} H_k^{iT} \underbrace{[H^i_k P_{k|t-1} H_k^{iT} + R^i]}_{\text{Innovation Covariance}}^{-1}, \tag{2.54}$$

$$\hat{X}^t_{k|k} = \hat{X}^t_{k|k-1} + K_k \underbrace{[z^i_k - H^i_k \hat{X}^t_{k|k-1}]}_{\text{Innovation}}, \tag{2.55}$$

$$P_{k|k} = [1 - K_k H^i_k] P_{k|k-1}, \tag{2.56}$$

with the exception that A^t_k and H^i_k are now given by first-order linearisations of the non-linear target evolution and observation functions:

$$A^t_k = \frac{\partial f^t}{\partial X}|_{\hat{X}^t_{k-1|k-1}, u^t_k}, \tag{2.57}$$

$$H^i_k = \frac{\partial h^i}{\partial X}|_{\hat{X}^t_{k-1|k-1}}. \tag{2.58}$$

Note that the observation matrix, H^i_k, is now dependent on the target that is being observed and is time-varying. The IF can also be extended to deal with non-linear systems in the same way, to form the Extended Information Filter (EIF) - see [1].

2.3.3.2 The Unscented Kalman Filter

It is widely accepted in the fusion and tracking community that the EKF is very difficult to tune and is only applicable where the non-linearity of the process or measurement model is small over the length of the update interval [25]. The UKF was proposed to offer improved estimation performance for non-linear systems [25], [26], [27]. Based on the Unscented Transform, a method for calculating the statistics of a random variable which undergoes a non-linear transformation, the UKF utilises a carefully chosen set of sample points (known as sigma points), that capture the posterior mean and covariance accurate to second-order (in relation to a Taylor series expansion) for *any* non-linearity.

An n-dimensional random variable can be approximated by a set of $2n+1$ weighted sigma points given by the following deterministic algorithm:

$$\begin{aligned} \mathrm{X}_{k|k-1}(0) &= \mathrm{X}_{k-1|k-1} & W(0) &= \lambda/(n+\lambda), \\ \mathrm{X}_{k|k-1}(1:n) &= \mathrm{X}_{k-1|k-1} + (\sqrt{(n+\lambda)P})(i) & W(i) &= 1/2(n+\lambda), \\ \mathrm{X}_{k|k-1}(n:2n) &= \mathrm{X}_{k-1|k-1} - (\sqrt{(n+\lambda)P})(i) & W(i+n) &= 1/2(n+\lambda), \end{aligned} \tag{2.59}$$

where $\lambda \in \Re$ and $(\sqrt{(n+\lambda)P})(i)$ is the i-th row or column of the matrix square root of $(n+\lambda)P$ and $W(i)$ is the weight which is associated with the i-th point.

The set of sigma points, X are transformed by propagating each point through the process model. The exact process model is used:

$$\mathrm{X}_{k|k-1}(i) = f^t\left(\mathrm{X}_{k-1|k-1}(i), u_k^t, 0\right). \tag{2.60}$$

The predicted mean is computed by

$$\hat{X}_{k|k-1}^t = \sum_{i=0}^{2n} W(i)\mathrm{X}_{k|k-1}(i), \tag{2.61}$$

and the predicted covariance is computed by

$$P_{k|k-1} = \sum_{i=0}^{2n} W(i)\left[\mathrm{X}_{k|k-1}(i) - \hat{X}_{k|k-1}^t\right]\left[\mathrm{X}_{k|k-1}(i) - \hat{X}_{k|k-1}^t\right]^T. \tag{2.62}$$

Each point is then propagated through the observation model (again the exact observation model is used):

$$\gamma_{k|k-1}(i) = h^i\left(\mathrm{X}_{k|k-1}(i)\right). \tag{2.63}$$

The predicted observation is given by:

$$\hat{z}_{k|k-1}^i = \sum_{i=0}^{2n} W(i)\gamma_{k|k-1}(i). \tag{2.64}$$

Under the assumption of independent additive noise the innovation covariance, P^{zz}, is computed by:

$$P^{zz}_{k|k-1} = \sum_{i=0}^{2n} W(i) \left[\gamma_{k|k-1}(i) - \hat{z}^i_{k|k-1}\right] \left[\gamma_{k|k-1}(i) - \hat{z}^i_{k|k-1}\right]^T. \tag{2.65}$$

The cross correlation matrix, P^{xz}, is determined by:

$$P^{xz}_{k|k-1} = \sum_{i=0}^{2n} W(i) \left[\mathrm{X}_{k|k-1}(i) - \hat{x}^t_{k|k-1}\right] \left[\gamma_{k|k-1}(i) - \hat{z}^i_{k|l-1}\right]^T. \tag{2.66}$$

The update stage is described as follows:

$$\kappa = P^{xz}_{k|k-1} {P^{zz}_{k|k-1}}^{-1}, \tag{2.67}$$

$$\hat{X}_{k|k} = \tilde{X}_{k|k-1} + \kappa(z^i_k - \hat{z}^i_{k|k-1}), \tag{2.68}$$

$$P_{k|k} = P_{k|k-1} - \kappa P^{zz}_{k|k-1} \kappa^T. \tag{2.69}$$

The method differs from traditional Monte-Carlo (MC) techniques (see section 2.3.3.3), which require orders of magnitude more samples to maintain the state distributions [28]. In addition, the samples used in the UKF are not drawn at random, unlike MC samples. For Gaussian processes the filter is accurate to third-order and accurate to at least second-order for non-Gaussian processes. Some tuning of the filter with respect to higher order terms is possible through appropriate choice of λ, and matrix square-root operation. The computational complexity is of the same order as the EKF [28], and a particular advantage is that no calculations of explicit Jacobians are required. However, the matrix inversion operation remains. The filter can also be applied to non-differentiable non-linear systems, and is not restricted to additive Gaussian noise assumptions.

Ultimately, the key difference between the EKF and the UKF is that the UKF is based on the notion that it is easier to approximate a Gaussian PDF than it is to approximate an arbitrary non-linear function. Both the EKF and the UKF approximate the posterior state PDF with a Gaussian distribution but differ in their approximation of the transformation of the PDF through the non-linear dynamics.

2.3.3.3 The Particle Filter

The PF [16, 29, 30, 31] also utilises sample-based approximations of the posterior state density. The PF method is also commonly known as condensation filtering, Sequential Monte-Carlo (SMC), interacting particle approximations, bootstrap filtering and survival of the fittest. The PF implements the recursive Bayesian process by MC simulation. There are many different incarnations of the PF, but the Sequential Importance

Sampling (SIS) algorithm described in this section forms the basis for most SMC-based methods [31].

The posterior state density is represented by a set of weighted random samples. In the limit, where the number of particles becomes very large, the sampled-based representation is equivalent to an exact representation of the posterior density and the particle filter becomes an exact Bayesian solution. The posterior density $p\left(X_k|Z_k\right)$ is approximated as

$$p\left(X_k|Z_k\right) \approx \sum_{i=1}^{N} W_k(i)\delta\left[X_k - \chi_k(i)\right], \tag{2.70}$$

where $\chi_k(i), i = 0, \ldots, N$ is a finite set of N support points with weights $W_k(i), i = 1, \ldots, N$. The weights are normalised to sum to one:

$$\sum_{i=1}^{N} W_k(i) = 1. \tag{2.71}$$

In the SIS algorithm, the sample weights are selected according to the Importance Sampling Principle. This principle allows samples to be evaluated, up to a proportionality constant, from a density π (which may be difficult to draw from) through the use of an *importance density* or *proposal density*, $\mathrm{q} \propto \pi$. For the recursive Bayesian filtering case it has been shown that the sample weights are updated according to:

$$W_k(i) \propto W_{k-1}(i)\frac{p\left(Z_k|\chi_k(i)\right)p\left(\chi_k(i)|\chi_{k-1}(i)\right)}{\mathrm{q}\left(\chi_k(i)|\chi_{k-1}(i), Z_k\right)}, \tag{2.72}$$

where $\mathrm{q}\left(\chi_k(i)|\chi_{k-1}(i), Z_k\right)$ is the selected importance density and the posterior density estimate is approximated as in equation 2.70. The sample-based estimate approaches the true density, $p(X_k|Z_k)$, as the number of particles becomes larger[9]. The SIS filter is the recursive propagation of weights and particles drawn from the importance density as new measurements arrive.

It is only in the last decade that the computational performance required to operate PFs with large numbers of particles (which are often required for satisfactory estimation performance) has become widely available and computational load remains the main disadvantage of the PF. Despite this, it can offer much improved estimation accuracy over the EKF and UKF for systems with large non-linearities. Recent work has been directed at developing the PF to reduce the associated computational requirements. One of the main issues centres around the computational cost of sampling in high-dimensional systems. Significant progress was made in the late 1990s including some important theoretical analysis [32]. For instance, research presented in [33] discusses the exploitation of the dynamic Bayesian structure to increase the efficiency of the filter.

[9]The PF is asymptotically *efficient* in the number of particles. Estimator efficiency is discussed in more depth in Section 2.4.5.

Such 'Rao–Blackwellised' filters have been shown to outperform the standard PF in a variety of applications.

2.3.4 Multiple-Target Tracking

The estimation techniques described in the previous section were presented in a form that is primarily applicable to single target tracking. Tracking multiple targets using these techniques can be achieved by using multiple filters, except insofar as it is difficult to associate observations with tracks, especially when in the presence of clutter. Many extensions of the above techniques have been proposed to combat this problem, for instance the Multiple Hypothesis Tracker (MHT), Joint Probabilistic Data-Association Filter (JPDAF), and many more. A detailed exposition of the standard multiple target tracking methods is not appropriate in this text, and the reader is referred to [34] for more information.

2.4 Properties of Estimation Methods

The sensor control research presented in later chapters of this thesis is based upon exploiting knowledge of the target state and the relationship between sensor state and filter performance in order to optimise sensor control inputs. The optimality of these control inputs will therefore depend on the performance of the state estimation process which is under control. This implicit feedback-loop is fundamental to the research presented in Chapter 5. In this section, measures of state estimation performance are defined.

2.4.1 Estimator Optimality

As described in Section 2.3.1, a state estimation algorithm is optimal in the *probabilistic* sense if it computes the posterior target state density exactly. This is expressed formally in the following way:

$$\underbrace{p(\hat{X}_k|Z_k)}_{\text{Estimated State Density}} = \underbrace{p(X_k)}_{\text{True State Density}} \tag{2.73}$$

In general, this level of optimality is only ever achieved if all the assumptions used in the estimation model, i.e. the observation and target process models, and associated noise statistic models, are true and *exact*, and if the filter does not approximate any part of the Bayesian state estimation formulation. In reality, there is almost certainly some level of approximation or model error in *every* system and thus no practical tracking system can be truly optimal. Additionally, practical target tracking scenarios entail non-linear observation and target models, and thus even if the assumed system model

were perfect, it is unlikely that an exact filter could be developed. However, in some special cases, the tracking process may be *near-optimal* depending on the magnitude of the approximations or modelling error and the magnitude of the non-linearity present in the system.

2.4.2 Estimator Bias

A key property in estimator design is bias. An estimator is unbiased if the expected value of the estimator is equal to the expected value of the true state [19]. This expectation involves integration over both the target and observation sets, $\mathcal{X}$ and $\mathcal{Z}$:

$$E_{\mathcal{X},\mathcal{Z}}\left\{X_k - \hat{X}_k\right\} = 0. \tag{2.74}$$

It should be noted that in some cases, it is possible to design to a biased estimator which performs better than the optimal unbiased estimator.

2.4.3 Online Estimator Accuracy

A common property of probabilistic state estimation algorithms is their maintenance of an estimate of the accuracy associated with the state estimate at each time step. In cases where the KF, EKF, or UKF is used, the posterior state PDF is modelled using a Gaussian distribution, and the accuracy estimate corresponds to the covariance matrix estimates in equations (2.39), (2.56) and (2.69) respectively. The accuracy of these estimates plays a key role in both the estimation process itself (due to the inherent feedback-loop in a sequential estimation system) and also in the performance of any SM algorithm which is built around the estimation process. Measures of the accuracy of the *assumed* or *apparent* filter performance are the subject of the following sections.

2.4.4 Estimator Consistency

It is easily assumed that the filter accuracy estimates maintained by typical tracking algorithms give true indications of the filter performance. Unfortunately, this is not always the case. In general, the estimated filter accuracy (e.g. the covariance matrix associated with a KF estimate) is only ever equivalent to the *true* filter accuracy (e.g. the actual Root Mean Squared Error (RMSE) error of the estimate computed with ground truth data) if the filter is optimal in the probabilistic sense, i.e. if equation (2.73) holds. In the general case, there will always be some discrepancy between the apparent filter error and the true filter error.

For brevity, from this point onwards it will be assumed that the state estimation technique in question maintains an estimate of the *covariance* associated with the measurement. This assumption is valid for the KF, EKF and UKF cases but is less appropriate in cases where a PF is used, as this type of filter does not rely on approximating the target state with a Gaussian distribution. In addition, for clarity of exposition, from this point onwards the estimated covariance will be known as the *apparent covariance*[10].

Two properties can be now defined in this context. Firstly, an estimator can be said to be *consistent* or *conservative* if the apparent covariance is greater than or equal to the true covariance:

$$\hat{P}_k - E_{\mathcal{X},\mathcal{Z}}\left\{(X_k - \hat{X}_k)(X_k - \hat{X}_k)^T\right\} \geq 0 \tag{2.75}$$

Note the introduction of the hat notation on the apparent covariance to highlight that it may be different from the true filter performance. Again it is noted that this expectation is taken with respect to both the target state and the observation set. The consistency property ensures that the filter does not underestimate its own error. This concept was central to the development of the UKF as it is known that the EKF can be inconsistent and that this can be linked to filter divergence [25].

This property is also key to the methodology presented in Chapter 5, where it will be shown that inconsistency in a filter can propagate through the planning process and lead to inconsistent evaluations of the benefit of different sensor actions.

2.4.5 Estimator Efficiency

Knowing that an estimator is consistent does not guarantee that the estimate is of any practical use. As Julier and Uhlmann pointed out in [25], the apparent covariance, $\hat{P}_k$, may be greatly in excess of the actual covariance and, ideally, the following quantity should be minimised:

$$\hat{P}_k - E_{\mathcal{X},\mathcal{Z}}\left\{(X_k - \hat{X}_k)(X_k - \hat{X}_k)^T\right\}, \tag{2.76}$$

that is to say, the filter should be *efficient*. This property is also key to the error propagation model derived in Chapter 5, where it will be used to show how an inefficient filtering process can yield an inefficient planning process.

2.4.6 Performance Bounds for Estimation

In practice, it is useful to be able to bound filter performance so that estimates provided by the filter can be fully exploited. Fortunately, it is possible to provide lower bounds on state estimation performance in many scenarios.

[10] This is to avoid repetitive and confusing use of the term 'estimated'.

2.4.6.1 The Posterior Crámer-Rao Lower Bound

The Posterior-Crámer-Rao-Lower-Bound (PCRLB) is an extension of the Crámer-Rao-Lower-Bound (CRLB) to sequential state estimation processes. The bound provides a lower-limit on filter performance; it specifies the *maximum* possible performance that could be achieved given an understanding of the estimation model.

Assume $\hat{X}(Z)$ is an unbiased estimator of a parameter vector X, and is a function of a measurement vector, Z. The CRLB for the covariance of the estimator is the inverse of the associated Fisher Information Matrix (FIM), $\mathcal{J}$:

$$P_{cr} \stackrel{\triangle}{=} E_{\mathcal{X},\mathcal{Z}} \left\{ (\hat{X}_k - X_k)(\hat{X}_k - X_k) \right\} \geq \mathcal{J}^{-1}. \tag{2.77}$$

For an unknown random vector, the individual terms in the bounded covariance matrix are given by:

$$\mathcal{J}(i,j) = E_{\mathcal{X},\mathcal{Z}} \left\{ -\frac{\partial^2 \ln p(Z,X)}{\partial X(i) \partial X(j)} \right\}, \tag{2.78}$$

where $p(Z,X)$ is the joint PDF of the state and measurements. The bound can be computed recursively for linear dynamics and Gaussian noise processes:

$$\mathcal{J}_{k+1} = (Q_k^t + A_k^t \mathcal{J}_k^{-1} {A_k^t}^T)^{-1} + \mathcal{J}_{k+1}^z, \tag{2.79}$$

where $\mathcal{J}^z$ is the measurement contribution. It is only recently that a method for computing the bound for the general non-linear filtering problem has been derived [35]. The bound has been extended to account for various other tracking problems including multiple target tracking, manoeuvering target tracking, and problems where there is a non-zero probability of detection. The PCRLB has also been used as the basis for a body of sensor control research, and this is discussed in more depth in Chapter 3.

2.5 Multiple-Sensor Data-Fusion

In the target tracking context, the purpose of multiple-sensor DF is to obtain more accurate state estimates than would be derived using individual sources. In many cases the sensors can be disparate and provide different types of information regarding the target and this data must be fused in a coherent manner in order to exploit the extra performance that multiple-sensor systems can offer.

There are many different fusion system structures in the target tracking literature, but they can be broadly classified according to the fusion *method*, and the fusion *architecture*. These concepts define what information is fused and where it is fused within the fusion system. This section details the common fusion methods found in multiple-sensor target tracking schemes and identifies the core fusion architectures that are associated with

them. This material is presented for two reasons: firstly, it motivates and contextualises the choice of fusion structure described in the development in Chapter 4; and, secondly, it will be shown that the fusion process can introduce additional approximations into the estimation process, and thus the performance of different fusion schemes is also linked to the development in Chapter 5.

2.5.1 Data-Fusion Methods

Data-Fusion methods can be categorised into measurement-fusion or track-fusion methods. Unsurprisingly, measurement-fusion methods combine measurements, or observation data, to generate state estimates. Track-fusion methods combine state estimates from separate sources to yield fused state estimates.

2.5.1.1 Measurement-Fusion

There are two principal measurement-fusion methods which differ according to how the observation data is combined. The first method combines data without using the statistical properties of the measurement processes. The second method utilises the covariance matrices of the noise processes to combine measurements in an optimal manner.

Output Augmented Fusion Output Augmented Fusion (OAF) merges measurements into an augmented observation vector, which emulates a single sensor that provides all the measurements of the total sensor suite [15]. As an illustrative example, consider observation data from two sensors in the form of two measurement vectors: z_k^1 and z_k^2. These vectors are merged into a new augmented measurement vector, Z_k^a:

$$Z_k^a = \left[{z_k^1}^T {z_k^2}^T \right]^T . \tag{2.80}$$

Individual observation matrices, C_k^1 and C_k^2, and individual noise vectors, w_k^1 and w_k^2, are also formed into augmented equivalents:

$$H_k^a = \left[{H_k^1}^T {H_k^2}^T \right]^T , \tag{2.81}$$

$$W_k^a = \left[{w_k^1}^T {w_k^2}^T \right]^T , \tag{2.82}$$

leading to the multiple-sensor observation equation constructed using augmented components:

$$Z_k^a = H_k^a \left[X_k^t + W_k^a \right] . \tag{2.83}$$

The covariance matrix for the augmented measurement noise is defined on the assumption of statistical independence between sensors and is constructed using the individual

sensor covariances: R_k^1, and, R_k^2:

$$R_k^a = \begin{pmatrix} R_k^1 & 0 \\ 0 & R_k^2 \end{pmatrix}.$$

Extension to sensor suites with more than two sensors is straightforward and is performed by augmenting all sensors in an identical fashion. The merged data can then be filtered using a standard state estimation technique, such as the KF [15]. OAF increases the dimensionality of the observation equation and therefore places larger computational load on the estimation process.

Optimal Weighted Fusion An alternative approach to augmenting multiple sensor measurements is to combine them in a mean square optimal manner and generate tracks using the resulting fused estimate. This procedure is known as Optimal Weighted Fusion (OWF). Assuming that the measurement noise for each sensor is statistically independent, the MMSE fusion equations for the linear case with N_s sensors are given by [15, 36]:

$$Z_k^w = \left[\sum_{i=1}^{N_s} {R_k^i}^{-1}\right]^{-1} \sum_{i=1}^{N_s} {R_k^i}^{-1} z_k^i, \tag{2.84}$$

$$H_k^w = \left[\sum_{i=1}^{N_s} {R_k^i}^{-1}\right]^{-1} \sum_{i=1}^{N_s} {R_k^i}^{-1} H_k^i, \tag{2.85}$$

$$R_k^w = \left[\sum_{i=1}^{N_s} {R_k^i}^{-1}\right]^{-1}, \tag{2.86}$$

where R_k^i, z_k^i,and H_k^i are the process noise covariance, observation vector and observation matrix at time k for sensor i respectively. The combined observations and measurement matrices are normalised weighted sums of observations and measurement matrices weighted by the inverse of the associated covariance matrices of measurement noise.

It is interesting to note that under the conditions that $H_k^1 = H_k^2$, and the measurement matrices are independent of the state noise covariance, Q_k^t, OAF is mathematically equivalent to OWF (for a proof, see [15]).

2.5.1.2 Track-Fusion

The optimal measurement-fusion process outlined in Section 2.5.1.1 utilises the statistical properties of the measurement noise processes in order to obtain more accurate fused data. It is logical therefore that optimal *track*-fusion methods utilise the statistical properties of the estimation error, for instance the KF estimate covariance given by equation (2.39), for fusion. Unfortunately, despite independence between measurement

noise processes in different sensors, state estimates of the same target can be dependent due to common process noise [37]. Additionally, estimates can be inconsistent, particularly when based on approximations of process and measurement models, and the state estimation covariance measures may not be realistic. These difficulties make track-fusion more problematic than measurement-fusion [15]. However, track-fusion methods can offer some advantages over measurement-fusion methods as will be described later in this section.

As with measurement-fusion there are two principal track-fusion methods for combining separate state estimates which differ according to how the state estimates are combined.

State-Vector Assimilation Fusion State Vector Assimilation Fusion (SVAF) is a sub-optimal fusion method based on assimilation of data sources, and does not make use of knowledge of the statistical properties of the state estimates. The definition of SVAF given below is based on an IF representation but this does not imply that SVAF is limited to the IF case [15]. Local information state and inverse covariance estimates at each node, i, are given by (as in 2.45 and 2.46):

$$\hat{\mathrm{y}}^i_{k|k} = \hat{\mathrm{y}}^i_{k|k-1} + {H^i_k}^T {R^i_k}^{-1} z^i_k, \tag{2.87}$$

$${P^i_{k|k}}^{-1} = {P^i_{k|k-1}}^{-1} + {H^i_k}^T {R^i_k}^{-1} H^i_k. \tag{2.88}$$

Assimilation fusion is then performed as follows:

$$\hat{\mathrm{y}}_{k|k} = \hat{\mathrm{y}}_{k|k-1} + \sum_{i=1}^{N_s} \left[\hat{\mathrm{y}}^i_{k|k} - \hat{\mathrm{y}}^i_{k|k-1} \right], \tag{2.89}$$

$${P_{k|k}}^{-1} = {P_{k|k-1}}^{-1} + \sum_{i=1}^{N_s} \left[{P^i_{k|k}}^{-1} - {P^i_{k|k-1}}^{-1} \right], \tag{2.90}$$

where $\hat{\mathrm{y}}^i_{k|k}$ and ${P^i_{k|k}}^{-1}$, are local estimates of the information state vector and information matrix, and $\hat{\mathrm{y}}_{k|k}$ and $P^{-1}_{k|k}$ are the fused information state vector and information matrix. SVAF is a simple method with low computational cost. In addition, it does not require estimate covariance information which reduces the communication requirements associated with its application.

Track-to-Track Fusion The Track-to-Track Fusion (TTF) method utilises estimate covariances to combine tracks [15, 36, 38, 39]. Consider the fusion of two KF-based state estimates originating from two independent sensor systems. The two separate

state estimates, $\hat{X}^{t,1}_{k|k}$ and $\hat{X}^{t,2}_{k|k}$, are fused to form a new estimate of the state vector[11]:

$$\begin{aligned}\hat{X}^{TTF}_{k|k} =& \hat{X}^{1}_{k|k} + [P^{1}_{k|k} - P^{12}_{k|k}][P^{1}_{k|k} + P^{2}_{k|k} - P^{12}_{k|k} - P^{21}_{k|k}]^{-1} \\ & \times \left(\hat{X}^{t,2}_{k|k} - \hat{X}^{t,1}_{k|k}\right), \end{aligned} \tag{2.91}$$

where $P^{i}_{k|k}$ is the covariance matrix for the state estimate $\hat{X}^{t,i}_{k|k}$, based on the measurements from sensor i, and $P^{ij}_{k|k}$ is the cross-covariance matrix between state estimates $\hat{X}^{t,i}_{k|k}$ and $\hat{X}^{t,j}_{k|k}$. State estimate cross-covariances are computed recursively by:

$$\begin{aligned} P^{ij}_{k|k} =& \left(I^{m} - K^{i}_{k} H^{i}_{k}\right) A_{k-1} P^{ij}_{k-1|k-1} A^{T}_{k-1} \left(I^{m} - K^{j}_{k} H^{j}_{k}\right)^{T} \\ & + \left(I^{m} - K^{i}_{k} H^{i}_{k}\right) Q^{t}_{k-1} \left(I^{m} - K^{j}_{k} H^{i}_{k}\right)^{T}, \end{aligned} \tag{2.92}$$

The TTF equation in (2.91) is the solution of the linear estimator, and the method is therefore sub-optimal in the general case. In addition, it has been shown that TTF is only ever optimal on the maximum likelihood sense [37], and that measurement-fusion outperforms TTF [40]. The cross-covariance is the critical issue that influences the accuracy of the fused state estimate and its computation is very complex. This adds to the difficulty of implementing TTF. As TTF requires the communication of track covariances, it is less suitable for decentralised implementation than SVAF (see section 2.5.2.2 for details on decentralised fusion architectures).

Track-fusion nodes generate *local* state estimates and receive local estimates from other nodes in order to construct the global estimate. Track-fusion, therefore, offers a level of data compression over measurement-fusion methods, especially in cases where the measurement data is high dimensional.

A modified form of track-fusion has been proposed in [41] which is reportedly more efficient in terms of computational cost, and is said to outperform measurement-fusion and conventional track-fusion for dissimilar sensors and high levels of process noise (which is often the case in target tracking).

2.5.2 Data-Fusion Architectures

A key principle in multiple-sensor DF is the selection of the fusion architecture. This principle defines the physical location of the fusion process and typical architectures can be broadly classified into centralised, decentralised, and hybrid categories. Each architecture has its own particular advantages and disadvantages, and the fusion system

[11] With a slight abuse of notation, $\hat{X}^{t,i}_{k|k}$ is used to refer to the state estimate of the target from the i-th node, rather than the state estimate of the i-th target.

designer must pick an appropriate architecture according to various criteria. These criteria will include fusion accuracy (the performance of fusion methods within the architecture), computational and communication requirements, system robustness, scalability, and modularity.

Measurement-fusion might typically be used in situations where each sensor is a low-cost device with limited computational power or within a spatially-centralised fusion platform. In many military tracking scenarios, however, there will be a number of separate sensor systems tracking a target. Each one of these sensor systems may produce its own individual state estimate or track, which drives research into track-fusion methods.

2.5.2.1 Centralised Fusion Architectures

In a centralised DF architecture, a single entity or fusion *node* is responsible for performing fusion on the data obtained from multiple sources. Data is typically passed from the various sources to the fusion node through a communication channel which may often only provide one-way communication. Fused data may be also be transmitted to external sources for further analysis or as part of a hybrid fusion architecture. Both measurement and track-fusion methods can be implemented within a centralised fusion architecture. A centralised measurement-fusion architecture is illustrated in Figure 2.4.

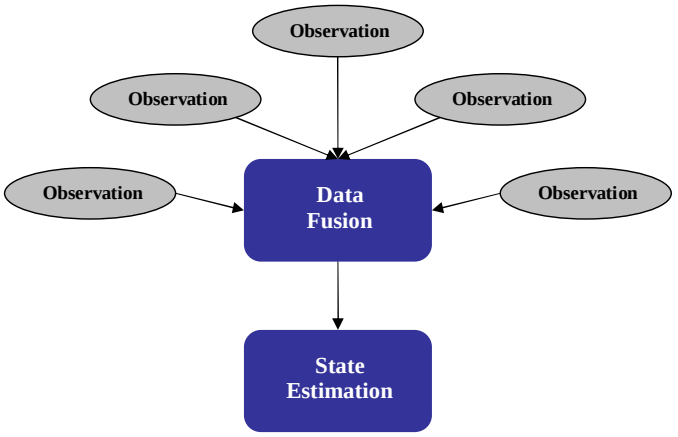

FIGURE 2.4: Centralised measurement-fusion architecture.

Due to the fact that only one working entity fuses data from multiple sensors, the algorithms employed in centralised fusion architectures are often similar in complexity to single sensor algorithms. In addition the central node can have access to all information from all sensors which is advantageous from the sensor management perspective.

A significant disadvantage of the centralised architecture is that it acts as a computational bottleneck. If new sensors are added, the fusion node must have access to sufficient computational resources to assimilate the new data, otherwise the node may become saturated. However, the most important disadvantage of the centralised paradigm is its

inherent lack of robustness against sensor failure; if the central fusion node fails, the fusion process stops and the entire sensor suite is potentially rendered useless. In safety critical or battlefield scenarios, such system failures may lead to unacceptable consequences. Decentralised fusion architectures are, therefore, of much greater concern in the current research climate.

2.5.2.2 Decentralised Fusion Architectures

Decentralised fusion architectures do not require a global fusion node. Instead each fusion node transmits and receives information to and from other fusion nodes[12]. As with centralised architectures, both measurement and track-fusion can be performed in a decentralised manner. Decentralised fusion systems are sometimes based on replication of the centralised fusion concept at each and *every* node. For example, the Decentralised Information Filter (DIF) consists of replication of the IF at every node. The result is that each node requires its own computational and memory resources in addition to communication requirements. Decentralised fusion nodes may, therefore, be of larger physical size and of more complex design. A decentralised track-fusion architecture is illustrated in Figure 2.5.

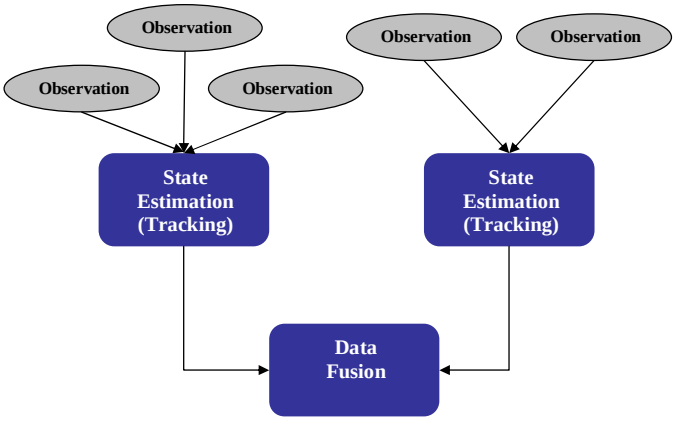

FIGURE 2.5: Decentralised track-fusion architecture.

The decentralised fusion architecture, whether it is based on measurement or track-fusion, is highly robust as there is no single critical node. If appropriate communication protocols are in place, sensors can be added as well as removed from the fusion process automatically, without the scalability and failure problems which arise with centralised systems. This robustness has important implications, especially in the military domain, as it improves the survivability and usefulness of the multiple-sensor fusion system. The efficiency of the decentralised fusion process depends on the communication mechanisms employed between fusion nodes [15].

[12]Note that decentralised fusion systems can be designed with both fully connected and partially connected communication networks. The reader is directed to [1] for examples.

2.5.2.3 Hybrid Fusion Architectures

Many practical fusion systems will consist of a hybrid of centralised and decentralised architectures. In this case, the overall performance and robustness of the system will depend closely on the detail of the fusion structure.

2.5.3 Performance of Fusion Systems

Evaluating the performance of a multiple-sensor fusion system requires an understanding of the performance of the individual state estimation components, and also the performance of the selected fusion method under the assumed fusion architecture. This is a non-trivial task in the case of decentralised or hybrid architectures due to the possible permutations of data that could be communicated between local nodes. However, in the centralised case this process is more simple. Analyses of the efficiency of several fusion methods has been examined in both centralised and decentralised cases [42].

2.6 Concluding Remarks

This chapter has presented the technical background material relating to state estimation and multiple-sensor fusion systems that form the basis for the research presented in Chapters 4 and 5. The primary area of focus in this chapter was analysing the *performance* of the different approaches to state estimation. This is a fundamental factor in any sensor control algorithm, as it is the performance of the state estimation process that becomes the system under control in a SM algorithm. The following chapter presents a review of performance measures and sensor control theory.

Chapter 3

Performance Metrics and Sensor Control Theory

3.1 Introduction

In the previous chapter, various algorithms and architectures for tracking targets of interest were explored, and a number of high-level concepts relating to the assessment of state estimation performance were discussed. This chapter reviews sensor control theory and the performance measures on which it is based. A broad remit is established which covers a range of different platform and control contexts. This chapter therefore corresponds to the processing architecture sub-systems highlighted in red in Figure 3.1.

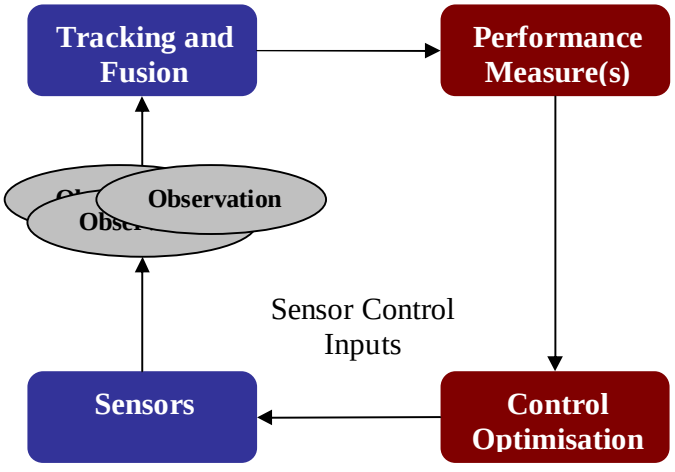

FIGURE 3.1: Performance measures and sensor control aspects of Sensor Management architecture.

The goal of SM is to identify the optimal sensor control input(s) or sensor action(s) over time. Optimality must of course be defined with respect to some specific measure or *metric*, which may often be heavily dependent on the problem and context. The

performance measure ultimately dictates many aspects of the nature of the control process and is thus a useful context within which to review the wide range of SM approaches that can be found in the open literature.

This chapter begins with a brief discussion of performance measure terminology followed by background regarding formal concepts of decision theory. It then presents a critical review of SM theory with specific attention drawn to the nature of the performance measure used. The limitations and weaknesses of the current approaches are analysed and used as basis for justification of the novel sensor control formulations described in Chapters 4 and 5.

While many of the approaches discussed in this chapter consider the sensor control problem from different theoretic perspectives, they all share common properties and can be related to general concepts of decision theory, planning, and optimisation.

3.2 Performance Measure Terminology

There is a variety of terms in the literature arising from different contexts that can cause some confusion, such as performance metric, measure of performance, measure of effectiveness, cost or objective function, and utility function. A further term which is beginning to arise in the SM literature is 'Quality-of-Information' (QOI) [43]. This expression is more commonly found in sensor network and algorithmic-agents research fields where it is used as a high-level term to express the overall quality of the information supplied by a network in relation to the key perceptive tasks (e.g. probability of event detection, time to event detection etc). As the links between the sensor network and SM research fields continue to grow, this term is likely to become increasingly common. At the time of writing, there is no generally accepted terminology however. Due to the confusion associated with the terms introduced above, three simple definitions are offered here for clarity:

1. **Fusion System Task (FST)**

   A perceptive or non-perceptive task. For example target tracking, classification, or sensor survival.

2. **Fusion System Performance Measure (FSPM)**

   A performance measure defined as a direct result of a particular FST. For example track estimation error or probability of detection.

3. **Sensor Management Objective Function (SMOF)**

   An objective or cost function that is intended to be used in the optimisation process within a Sensor Management system. This function

is likely to be designed or selected for efficient and high performance optimisation according to a specific FSPM.

In the previous chapter, a number of aspects of estimator performance were explored. However, evaluating target tracking is more complex than simply evaluating filter performance [44]. This is primarily due to the additional complexities that arise in real scenarios, such as non-zero probability of detection and non-zero probability of false alarm. These data-association factors complicate the evaluation of tracking performance as tracks have to be assessed according to wider criteria than purely tracking error.

There is a plethora of well-established performance measures in the target tracking literature[1], yet it is clear that many such metrics require ground truth data and, therefore, prove unsuitable for online SM. At this point, it is important to distinguish the difference between an 'online' performance measure which is used explicitly in the formulation of a control process and an 'offline' sensor control performance measure. An offline performance measure provides a mechanism with which to assess the performance of a given controller with respect to the ground-truth[2]. For example, an offline measure might include the true RMSE error of a tracking system, whereas an online measure might be the apparent filter covariance. While in some cases[3], the online performance measure may be equivalent to the offline measure (in special cases they may converge), in many cases the online measure may differ significantly from a ground-truth-based analysis.

Performance measures for fusion tasks other than detection and tracking are less common, although there has been some recent progress into formulating a set of measures inline with the JDL model [13]. In addition, it is now becoming clear that information-based measures are capable of being used as a universal proxy [45].

3.3 Background Theory

As described in Chapter 1, SM involves planning in the presence of uncertainty. If there were no uncertainty associated with the observation tasks then there would be no need for any state estimation processes, or any management algorithms to control them. In Chapter 2, a number of probabilistic state estimation and DF techniques were explored, as these typically form the basis for SM algorithms. It is usual, therefore, to formulate the SM process in a probabilistic framework.

This section presents brief background theory regarding relevant concepts of decision theory and utility analysis for scenarios with imperfect information.

[1] A comprehensive treatment of multiple-target tracking performance is inappropriate here and can be found elsewhere [44].

[2] The ground-truth may be available in test cases and in simulation.

[3] For instance the KF case.

3.3.1 Metric Properties

It is worth noting at this stage that a metric in the formal sense is a measure which satisfies a number of specific properties. It is a mapping, m, onto the real line which satisfies the following conditions [1]:

$$m(a,b) \geq 0 \qquad \text{(Non-Negativity)} \tag{3.1}$$

$$m(a,b) = 0 \quad \text{iff} \quad a = b \qquad \text{(Identity of Indiscernibles)} \tag{3.2}$$

$$m(a,b) = m(b,a) \qquad \text{(Symmetry)} \tag{3.3}$$

$$m(a,c) \leq m(a,b) + m(b,c) \qquad \text{(Triangle Inequality)} \tag{3.4}$$

where a and b are the metric's operands. It is noted that the triangle and identity of indiscernible conditions are sufficient for non-negativity. The imposed conditions are due to the notion of a metric being identified with the concept of distance between neighboring points of a given set.

3.3.2 Elements of Bayesian Decision Theory

One of the fundamental concepts underlying planning in the presence of uncertainty is that of expected value theory. By computing the expected value associated with making different decisions, it is possible to rank them accordingly. However, it was shown as early as the eighteenth century that expected value theory is normatively wrong[4]. In its purest form, the theory may yield irrational decisions due to its inability to account for practical aspects of the decision making process. Bernoulli demonstrated that the introduction of a utility function and the subsequent computation of expected utility can resolve such problems. It is not necessary to consider non-normative aspects of decision theory in the autonomous control context as it can be assumed that the control system will act rationally at all times.

Consider an unknown state, X, and an action $a \in \mathcal{A}$, which is a member of the set of all possible actions, $\mathcal{A}$. A utility function, $\mathcal{U}(X,a)$, or conversely, a loss function, $L(X,a)$, defines the utility (or loss) incurred, if action a is performed and the true state is X. A utility function is therefore a mapping between a state and an action to a number on the real line.

The relations that a utility function dictates must satisfy the following rationality axioms [1]:

- For any two actions $a_1, a_2 \in \mathcal{A}$, $\mathcal{U}(X,a_1) < \mathcal{U}(X,a_2)$ or $\mathcal{U}(X,a_1) = \mathcal{U}(X,a_2)$ or $\mathcal{U}(X,a_1) > \mathcal{U}(X,a_2)$ (Preference)

[4]The reader is referred to the famous St Petersburg Paradox.

- If $\mathcal{U}(X, a_1) < \mathcal{U}(X, a_2)$ and $\mathcal{U}(X, a_2) < \mathcal{U}(X, a_3)$ then $\mathcal{U}(X, a_1) < \mathcal{U}(X, a_3)$ (Transitivity)
- If $\mathcal{U}(X, a_1) < \mathcal{U}(X, a_2)$ then $\alpha\mathcal{U}(X, a_1) + (1 - \alpha)\mathcal{U}(X, a_3) < \alpha\mathcal{U}(X, a_2) + (1 - \alpha)\mathcal{U}(X, a_3)$ for any $0 < \alpha < 1$

In general, only probabilistic knowledge of the state X is available, and thus, if a number of different decisions can be taken, each leading to a number of different outcomes, then the natural course of action is to identify the action that, on average, will lead to the highest return. For this research it is important to make explicit the nature of expectation in this context. Given an action, a, the expected utility is given by:

$$E_{\mathcal{X},\mathcal{Z}}\left\{\mathcal{U}(X, a)\right\} \triangleq \int_{-\infty}^{+\infty} \mathcal{U}(X, a)p(X)dX. \tag{3.5}$$

The expectation is performed over the set of possible values of the state and observation. Selecting the action that maximises the value in (3.5) is equivalent to a *maxi-mean* hedging strategy. An intuitive interpretation is as follows: assume that the choice of action corresponding to the maximum expected value has been identified as a; if the decision process or experiment were repeated numerous times in exactly the same fashion, the action a would, in the limit as the number of experiments tends to infinity, lead to the highest average utility. The expected loss is defined similarly as:

$$E_{\mathcal{X},\mathcal{Z}}\left\{L(X, a)\right\} \triangleq \int_{-\infty}^{+\infty} L(X, a)p(X)dX. \tag{3.6}$$

The expected loss (3.6) is often known as the *risk*. The concept of expected utility, when combined with the Bayesian state estimation framework outlined in Chapter 2, yields the *Bayes Action*. In the SM context, one formulation of the Bayes Action, $a^\star$, is the action which maximises the expected utility according to the predicted state PDF conditioned on the action a, $p(X_k|a)$:

$$a^\star = \arg\max_a \int_{-\infty}^{+\infty} \mathcal{U}(X_{k+1}, a)p(X_{k+1}|a)dX_{k+1}dZ_{k+1} \tag{3.7}$$

This demonstrates how sensor control inputs or actions can be optimised according to different utility functions (e.g those relating to target tracking accuracy). Of particular note is the dependence of (3.7) on the predicted target state PDF which is the result of equation (2.22). This prediction often emerges naturally from the chosen underlying state estimation technique, for example it can be based on the KF prediction equations (2.35) and (2.39).

The utility theory presented above encodes the notion of risk, and thus the choice of utility function determines the behaviour of the resulting control system with regard to risk. The concept of *Risk-Aversion* relates to a utility function that prefers a lower-utility if it is accompanied by a higher probability of occurrence:

$$\mathcal{U}(E_{\mathcal{X},\mathcal{Z}}\{X\}, a) \geq E_{\mathcal{X}\mathcal{Z}}\{\mathcal{U}(X, a)\}. \tag{3.8}$$

Consider for instance a competition where the contestant must choose between a low-value prize with an almost certain probability of winning, or a high-value prize with a much lower probability of winning. The Risk-Averse strategy would select the low-value prize. Conversely, the *Risk-Prone* strategy (which is the inverse of the Risk-Averse strategy) would select the high-value prize. A *Risk-Neutral* strategy is indifferent to the two prizes. Measures of risk-aversion relate to the curvature of the utility function.

3.3.3 Other Notions of Risk

It is important at this point to introduce an entirely different notion of risk. While the utility and loss functions defined above capture uncertainty in a quantitative manner, there may well be other aspects of uncertainty or risk which are not modelled by the assumed utility function. This is due to inaccurate or approximate formulations in the model. In 1921, the economist Frank Knight established a distinction between these two concepts:

> '*The essential fact is that 'risk' means in some cases a quantity susceptible of measurement, while at other times it is something distinctly not of this character... It will appear that a measurable uncertainty, or 'risk' proper, as we shall use the term, is so far different from an unmeasurable one that it is not in effect an uncertainty at all.*' (Frank Knight, 1921).

The epistemological uncertainty described in the quote above is generally neglected in SM control research as it is usually assumed that the models which are used are sufficiently accurate that it can be neglected. However, as will be shown in Chapter 5, this is often not the case, and the propagation of such error may play a key role in system performance.

3.4 Performance Measures and Sensor Control Review

Now that the appropriate background theory has been established, the following sections present a review of the different performance measures and sensor control frameworks that can be found in the open literature.

3.4.1 Information-Based Methods

Measures of information are prolific in the SM field, and they are often used as the online performance measure for various SM tasks. Typical information measures include Entropy, Mutual Information (MI), and Fisher Information (FI). Much of the pioneering work in SM was carried out in the 1990s at the ACFR, and the majority of the work resulting from this period is almost exclusively based on information measures. A few of many examples can be found in [1], [2], [46], [47], [48], and [49].

Information measures have broad applicability across various different FSTs and FSPMs [45]. One of the simplest information-based measures is Entropy. Entropy (or Shannon information), is a measure of the uncertainty associated with a particular probability distribution. The entropy of a distribution, $p(X)$, is defined as:

$$\mathcal{H}(X) \triangleq -E_{\mathcal{X}}\left\{\log p(X)\right\} = -\int_{-\infty}^{+\infty} p(X)\log p(X)dX. \tag{3.9}$$

Entropic information is the complement of Entropy:

$$i(X) = -\mathcal{H}(X). \tag{3.10}$$

An important result is that the Entropic information associated with a Gaussian distribution is given by [1]:

$$i(X) = -\frac{1}{2}\log\left[(2\pi e)^n|P|\right]. \tag{3.11}$$

where n is the dimension of the distribution, and P is the covariance matrix. Figure 3.2 illustrates the Entropic information associated with Gaussian distributions of different variances. This measure defines how much is known about the random variable, X. Equation (3.11) is important in the SM context, because many state estimation algorithms represent the posterior state PDF as a Gaussian distribution, and it is the uncertainty in this distribution that is the subject of interest.

Consider now the case where the covariance matrix in equation (3.11) is replaced with the apparent covariance of a KF or IF. This yields a measure of the current uncertainty associated with the target state. It is easy to show that the result of assimilating a new measurement from a sensor with a known observation noise covariance matrix can be assessed according to the resulting change in Entropic information. This is the strategy employed in many SM algorithms including those found in [1], [2], [48] and related works.

The concept of Entropy can be extended to the conditional case:

$$\mathcal{H}(X|Z) \triangleq -E_{\mathcal{X}}\left\{\log p(X|Z)\right\} = -\int_{-\infty}^{+\infty} p(X|Z)\log p(X|Z)dX. \tag{3.12}$$

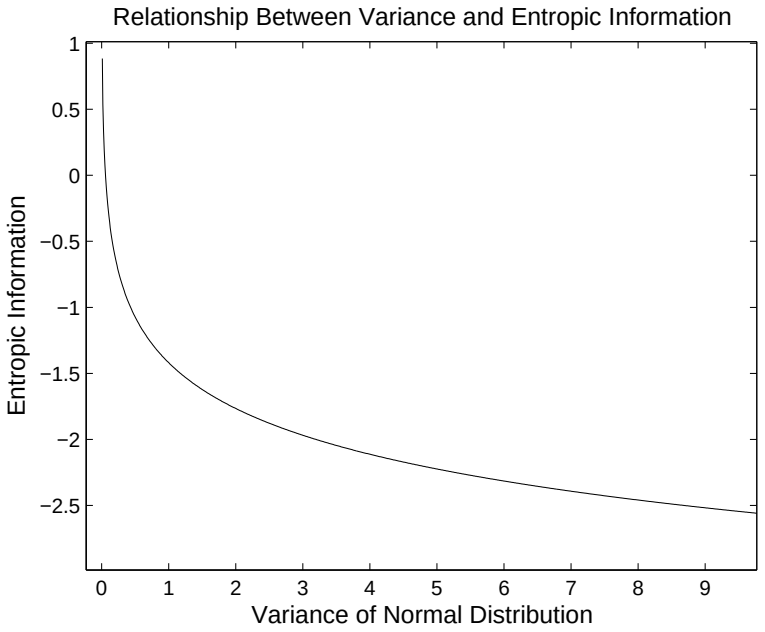

FIGURE 3.2: Relationship between variance and Entropic information.

Note that the conditional Entropy is a function of the observation vectors, Z. This formulation can be used to estimate the average Entropy associated with a measurement, through the *mean conditional Entropy* [48]:

$$\bar{\mathcal{H}}(X|Z) \triangleq E_{\mathcal{Z}}\{\mathcal{H}(X|Z)\} \tag{3.13}$$

$$\bar{\mathcal{H}}(X|Z) \triangleq -\int_{-\infty}^{+\infty} p(Z) \int_{-\infty}^{+\infty} p(X|Z) \log p(X|Z) dX dZ. \tag{3.14}$$

In this case the expectation is taken over both the state and the observation vectors:

$$\bar{\mathcal{H}}(X|Z) \triangleq -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} p(X,Z) \log p(X|Z) dX dZ, \tag{3.15}$$

and represents the average information to be gained by taking an observation. It is possible to define an information form of Bayes' law by taking the expectation of the logarithm of Bayes with respect to X and Z giving [48]:

$$\bar{\mathcal{H}}(X|Z) = \bar{\mathcal{H}}(Z|X) + \mathcal{H}(X) - \mathcal{H}(Z), \tag{3.16}$$

which permits a model of the information propagation in a state estimation system. The reader is directed to [48] for further material relating to information dynamics.

Mutual Information measures can also be used to assess the impact of observation data. Mutual Information has been used extensively by Grocholsky and Durrant-Whyte to

calculate the utility of a particular sensing action [47], [50], [51]. Mutual Information is defined as the information about one variable contained in another. For DF systems it is appropriate to define the information about a state, X, contained in an observation, Z, as MI:

$$\mathcal{I}(X,Z) = -E_{\mathcal{X},\mathcal{Z}}\left\{\log\frac{p(X,Z)}{p(X)p(Z)}\right\}, \tag{3.17}$$

$$\mathcal{I}(X,Z) = -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} p(X,Z)\log\frac{p(X,Z)}{p(X)p(Z)}dXdZ. \tag{3.18}$$

The relationship between observations with difference associated variances, and the resulting MI for Gaussian distributions with different original variances is depicted in Figure 3.3. Note the strong dependence of MI on how much is known about the random

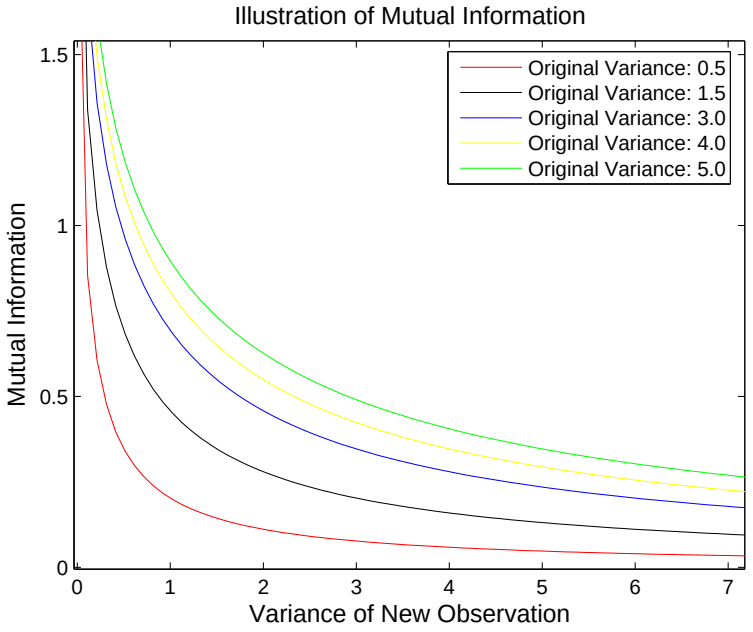

FIGURE 3.3: Mutual Information for Gaussian distributions with different variances.

variable before the observation is made. An alternative representation for the information about a state X contained in a set of observations is FI. Fisher Information is related to Entropy through the log-likelihood function, and it has been shown that minimising Entropy is equivalent to maximising FI [1]. It was shown in Chapter 2 that the FIM is defined as:

$$\mathcal{J}(i,j) = E_{X,Z}\left\{-\frac{\partial^2 \ln p(Z,X)}{\partial X(i)\partial X(j)}\right\}. \tag{3.19}$$

Figure 3.4 illustrates the relationship between FI and variance for a Gaussian distribution. Fisher Information is particularly useful when using an IF-based state estimation process. In such cases the FI contained in the posterior state estimate, $p(X_{k|k})$, and the prior state estimate, $p(X_{k|k-1})$ of the IF is given by $P_{k|k}^{-1}$ and $P_{k|k-1}^{-1}$ respectively.

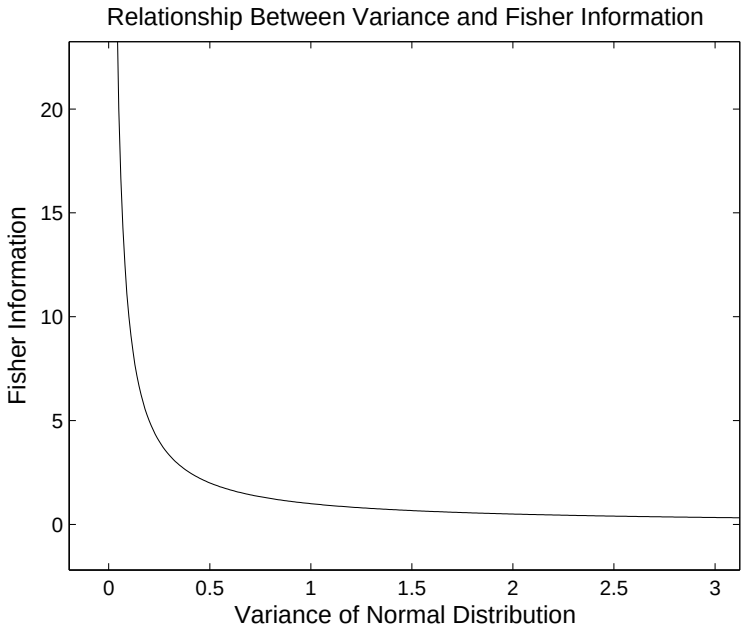

FIGURE 3.4: Relationship between Fisher Information and variance.

The information given by an observation is $H_k^T R^{-1} H_k$ or I_k as defined in 2.51. The determinant or a norm function can then be used to derive a scalar information measure from the FIM. As the information content of an observation is dependent on I_k, it can be related to the sensor configuration through the observation matrix.

The reader is reminded that the inverse of the FIM, $\mathcal{J}^{-1}$, is the CRLB which provides a lower bound on the performance of any unbiased filter (as discussed in section 2.4.6.1). Like Entropy, FI is additive; FI is only generally applicable to continuous distributions and leads to Risk-Prone SM [1]. This is because it selects actions that offer the most information gain regardless of the probability of obtaining it, whereas Entropy leads to Risk-Averse behaviour. Entropy is also applicable to both discrete and continuous distributions and is therefore recommended as an expected sensor utility action in [1].

Another important result is that if the log-likelihood[5] of the posterior state PDF, $\log p(X_{k|k}|a)$, given action a, is used as the utility function in equation (3.7), then the expected utility equates to Entropic information [48].

In many multiple-sensor situations, the expected information to be gained by a potential sensor observation assignment can be therefore be calculated analytically, without performing the observation itself. The information measures presented above have been used in a range of SM contexts, including sensor-target hand-off assignment [47], observer trajectory optimisation for UAVs locating ground features [48], trajectory optimisation

[5]This has been shown to satisfy the axioms of rationality.

for dynamic targets [2], and more recently, for the control of Simultaneous Localisation and Mapping (SLAM) activities [52].

In [47], this work was applied to a continuous area coverage example where two vehicles are controlled in order to maximise information gained about the terrain on which they are travelling. A team of indoor robots is controlled in [51] using a parameterised optimal control solution to the utility maximisation problem in order to observe and localise a number of stationary features . For zero look–ahead management horizons, the framework was extended to control platform dynamics by the MI gradient field.

A variety of different expected posterior target state PDF utility formulations can be derived from the information measures above. For instance, [48] considers a number of functions of the FIM, including the trace, determinant and various functions of the Eigenvalues. Each of these has slightly different attributes, according to how much emphasis they place on prior information.

Similar metrics can be derived for filter predictions and Bayesian classification [1]. In a decentralised DF architecture, each sensor node can reconstruct every other sensor node's partial information and observation information if the necessary communication or computational architecture is present. Both the partial posterior information and the likelihood information can be used to assess the merits of different sensing actions by analysing a priori, the resulting post-observation distributions. Use of the partial posterior information measure may require extra communication or computational load in order to construct the set of local information measures. Use of the likelihood information only assesses the information in the current observation in contrast to the partial information which inherently utilises any information already known about the state. The optimal sensing action is that chosen by maximising the expected utility.

Information measures such as those outlined in this section are widely accepted, and their properties are well known and understood. The popularity of such measures lies in their strong theoretical foundation and flexibility of application (including those with FSTs other than target tracking). Another reason for their popularity is the convenient, and elegant integration with probabilistic state estimation methods.

3.4.2 Kullback–Leibler Divergence Measures

Kullback–Leibler information (also known as discrimination gain or cross-entropy) is defined as the average of the log-likelihood of two distributions, $p(X)$ and $q(X)$:

$$D(p(X), q(X)) = E_{\mathcal{X}} \left\{ p(X) \log \frac{p(X)}{q(X)} \right\}. \tag{3.20}$$

Expected discrimination gain is utilised in [53] in order to determine electronically scanned aperture Radar threshold settings and search orders. A scenario is postulated

where a number of discrete detection cells must be interrogated. In cases where the signal-to-noise-ratio is low, the cells are commonly sampled multiple times to allow detection and classification to occur. The goal of the SM process in this work is defined as optimising the probability densities that produced by the DF system. The discrimination gain can be interpreted as a relative measure of likelihood for different densities. These types of measure have some theoretical links with the information-based measures outlined in Section 3.4.1, however, their use is less widespread. The gain for each cell is predicted using current estimates and the results are used to chose the cell which is to be sampled next. The authors remark that this process is similar to using the Riccati equation to compute the expected covariance for an estimate for different observations.

The work presented in [53] contrasts with many SM research approaches as it directly considers non-kinematic aspects of the autonomous sensing problem (i.e. detection and classification).

3.4.3 Geometric Objective Functions

In [54], a Finite-Set-Statistics (FISST) based approach to SM is developed. The approach concerns the maximisation of a multiple-target Kullback-Leibler discrimination measure and is reported to have a number of computational limitations (i.e. the optimisation can be intractable [55], [56]). The authors extend the theory by proposing a number of Geometric Objective Functions (GOFs) derived from information discrimination functionals which generalize the Kullback-Leibler discrimination under certain assumptions [57]. It is shown that some of the resulting GOFs can be used to develop computationally tractable SM algorithms when used in conjunction with specific state estimation algorithms. A number of experimental tests are offered in [55] to demonstrate the theoretical work in [57].

Two important concepts are defined in [55], [56], and [57], that lead to the derivation of the GOFs. A 'probability generating functional' is used to derive a representation of the probability that a target state is entirely contained within a sensor FOV. The total probability of detection (also termed the total FOV) is defined as the probability that at least one sensor will observe a target. A number of GOFs (geometric in the sense that they are related to a fundamental objective function which measures the geometric overlap of the total multiple-sensor FOV with the predicted state-set) are then proposed as SMOFs. An example GOF is the *predicted belief*, which directly measures the overlap of the total sensor FOV with the predicted state.

The objective functions can be expressed in terms of the probability generating functional of the estimated posterior multiple–target density. A tracker is then used as the prediction step of a control-theoretic SM algorithm which allows the probability generating functional to be expressed in a simple form, leading to tractable objective

functions. Further simplification of the objective functions through various assumptions is suggested.

Geometric Objective Functions such as those outlined in this section provide a means to evaluate possible sensor actions, without using computationally expensive statistical measures. This makes them very attractive from a AV context, where there may be limited processing power available on the platform. It is interesting to note that the conceptual basis behind these measures could be used to simplify the information-based approaches presented in Section 3.4.1. It will be shown later in this thesis that information-based methods often induce changes in the sensor-target geometry. Surrogate measures, based on the geometric properties of a scenario, could, therefore, be used to reduce the cost of computing these measures.

3.4.4 The Posterior Crámer-Rao Lower Bound

As discussed in Section 2, the PCRLB is defined as the inverse of the FIM for the posterior estimate of a random vector, and provides a lower bound on the performance of an unbiased filter. Different sensor actions can be assessed by considering the resulting measurement contribution. For instance, in the case where the observation function is linear, the total measurement contribution from N_s sensors, J_k^Z can be reduced to [58]:

$$J_k^Z = \sum_{i=1}^{N_s} E\{H_k^{i\,T} R^{i^{-1}} H_k^i\} \tag{3.21}$$

The research presented in [58] and [59] utilises the PCRLB for dynamic SM by quantifying and subsequently controlling the accuracy of target state estimation using the bound. The algorithms presented utilise a Riccati-like recursion expression to determine the FIM for the general non-linear filtering problem (as discussed in Chapter 2). The filter RMSE is chosen for the SMOFs. Their approach attempts to dissociate the theory from the heuristics involved in application, and indeed the PCRLB is applicable for different filtering techniques, in a similar fashion to information-based methods. The authors also provide a mechanism for accounting for measurement origin uncertainty (i.e. false alarms) by modelling the uncertainty as a state-dependent information reduction factor under certain measurement distribution assumptions.

An anti-submarine-warfare scenario is considered in [59] where the SM process is divided into three optimisation problems: optimisation of the interval between sensor deployment; and selection of the new sensor locations and the number of new sensors. A further three applications are illustrated in [58], along with a number of recent developments regarding overcoming some of the limitations of the PCRLB. It is shown that the PCRLB can be overly optimistic in real world scenarios (where there is a non-zero probability of missed detections, or when the target state can be multi-modal). The

authors propose an alternative performance measure (APM) which takes account for the order in which the detections and missed detections occur and averages over state evolutions rather than motion model sequences. They demonstrate that the APM is capable of accurately predicting the true filter performance in simulated fast-jet flight planning, ground-moving-target-indicator and electronic-support-measure based scenarios. It is noted that, unlike the PCRLB, the new APM does not necessarily bound the performance of a filtering algorithm.

The algorithms described above have been extended by the author to various other problems, including a fast-jet pursuer evader game [60]. The framework has also been used to analyse the performance gains associated with multiple-step planning. This is discussed in more detail later in this chapter.

3.4.5 State Estimation Covariance

A covariance-control approach to SM for target tracking, where sensors are controlled in order that the state estimation system meets specific estimation bounds, is presented in [61] and [62]. This type of approach deals directly with state estimation requirements, which may derive from system hand-off processes. The difference, ΔP, between the target estimate covariance, $P_{k|k}$, and the desired covariance, P_d, is analysed directly, rather than through information measures.

A number of SMOFs based on the divergence between the desired and actual covariances are discussed in terms of their ability to account for directionality in the target estimate covariance. Three algorithms that seek to find sensor assignments which meet the covariance requirements, while efficiently utilising the sensor suite, are presented: 1) the *Eigenvalue-Minimum Sensor* algorithm - this algorithm selects the sensor combination with the fewest sensors that meets the estimation covariance requirements (determined by testing to see if all the Eigenvalues of ΔP are positive); 2) the *Matrix-Norm* algorithm - this algorithm selects the sensor combination that minimises the norm of ΔP; 3) the *Norm-Sensors* algorithm which relaxes the *Matrix-Norm* algorithm to allow the covariance to vary according to a pre-specified limit, and chooses the sensor combination that uses the fewest sensors and keeps the covariance within this limit.

Controlling covariance in this fashion is equivalent to maximising information gain when the desired covariance is set to all-zero. However, the covariance control method allows specific tailoring of the estimation covariance against predefined performance requirements. These requirements may derive from sensor hand-off processes (such as the hand-off to another tracking system such as a fire control system). In military surveillance tasks, the ability to deal with state estimation performance directly in this manner is of advantage, as it is more intuitive than operating in information space. A further advantage of this approach is the direct integration with secondary systems aspects,

by selecting the sensor combination with the fewest sensors, that meets the estimation performance. In reality, this may equate to a significant saving in power (as well as reducing the probability of intercept).

3.4.6 Non-Myopic Sensor Control Theory

The majority of the work considered in the previous sections can be classed as myopic or greedy control algorithms. Myopic control algorithms are those that compute optimal sensor actions according to single-step predictions of future states, and are, therefore, 'greedy' with respect to immediate increase in utility. It has been shown in a number of studies, that such approaches are sub-optimal, in that there can exist non-myopic strategies (i.e. those that involve multiple-step predictions) that can outperform them. Thus the formulations of utility presented in Section 3.3.2 have to be computed over multiple steps.

The advantages of non-myopic planning are due to the sequential decision making problem that is inherent in many SM scenarios. Consider for instance a single sensor tasked with tracking a number of targets in a complex urban domain where targets can become obscured by terrain and buildings. If the sensor is required to maintain robust tracks on all targets, the SM system must consider the long-term implications of immediate sensor actions. For example, if the majority of the targets are moving in an open area within the domain, yet one of the targets is about to become obscured temporarily, it may be beneficial to observe that target, even if other targets offer more immediate utility (for instance because they are nearer, and offer more potential for information gain). This reflects the sequential decision making nature of many SM scenarios. The exploitation of advanced multiple-step planning strategies represents cutting-edge SM research, as it is these methods that will ultimately offer the highest levels of performance. This section reviews a number of non-myopic SM approaches.

A sub-optimal, heuristic non-myopic sensor control approach is proposed in [63] based on approximating the long-term ramifications of control inputs by a value which approximates the so called 'gain in waiting'. The scenario under consideration involves measurements which can be subject of line-of-sight obscuration induced by sensor movement and terrain. The value is formed of two parts: an indicator function which determines if the future gain of an action is less than the current gain for the action; and a second part which measures the difference between the distribution of gains at those time steps. A positive value for the indicator function encourages the action to be taken now instead of later, and a negative value discourages the action as its application in the future will yield more gain. This approximation essentially measures the difference between myopic gains associated with an action taken now, and the myopic gains associated with the same action taken later. This is described in the work as a means of measuring 'opportunity' or 'regret'.

A weighting factor is used to balance the myopic and non-myopic parts of the resulting utility function. As the weighting factor tends to zero, the system acts myopically, and as the weighting factor tends to infinity, the system considers only the future gains. The author points out that an appropriate value for the weighting factor is one that balances the present and future gain analysis, although no method for finding it is suggested. This approximation has computational complexity which is reported to be linear in the horizon length. The authors also propose an approach to *learn* the value-to-go (value-to-go relates to the expected non-myopic portion of the utility) using the Reinforcement Learning technique.

An example application where an AV tracks a ground target using two different sensor capabilities compares the performance of the myopic and non-myopic strategies (with horizon lengths of a few steps). The sensor models represent an X-band Radar which suffers from line-of-sight obscuration, but provides high accuracy detection, and a high frequency Radar which is capable of detecting through obstructions, but provides lower accuracy observations. Due to the dynamics of the sensor-target geometry, the target is sometimes obscured by environmental terrain. The non-myopic approach is able to predict such obscuration and schedule extra sensor dwells on the target before this occurs, thus increasing the accuracy of the prediction of where the target will emerge. Results show that the best performance is offered by the Reinforcement Learning approach, closely followed by the approximate value-to-go approach. It is difficult to estimate the relative performance of these techniques in situations where different amounts of training data are available.

Other motivating scenarios are described in the author's thesis [64]. One example is a scenario centered around multiple intersecting targets that are being detected and classified by a moving target indicator and synthetic aperture automatic target recognition Radar. When multiple targets occupy the same cell, the moving target indicator sensor returns elevated energy, and the Radar sensor performs poorly. The non-myopic strategy is able to predict target intersection and mitigate against the reduced identification performance by utilising the identification sensor before this occurs.

A variety of methods for efficient implementation are discussed including pruning the resulting decision tree[6] and directed search strategies. Directed search strategies aim to make the most efficient use of a limited number of path samples. A method which selects which path to search in the tree based on how much information is known about that path is also presented[7].

A MC roll-out method is also discussed which has different computational complexity. Although the complexity is independent of the state dimension it is still exponential in both the number of possible actions and the horizon length. It should be noted that

[6] Pruning is the process of removing parts of the tree that contain inferior solutions, and, in the process, decreasing the computational requirements of searching the tree.

[7] An initial search of all paths of the same length begins this process.

the roll-out method is formulated in a slightly different framework from that in [65] and associated works, where roll-out is used to improve on the performance of a base policy. Roll-out in this context is used in the sensor control domain in [66], and related work.

The PCRLB SM framework described in Section 3.4.4 has also been extended to the non-myopic observer trajectory optimisation cases in [35] and related work. A MC roll-out style approach is employed over two steps. In order to avoid the computational costs associated with this approach, an efficient search method is proposed which dramatically reduces the number of PCRLB evaluations that have to be performed.

3.4.7 Solving Sensor Management Optimisation Problems

Sensor Management problems ultimately resolve into complex optimisation problems. Occasionally, specific formulations of certain problems can be reduced so that they can be solved by simpler methods such as linear programming. For example, the sensor hand-off problem in [48] is reduced to a linear assignment problem, which can be solved in polynomial time.

More complex problems however, cannot easily be simplified in this manner, and it is often necessary to resort to stochastic search methods such as GAs [67] or Ant Colony Optimisation [68]. This is due to the multi-modal nature of such problems, which cannot be solved easily using gradient-based approaches. The solution of the sequential decision making problems outlined in the previous section is likely to present even more difficulties for traditional gradient-based optimisation techniques due to the increased likelihood of local minima. One approach to solving these problems is to apply a stochastic search technique over the policy or planning space. Alternatively, efficient search methods and branch and bound strategies can be used as in [64] and [66].

3.5 Concluding Remarks

This chapter presented a review of the SM literature. The sensor control techniques were broadly categorised according to the choice of SMOF and connections between the approaches and decision theory where illustrated. A number of conclusions can be drawn from this review which motivate the contributions presented in this thesis.

It is clear that information-based sensor control methods are well established in the literature and provide a convenient and theoretically elegant means to integrate the state estimation and SM processes. However, an important aspect of such work is the method by which the state estimation control process is integrated with other system planning activities, such as mission planning, and other system objectives. These issues are of great importance to the AV community, as it is vital to understand the performance

of the platforms at a system-level. This concern is compounded by the fact that many secondary system-level objectives are likely to conflict with the primary state estimation objective. To date, there has been little work carried out in this area, for instance there are no commonly adopted methods for trading-off power consumption in observer trajectory control, with state estimation performance. The closest studies of relevance are those presented in [61] and [62], but the results have not been extended to the mobile sensor case. In addition, relatively few approaches examine the impact of practical sensor control constraints, which adds to the motivation to consider the observer control problem from a broader optimisation context.

The use of non-myopic planning strategies is ultimately expected to provide state of the art SM performance, and thus the development of such strategies is of great interest. However, despite a number of studies into these areas, there remain a number of issues which need to be addressed. Firstly, to date, there has been no direct study into the fundamental mechanisms that drive the performance of non-myopic strategies in SM scenarios. For instance, there is no open literature method which is able to model the increase in utility associated with increasing the lookahead of multiple-step planning strategies. In addition, there is little to no understanding of the *limits* of these approaches, and how far the increase in expected performance can be pushed. In fact, it will be shown later in this thesis that there is a primary limiting mechanism relating to the propagation of error through the control feedback loop, that limits the gains that can be achieved.

Finally, it is observed that, as the AV field continues to mature, and multiple-platform systems become more flexible, the nature of the optimisation problem that the sensor control problem yields is likely to become more complex. This is especially pertinent when considering AV systems that may be deployed for multiple tasks. In such cases, the online optimisation problems may change over time (as may the constraints) and thus it is important that flexible optimisation techniques are available. It is noted that many stochastic search techniques require significant levels of tuning, and thus may not prove appropriate for use in autonomous systems. Optimisation algorithms that require limited tuning requirements are therefore of great interest.

The conclusions presented above motivate the contributions presented in this thesis, which can be found in Chapters 4 to 6.

Chapter 4

Multiple-Objective Observer Trajectory Control

4.1 Introduction

The management of real-world sensor platforms to optimise perceptive tasks is constrained by numerous factors including limited power resources, dynamic constraints, and the necessity for the platform to maintain operation (or to survive). These issues are particularly relevant in military AV applications as the environment in which such platforms operate may be extremely hostile, and there may be no access to power sources after the vehicle has been deployed (low cost AVs may be used in a deploy and forget context). In civilian and security applications, there may be many cases where the platform may have access to a power source, for example a recharging station. The availability of a recharge station for the vehicle renders the optimisation of power-based objectives a tool with which to extend the time period for which the vehicle can operate for without returning to base (and therefore the maximum range). In both cases there is, of course, a requirement to minimise *cost* and thus developing power efficient systems is generally of significant interest.

While the sensor control algorithms outlined in Chapter 3 provide an elegant means to optimise the state estimation or information gathering process, it is somewhat unclear how the performance and applicability of such algorithms is affected by the practical limitations of real systems. Many secondary objectives, such as those relating to minimising power consumption, will directly conflict with those that aim to improve information gathering. This is because optimising state estimation performance typically requires energy to be expended to alter the sensor-to-target geometry (i.e. physically move the sensor platform). Different information gathering SMOFs result in different alterations to the geometry and thus different power consumption (depending on the platform power model).

Many system designs may attempt to bridge the gap between these practical aspects and the underpinnings of Chapter 3 by utilising a 'mission management' layer. The nature of this control layer must be determined through an understanding of the interaction between the state estimation optimisation framework and the process by which secondary objectives are managed. A number of questions arise from this consideration:

- How can the conflicting aims of information gathering and other secondary objectives be balanced and traded-off appropriately?
- How does the assumed model of the preference structure between information gain and other objectives affect overall performance?
- How does non-myopic planning impact on the optimisation of secondary objectives?

This chapter presents a number of contributions based on the above issues for ground-based AV scenarios. The motivation for an autonomous UGV scenario is based on the observation that ground-based platforms are not typically constrained to maintaining constant minimum velocities as is the usually the case with UAVs. Thus the management of power resources is likely to have a more serious impact on the global system design as there is a great opportunity to exploit this property of the physical nature of the scenario (i.e. maximise the opportunity to expend very little power by keeping the platform stationary).

In order to study the mechanisms outlined above, the SM problem is formulated as a constrained, multiple-objective optimisation problem. Multiple-objective optimality can be defined in a number of different ways [69], [70], and the resulting impact on system performance associated with these definitions is considered. The research presented in this thesis adopts a mobile sensor control framework similar to that in [50], whilst taking a pragmatic approach considering secondary objectives as in [61], for the mobile sensor case. The primary differentiator in this work is the consideration of the power issues related to a *mobile* platform combined with different multiple-objective optimisation formulations. The material presented in this chapter corresponds to the performance measures and control aspects of the generic SM architecture used throughout this thesis. This is illustrated by the sub-systems highlighted in red in Figure 4.1

The following contributions are derived from this work:

- a study of the impact of the choice of multiple-objective preference scheme on tracking performance and secondary survivability performance;
- and, a multiple-objective optimisation-based UGV observer trajectory optimisation algorithm and a comparison of its performance against standard greedy approaches.

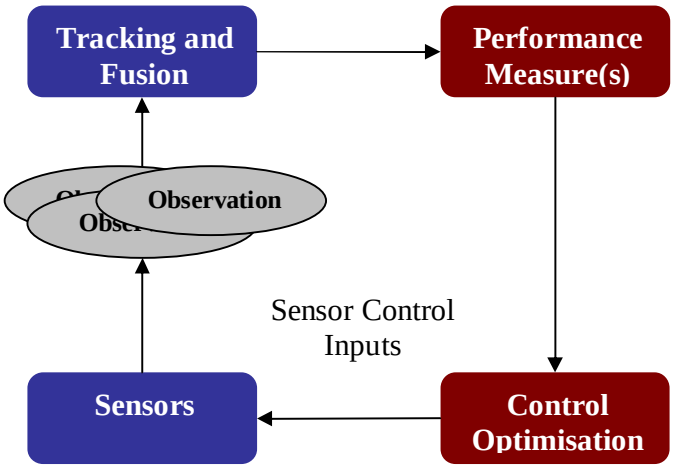

FIGURE 4.1: Sensor Management system framework.

4.2 Chapter Outline

This chapter is organised as follows. Section 4.3 introduces the assumed scenario and the underlying theoretical system architecture on which the subsequent analysis is based. Details on the selected target and sensor models, platform motion models (including power models) are to be found in Section 4.4. The assumed state estimation framework is described in Section 4.5. Details regarding the state estimation objectives and the secondary performance objectives relating to power consumption and survivability are presented in Section 4.6. The multiple-objective optimisation formulation is introduced in Section 4.7 and the different approaches to defining a preference structure between the objectives are considered in Section 4.7.2. The conclusions drawn from this analysis motivate the development of a minimax observer control algorithm, which is derived in Section 4.7.5. The results of application of the resulting algorithm are then presented in Section 4.7.6. Section 4.8 examines the potential benefits of utilising multiple-step planning strategies to the alleviate some of the limitations of the proposed algorithm. Concluding remarks are given in Section 4.9.

4.3 Scenario and System Architecture

The scenario that is at the heart of this chapter involves a ground-based surveillance problem. It is assumed that a group of UGVs is tasked with detecting and tracking targets of interest which enter a predefined local domain. This kind of scenario is relevant to both the military and civilian fields. Example applications include infrastructure or area protection, base patrol, and hostile environment surveillance. An illustration of the scenario is provided in Figure 4.2.

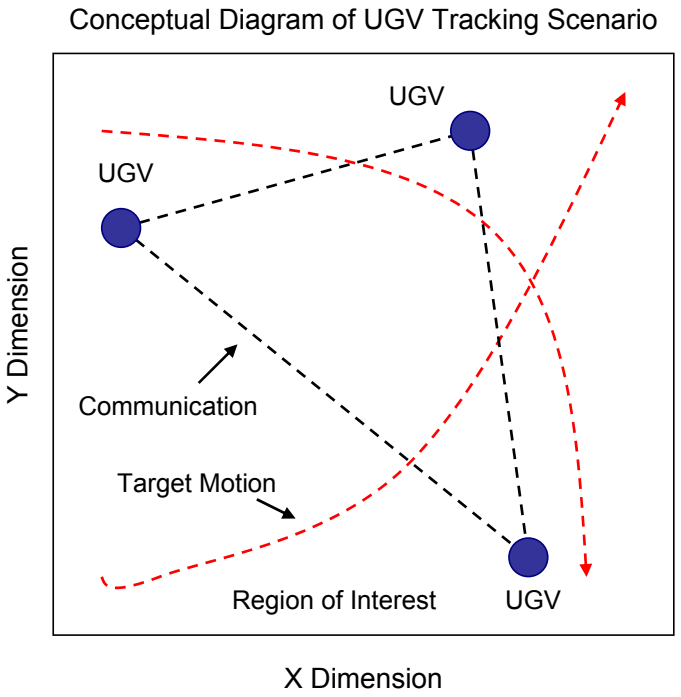

FIGURE 4.2: Ground-based surveillance scenario.

The red traces in the diagram indicate target motion across a Region of Interest (ROI), the blue markers represent mobile autonomous UGVs, and the dashed black traces indicate communication between the sensors. A range-bearing target tracking process is assumed, based on a DIF framework. The motivation for such a framework is based on its well understood properties and common use in the SM community.

The processing system architecture assumed for this problem is based on the following components: 1) a dynamic target model; 2) a sensor observation model and associated constrained platform dynamic model; 3) a state estimation framework which yields target state estimates from the sensor observations; 4) a set of performance objectives for potential platform states; and, 5) an optimisation algorithm which computes optimal platform control inputs. Due to the nature of the sub-components, this model can be broadly considered as a Model Predictive Control (MPC) architecture. A generic MPC based SM model is illustrated in Figure 4.3 for the military UGV application context. The links between this kind of problem and MPC theory are studied in more detail in Chapter 5.

The operation of such a system is as follows. A level of situational awareness (an understanding of 'what is going on' in the ROI) is derived from the output of the multiple-sensor fusion system, which yields target state estimates from observed data. In the initialisation stages of such a system, there may not be any valid tracks, and thus the awareness may derive from prior knowledge, or else the system may be tasked to search the ROI for targets. The awareness is utilised by a mission planner stage

to derive appropriate SMOFs and constraints for the sensor control optimisation stage. The optimisation stage also takes inputs from a predictive model (based on the results of fusion), and computes optimal sensor control inputs. The control inputs are actuated, and the fusion system then assimilates the new data.

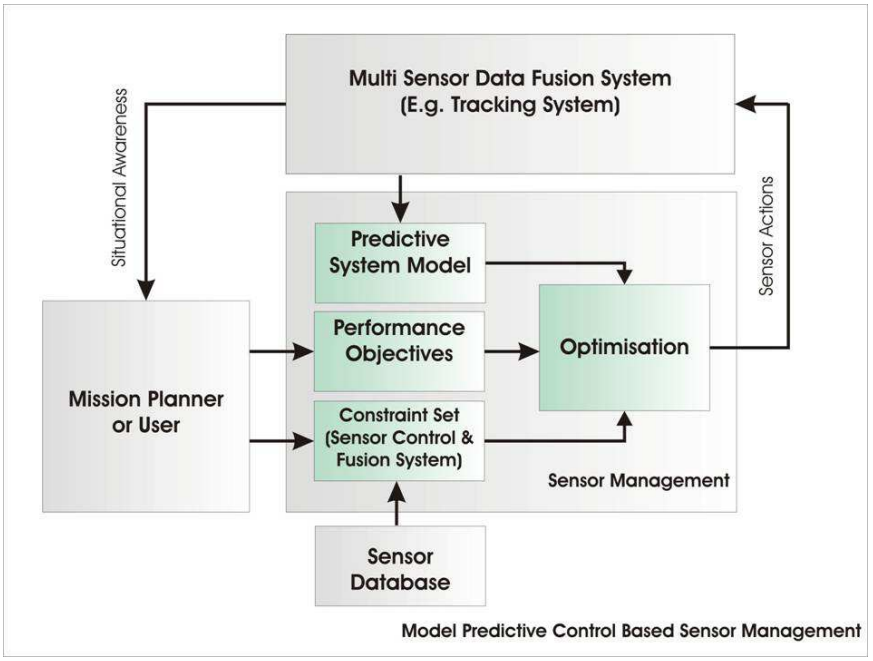

FIGURE 4.3: Predictive control process structure.

The target and observation models map to several parts of this architecture, including the predictive model, the DF system and the optimisation stage. The platform dynamical model and constraint set is primarily utilised in the optimisation stage[1].

4.4 Platform Dynamics and Power Models

This section presents the assumed platform dynamics and power consumption models which will subsequently be used in the control optimisation process.

4.4.1 Platform Dynamics

The control parameterisation or sensor control action matrix, $a_{k+1|k}$, which consists of the required sensor control inputs for N_s sensors is defined as follows:

$$a_{k+1|k} = \begin{bmatrix} x_{k+1}^{s,1}, \ldots, x_{k+1}^{s,N_s} \\ y_{k+1}^{s,1}, \ldots, y_{k+1}^{s,N_s} \end{bmatrix}, \tag{4.1}$$

[1]It is noted that if the sensor control dynamics were non-deterministic, it would be necessary to include a model of the associated uncertainty in the DF processing model.

where $x_{k+1}^{s,i}$, and $y_{k+1}^{s,i}$, for $i = 1, \ldots N_s$, are the two-dimensional Cartesian sensor coordinates for the next observation step. As described in Chapter 2, this model is aimed at abstracting away from any particular platform motion actuation process.

4.4.2 Platform Dynamical Constraints

The motion of a real sensor platform is often limited by a number of dynamical and physical constraints. In this work, two primary constraints are considered: the limited velocity of the platforms, and an absolute minimum separation between platforms. The velocity of the platform will clearly be constrained by the capability of its motion system and power reserves, and the minimum separation constraint reflects the fact that two platforms cannot occupy the same physical space.

To model these constraints, the maximum displacement a sensor platform can perform in a single action step is limited through a *maximum platform Euclidean displacement criterion*, D_{max}^i. In reality, this will be defined by the specifications of the sensor platforms. The constraint is expressed formally as:

$$\sqrt{(x_{k+1}^{s,i} - x_k^{s,i})^2 + (y_{k+1}^{s,i} - y_k^{s,i})^2} \leq D_{max}^i \quad \forall i. \tag{4.2}$$

In addition, a *minimum sensor platform separation criterion*, S_{min}, is imposed, again defined according to the sensor suite specifications. This will prevent the algorithm from requesting unfeasible platform positions (for example overlapping sensor locations). This constraint is defined as follows:

$$\sqrt{(x_{k+1}^{s,i} - x_{k+1}^{s,j})^2 + (y_{k+1}^{s,i} - y_{k+1}^{s,j})^2} \geq S_{min} \quad \forall i, j, i \neq j. \tag{4.3}$$

It is noted that this particular formulation for the constraint ensures that, for each platform i, no other platform moves within a circle of radius S_{min}, centred on platform i. In effect, this constraint implies that the sensor platforms have a circular physical boundary in the x, y plane. Other formulations can be derived for different shaped platforms but this is a reasonable approximation for many scenarios.

4.4.3 Platform Power Models

In order to study how efficient sensor control algorithms are with respect to power, it is necessary to formulate a model for the power consumption of a UGV. In this work, it is assumed that the sensing process operates at fixed, regularly spaced sampling intervals, and is, therefore, independent of sensor control. It is also assumed that processing load is constant and thus the power consumption relating to the processing node itself is fixed. These assumptions are somewhat difficult to justify for fusion systems that have been designed so that not all sensors are activated at every time step, which reduces

power drain from both sensing and processing. However, it is likely that the variation in power consumption due to physical motion of the platform will be more significant than variations in power due to changes in processing load (a modern embedded hardware processing solution may consume as little as a few Watts). For this reason, power drain due to platform motion is the selected aspect of investigation.

A basic model of a mobile sensor's power consumption can then be derived as follows. The power consumption of the platform, in moving from its position at time k to its desired position at time $k+1$, Pw_{k+1}, can be modelled as follows [71]:

$$Pw_{k+1}(m_p, v_p, a_p) = P_l + m_p(a_p + g\mu_{gf})v_p, \tag{4.4}$$

where μ_{gf} is the ground friction constant, m_p is the mass of the platform, v_p and a_p are the platform's velocity and acceleration respectively, P_l is the transmission power loss and g is the gravitational constant. The following assumptions are made to simplify the analysis: there is no transmission loss; the platform accelerates instantly to full velocity; and, any motion that the platform undertakes is at constant velocity. In this case the power consumed is directly proportional to the distance travelled:

$$Pw_{k+1}(m_p, v_p, a_p) = m_p g \mu_{gf} v_p. \tag{4.5}$$

It was shown in [71], and associated works, that this model is a reasonable approximation to the true power consumption of a Pioneer 3DX Mobile Robot. This is a typical mobile robot used in SM research, and is assumed to represent the kind of platform that might be deployed in the scenarios of interest.

4.5 State Estimation Framework

A DIF is used as the basis for the state estimation and fusion architecture. The DIF is widely used across the SM community and its properties are well understood.

Each platform is equipped with a range-bearing sensor and thus each observation is derived as in Equation 4.5:

$$z_k^i = \begin{bmatrix} r_k^i \\ \theta_k^i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k^{s,i} - x_k^t)^2 + (y_k^{s,i} - y_k^t)^2} \\ \tan^{-1}\left\{\frac{y_k^{s,i} - y_k^t}{x_k^{s,i} - x_k^t}\right\} - \psi_k^s \end{bmatrix} + w_k^i.$$

It is further assumed, without loss of generality, that the sensors are omni-directional (i.e. they have a 360-degree FOV).

The range-bearing observation model is non-linear in the target state, but can be converted to a linear model through one of two methods. Firstly, a Decentralised Extended

Information Filter (DEIF) form can be used, by differentiating the observation model about the current expected target state, as described in Chapter 2. An alternative approach, is to convert the polar observation model into a Cartesian observation model, which involves a rotation of the observation noise covariance matrix. Consider the DIF update equation which defines how the global information matrix is updated by assimilating knowledge of the performance of all sensors. In the case where a DEIF is used, the linear observation matrix becomes a function of the expected state of the target being observed, and therefore of time:

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + \sum_{i=1}^{N_s} H_k^{i,t} R^{i^{-1}} H_k^{i,t^T}. \tag{4.6}$$

In the case where the polar-Cartesian transformation is used, C becomes fixed, but the noise covariance matrix becomes a function of the expected target state, and thus time:

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + \sum_{i=1}^{N_s} H^i R_k^{i,t^{-1}} H^{i^T}. \tag{4.7}$$

In either case, the updated information matrix is a function of the sensor-to-target geometry through $x_k^{s,i}$ and $y_k^{s,i}$, and thus dependent on any control actions. The reader is referred back to Chapter 3 for information regarding how the equations above relate to information gain and other performance measures associated with state estimation performance.

4.6 Performance Objectives

Two classes of FSPMs and therefore SMOFs are considered in this chapter. Firstly, objective functions relating to the primary perceptive task, tracking, are considered. Secondly, other objectives relating to expected sensor lifetime are considered. For the latter class, the specific objectives chosen for analysis are power consumption and sensor separation. These are generic objectives with broad application to many fusion scenarios. It is not unreasonable to conjecture that these are two of the most important secondary objectives for such a system.

Power consumption directly affects how long the platform can operate for. As described earlier in this thesis, in the military context platforms may be deployed and subsequently left with no access to power. In the non-military domains, power consumption directly controls the expected range of the vehicle, and the time-frame it can operate in before returning to a base-station.

Sensor separation is a key driver in hostile environments, as sensor suites that are in close proximity have a high chance of being immobilised, or destroyed, by a single attack

from hostile fire. If the sensors can maintain separation, the expected lifetime can be increased.

4.6.1 Tracking Performance Measures and Requirements

The primary perceptive objective in this scenario is to maximise the state estimation accuracy. As discussed in Chapter 3 there are many possible choices of performance measure for optimising target track accuracy. In a scenario such as the one considered here, it is important to consider *how much* performance may be required. This is equivalent to establishing a requirement for tracking accuracy[2]. As was shown in [72], there are various ways in which such a requirement can be developed. Broadly speaking, if the exact nature of the resulting covariance is of primary importance, then a full specification of the desired covariance matrix must be employed. In many cases, however, a measure such as the trace or determinant of the covariance matrix can be used.

The fusion performance requirement in this work is specified through the use of a scalar threshold on the trace of the next-step DIF posterior state estimate covariance matrix:

$$trace\left(P_{k+1|k+1}\right) \leq \beta, \tag{4.8}$$

where β defines the tracking performance requirement and $P_{k+1|k+1}$ is the expected value of the next-step estimation covariance matrix.

4.6.2 Survivability Objectives

Two different power consumption cost functions are defined in this work. The first, ϕ_1, is the *total* sensor suite power utilisation function, $\xi(a_{k+1|k})$, which penalises sensor suite actions that require large overall power usage:

$$\phi_1 = \xi(a_{k+1|k}) = \sum_{i=1}^{N_s} Pwn(a^i_{k+1|k}), \tag{4.9}$$

where $Pwn(a^i_{k+1|k})$ is the power required to actuate the desired action vector $a^i_{k+1|k}$, and is calculated using equation (4.5). As (4.5) shows that the power consumed is proportional to the distance moved, the following surrogate function based on the Euclidean-norm is assumed:

$$Pwn(a^i_{k+1|k}) = \sum_{i=1}^{N_s} \sqrt{(x^{s,i}_{k+1} - x^{s,i}_k)^2 + (y^{s,i}_{k+1} - y^{s,i}_k)^2} \tag{4.10}$$

The nature of this cost function is illustrated in Figure 4.4. However, the main motiva-

[2]Typically, tracking requirements may derive from sensor track hand-off, or hand-off to a fire-control system.

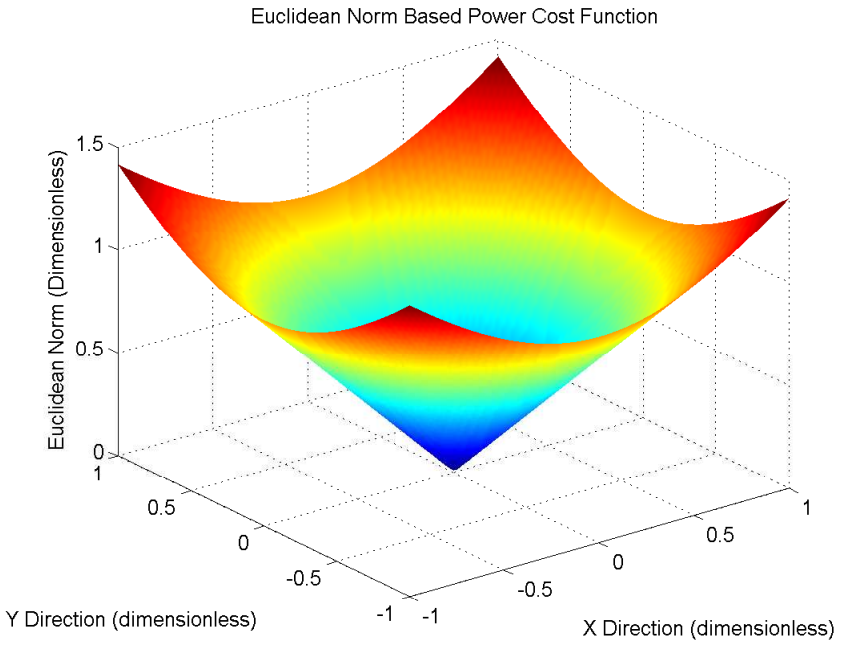

FIGURE 4.4: Euclidean-norm-based power cost function.

tion for minimising power is to prolong the life of the sensor suite, and the total power analysis does not take account of the variation in power reserves between different sensor platforms. An alternative approach therefore is to minimise the total *proportion* of the sensor suite power resources utilised by the sensor action. This penalises sensor actions that require large power usage relative to the sensor power reserves. Here, $\xi(a_{k+1|k})$ can be replaced with the following equation:

$$\tilde{\xi}(a_{k+1|k}) = \sum_{i=1}^{N_s} \frac{Pwn(a^i_{k+1|k})}{Pr^i}, \tag{4.11}$$

where Pr^i is remaining power associated with sensor i. The proportional measure penalises power consumption by an increased factor as the power reserves decrease. The intuitive explanation behind this is that power consumption should be more heavily penalised when the remaining power reserves are low. This has an impact on the global optimisation strategy as it will limit the resources available to maximise other objectives when the power reserves for the platform run low.

As described above, the survivability of the sensor suite can be increased by separating sensors, thus making them less likely to be destroyed by a single hostile attack. A further justification is that multiple sensors must survive in order to exploit the benefits of data-fusion methods. An additional secondary objective, ϕ_2, is therefore the Euclidean sensor

separation, $\varsigma(a_{k+1|k})$:

$$\phi_2 = \varsigma(a_{k+1|k}) = -\sum_{i=1}^{N_s} \sqrt{(x_{k+1}^{s,i} - x_{k+1}^{s,j})^2 + (y_{k+1}^{s,i} - y_{k+1}^{s,j})^2} \quad \forall i, j, i \neq j, \tag{4.12}$$

Various other objectives of this nature can be derived, such as maximising the minimum separation.

4.7 Multiple-Objective Optimisation Formulations

In contrast to single-objective optimisation, which is the traditional framework observed in SM research, multiple-objective optimisation problems, such as the one presented in this chapter, typically involve competing objectives and thus do not usually yield unique solutions (otherwise the objectives would not be competing). For example, the state estimation objectives and secondary objectives presented in this chapter compete directly with each other. This section will examine a number of multiple-objective formulations for the SM example presented above. The section begins with some preliminary background theory and follows with an analysis of different solution approaches.

4.7.1 Pareto Optimality

At the highest level, multiple-objective optimisation problems have *sets* of solutions. Such solutions are known as Pareto-optimal solutions (also known as efficient or non-dominated solutions) [69]. A solution is Pareto-optimal if there exists no other solution that is superior to it with respect to any objective function, without being inferior in another. An illustration of the set of Pareto solutions to an arbitrary optimisation problem with two objectives is given in Figure 4.5. The set of Pareto-optimal solutions lies on the so called Pareto-frontier or non-dominated frontier. The trade-off surface between multiple objectives may not necessarily be convex. Consider for instance the region of the surface between points A and B. This region of the boundary is convex, however, the region of the surface between points B and D is non-convex. It will be shown later in this chapter that this has important implications in regard to the selected optimisation technique.

The points A-D can also be used to understand the notion of Pareto-optimality, and the more basic concept of dominance. Dominance is a comparative measure for candidate solutions to a multiple-objective optimisation problem. A dominated solution is one that is sub-optimal, in that another solution exists which provides superior performance for *all* objectives. The concept of a non-dominated solution is slightly more complex; it is a solution which cannot be beaten by any other solution in any objective, without the other point being worse in another. Consider for example candidate points A and B.

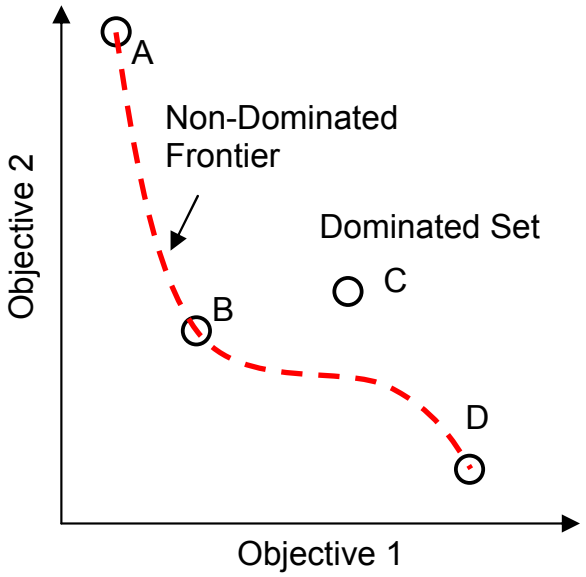

FIGURE 4.5: Two objective optimisation problem in objective space.

While point A is clearly superior to point B in regard to objective 1 (from a minimisation perspective), B is obviously superior with respect to objective 2 . These two solutions are both Pareto-optimal, as all other solutions must be worse in *at least one* objective. Point C, however, is sub-optimal as it is surpassed in both objectives by point B. Candidate solution D is Pareto-optimal as it cannot be beaten in relation to objective 1. Dominance is expressed formally as follows:

Definition 4.1. Let $S^a = [s_1^a, \ldots s_{N_o}^a]$ and $S^b = [s_1^b, \ldots s_{N_o}^b]$ be two candidate solution vectors to an optimisation problem with N_o objectives. S^a is said to dominate S^b if, and only if, the following two conditions are satisfied:

$$s_i^a \leq s_i^b \quad \forall i \tag{4.13}$$

$$s_i^a < s_i^b \quad \text{for at least one objective.} \tag{4.14}$$

The definition of a Pareto-optimal set follows naturally from the above:

Definition 4.2. The set of Pareto-optimal solutions is the set of all solutions for which there exist no other solutions that dominate them.

4.7.2 Preference Formulations

The generic multiple-objective SM optimisation problem can be specified mathematically as follows:

$$a^{\star}_{k+1|k} = \arg \min_{a_{k+1|k} \in \mathcal{A}} f(\phi_1, \ldots, \phi_{N_o}) \tag{4.15}$$

$$s.t. \quad a_{k+1|k} \in \Omega, \tag{4.16}$$

where Ω is the feasible set, i.e the set of solutions which satisfy all constraints. The presence of the function, f, in the above equation denotes the use of an a priori preference structure, e.g. the ranking of objective functions, and the specification of how the solution should address them. The preference function permits the conversion of the problem in equation (4.15) to a single-objective problem, that can be solved using standard non-linear optimisation techniques. It is noted that in some cases, there may be a lack of preference function information, e.g if there is no prior knowledge regarding the nature of the trade-off surface, or if the surface is changing due to a dynamic optimisation problem. In these situations, other methods such as Pareto-optimisation techniques must be employed. These techniques attempt to identify the non-dominated frontier, in order that this information can be used to develop preference relations (these methods are sometimes known as a posteriori articulation of preference [73]). This is the approach adopted in the work presented in Chapter 6. Another approach is to develop the preference relations online, by interaction with the optimisation process [69]. The focus of this chapter, however, is on the relative merits of different preference formulations for the assumed SM problem.

There are many different methods to account for the preference structure. Broadly speaking they can be categorised into the following classes:

- Weights - each objective is assigned a weight which reflects its relative importance.
- Priorities - each objective is assigned a priority which dictates its place in an ordered succession of single-objective optimisations.
- Goals - each objective is assigned a goal which must be achieved.

These types of approaches will be examined in more depth for the assumed SM problem in the following sections.

4.7.3 Weighted-Sum Approach

The weighted-sum approach is the baseline procedure for solving a multiple-objective optimisation problem. It is based on the minimisation of a simple weighted combination

of the individual objectives:

$$a^{\star}_{k|k+1} = \arg \min_{a_{k+1|k} \in \mathcal{A}} \sum_{i=1}^{N_o} w_i \phi_i \tag{4.17}$$

$$s.t. \quad a_{k+1|k} \in \Omega, \tag{4.18}$$

where the weights are used to represent the relative preference structure for the objectives. For example, assuming the objectives are given by those defined in Sections 4.6.1 and 4.6.2, the relative weights in equation (4.17) would have to be chosen in order to balance information gathering with survivability measures. This yields the generic weighted-sum observer control algorithm:

Algorithm 1 Generic Weighted-Sum Observer Control

loop

Predict the target states

Compute the optimal sensor action according to:

$$a^{\star}_{k|k+1} = \arg \min_{a_{k+1|k} \in \mathcal{A}} \sum_{i=1}^{N_o} w_i \phi_i \tag{4.19}$$

$$s.t. \quad a_{k+1|k} \in \Omega, \tag{4.20}$$

Actuate the sensor actions

Observe the new data

Update the states

end loop

It has been shown that for any set of positive weights, the solution generated by equation (4.17) is always a Pareto-optimal solution [69]. The weighed-sum approach can be understood geometrically as a line defined by the chosen weights which intersects the Pareto-frontier [69].

The difficulty with this approach is that it is non-trivial to compute weights which will yield appropriate state estimation performance. A further justification for some problems is that the method is, by nature of the convex combination in equation (4.17), unable to explore any portions of the Pareto-frontier that are non-convex.

In order to calculate appropriate weights it is necessary to know a priori what the relative magnitudes of the individual objective performance will be for the candidate solutions. While this is possible in the problem assumed in this chapter, it still remains to derive an appropriate mapping from expected magnitudes to weights.

4.7.4 Minimax Approach

In cases where the objective space is non-convex, the simple weighting method may not be capable of generating solutions on the non-convex part of the Pareto-frontier. An alternative, more general approach, is to formulate a *minimax* (also known as ∞-norm or ideal point) solution [69] to the problem in (4.15), which is capable of finding *all* solutions regardless of whether the objective-space is convex. The minimax method selects a feasible solution such that the combined deviation from the ideal solution is minimised in the sense of the ∞-norm [69]:

$$a^{\star}_{k|k+1} = \arg \min_{a_{k+1|k} \in \mathcal{A}} \max \left\{ \frac{\phi_i(a_{k|k+1}) - g^{\star}_i}{w_i} \right\} \tag{4.21}$$

$$s.t. \quad a_{k+1|k} \in \Omega, \tag{4.22}$$

where $g^{\star}_i$ is the ideal point for objective i, i.e. the solution to:

$$g^{\star}_i = \arg \min_{a_{k+1|k} \in \mathcal{A}} \phi_i(a_{k|k+1}) \tag{4.23}$$

$$s.t. \quad a_{k+1|k} \in \Omega. \tag{4.24}$$

However, it is not always guaranteed to produce non-dominated solutions. In addition, the minimax function is not readily differentiable in many cases, which presents a problem for traditional gradient-based solvers. For the purposes of this work, however, this problem is not an issue, as it is the inherent preference structure that is important. Geometrically, the approach can be interpreted as finding the point at which the Pareto-frontier intersects any of the edges or nodes of increasingly large hyper-rectangles centered on the ideal point. An alternative representation of the minimax method based on auxiliary variables can be found in [69].

4.7.5 Minimax Observer Control Algorithm

The previous sections have highlighted two popular approaches to formulating preference structures for multiple-objective optimisation problems, according to the specification of weights or goals for the objectives. It is noted, however, that both approaches present difficulty in the observer control context, due to the challenges associated with defining the weights. A more transparent approach is to formulate the primary perceptive objective, i.e. state estimation performance, as a constraint. The remaining survivability objectives can then minimised using a minimax approach using the candidate solutions which satisfy the performance constraint. As described earlier, the tracking performance constraint may take the form of a function of the apparent filter covariance, for example the trace or determinant.

A minimax observer trajectory control algorithm is motivated by these conclusions. The algorithm is summarised as follows:

- Predict the future target state:

$$\hat{\mathrm{y}}^t_{k+1|k} = P^{-1}_{k+1|k} A^t_k P_{k|k} \hat{\mathrm{y}}^t_{k|k} \tag{4.25}$$

$$\hat{X}^t_{k+1|k} = P_{k+1|k} \hat{\mathrm{y}}^t_{k+1|k} \tag{4.26}$$

- Compute the optimal sensor action, according to:

$$a^\star_{k+1|k} = \arg \min_{a_{k+1|k} \in \mathcal{A}} \max \left\{ \frac{\phi_i(a_{k+1|k}) - g^\star_i}{w_i} \right\} \tag{4.27}$$

$$s.t. \quad a_{k+1|k} \in \Omega, \tag{4.28}$$

where:

$$\phi_1 = \xi(a_{k+1|k}) = \sum_{i=1}^{N_s} Pwn(a^i_{k+1|k}) \tag{4.29}$$

$$\phi_2 = \varsigma(a_{k+1|k}) = -\sum_{i=1}^{N_s} \sqrt{(x^{s,i}_{k+1} - x^{s,j}_{k+1})^2 + (y^{s,i}_{k+1} - y^{s,j}_{k+1})^2} \quad \forall i, j, i \neq j \tag{4.30}$$

, (4.31)

and the constraint set is defined as:

$$\Omega = \left\{ \begin{array}{l} \sqrt{(x^{s,i}_{k+1} - x^{s,i}_k)^2 + (y^{s,i}_{k+1} - y^{s,i}_k)^2} \leq D^i_{max} \quad \forall i \\ \sqrt{(x^{s,i}_{k+1} - x^{s,j}_{k+1})^2 + (y^{s,i}_{k+1} - y^{s,j}_{k+1})^2} \geq S_{min} \forall i, j, i \neq j \\ trace\left(P_{k+1|k+1}\right) \leq \beta \end{array} \right\} \tag{4.32}$$

where $P_{k+1|k+1}$ is computed using:

$$P^{-1}_{k+1|k+1} = P^{-1}_{k+1|k} + \sum_{i=1}^{N_s} E_{\mathcal{X},\mathcal{Z}}\{H^{i^T} R^{i\,-1}_k H^i | a_{k+1|k}\}. \tag{4.33}$$

- If a solution to the above problem cannot be found, then compute:

$$\tilde{a}^\star_{k+1|k} = \arg \max_{a_{k+1|k} \in \mathcal{A}} \mathcal{I}(\hat{X}^t_{k+1}, Z_{k+1} | a_{k+1|k}) \tag{4.34}$$

$$s.t. \quad a_{k+1|k} \in \tilde{\Omega}, \tag{4.35}$$

where the relaxed constraint set is defined as:

$$\tilde{\Omega} = \left\{ \begin{array}{l} \sqrt{(x^{s,i}_{k+1} - x^{s,i}_k)^2 + (y^{s,i}_{k+1} - y^{s,i}_k)^2} \leq D^i_{max} \quad \forall i \\ \sqrt{(x^{s,i}_{k+1} - x^{s,j}_{k+1})^2 + (y^{s,i}_{k+1} - y^{s,j}_{k+1})^2} \geq S_{min} \forall i, j, i \neq j \end{array} \right\}, \tag{4.36}$$

- Actuate the sensor control, $a^{\star}_{k+1|k}$, or $\tilde{a}^{\star}_{k+1|k}$
- Observe the new data:

$$z^i_{k+1} = \begin{bmatrix} r^i_{k+1} \\ \theta^i_{k+1} \end{bmatrix} = \begin{bmatrix} \sqrt{(x^{s,i}_{k+1} - x^t_{k+1})^2 + (y^{s,i}_{k+1} - y^t_{k+1})^2} \\ \tan^{-1}\left\{ \frac{y^{s,i}_{k+1} - y^t_{k+1}}{x^{s,i}_{k+1} - x^t_{k+1}} \right\} - \psi^s_{k+1} \end{bmatrix} + w^i_{k+1},$$

- Update the state estimate

$$\hat{\mathrm{y}}^t_{k+1|k+1} = \hat{\mathrm{y}}^t_{k+1|k} + \sum_{i=1}^{N_s} {H^i}^T {R^i_{k+1}}^{-1} z^i_{k+1}, \tag{4.37}$$

$$P^{-1}_{k+1|k+1} = P^{-1}_{k+1|k} + \sum_{i=1}^{N_s} {H^i}^T {R^i}^{-1}_{k+1} H^i. \tag{4.38}$$

- **Repeat**

Enforcing a predefined state estimation performance requirement as an optimisation constraint will clearly lead to cases where a feasible solution cannot be identified (this is particularly likely in the early stages of a tracking process where very few observations will have been conducted, and thus the tracking performance may be well below the desired limits). To account for this problem the state estimation performance constraint in the algorithm presented above is relaxed when a solution that meets the constraints cannot be found. Specifically, when a solution cannot be found, the algorithm searches for the sensor action which yields the most MI. Under these conditions the algorithm reduces to the standard greedy information gain approach.

The justification for this relaxation is that the primary objective must be satisfied as a priority, and that power consumption and sensor separation are sacrificed until the tracking performance requirements have been met. The trade-off between power and separation, therefore, only applies if at least one sensor action that meets the covariance matrix trace threshold can be found. If the constraint can be fulfilled then the algorithm chooses the solution which provides the best balance between the remaining survivability objectives, according to the preference inferred by the relative weights.

This is an intuitive approach because the primary objective is to meet the state estimation requirement, and the algorithm presented above achieves this directly without having to model the effect of weighting parameters. It is noted that various other techniques to relax the constraint could be employed. For example, the tracking performance constraint could be specified within an upper and lower limit in order to provide some additional flexibility.

4.7.6 Example Problem

In this section, example results from application of the observer control algorithm to a ground-based AV tracking scenario are presented. It is noted that some of the results presented herein were first published in [74]. In the following simulation-based experiment, two ground-based AV platforms equipped with range-bearing sensors are tasked with tracking targets which enter the ROI which has a size of $\{1000 \times 1000\}$. Dimensionless units are used to ensure that the demonstration is scale-invariant. The target is modelled using the NCV approach outlined in Chapter 2 and a DIF is utilised to track the motion of the target based on a polar-Cartesian transform.

The sensors are initially located at $\{600, 200\}$ and $\{200, 600\}$ and are managed at each time step by the minimax observer algorithm presented in this chapter. A period of 15 observation steps is permitted before the sensor control activity begins, in order to ensure that the target tracks are stabilised. The optimisation problem is solved using enumeration on a grid centered on each sensor's existing position. The tracking performance requirement is set to $\beta = 0.7$ (again using dimensionless units). This threshold was chosen in conjunction with the target and observation noise variances to ensure that the behaviour of the proposed control algorithm is illustrated effectively. More specifically, the threshold was chosen so that the algorithm switches between its multiple-objective phase and its greedy information phase during the simulation. A much lower threshold (i.e. a much more demanding tracking performance requirement) would prevent the algorithm from entering the minimax phase. In contrast a much higher threshold would not not induce the switching effect which is a key property of the algorithm's performance.

Figure 4.6 illustrates the typical trajectories that are induced by using the greedy information-based approach, which simply selects the new sensor position that yields the most mutual information. Of particular note here is the maximum possible sensor movement that is applied at each time step, as the MI measure attempts to drive the sensor toward the target. Note the nature of the sensor trajectory, which is due to the control-space discretisation that is employed.

Figure 4.7 illustrates the trajectories which are yielded by the Multiple-Objective Optimisation Sensor Management (MOOSM) algorithm presented in this chapter. The nature of the trajectories in this case are clearly much more conservative than those induced by the greedy approach, due to the fact that the algorithm accounts for power consumption, which is proportional to the distance moved by the platform. In this case the secondary objective weights (for power and separation respectively) are set at $\{0.5, 0.5\}$.

Figure 4.8 illustrates the results that are observed for the case when the secondary objective weights are set at $\{0.1, 0.9\}$ to demonstrate the algorithms capability to account

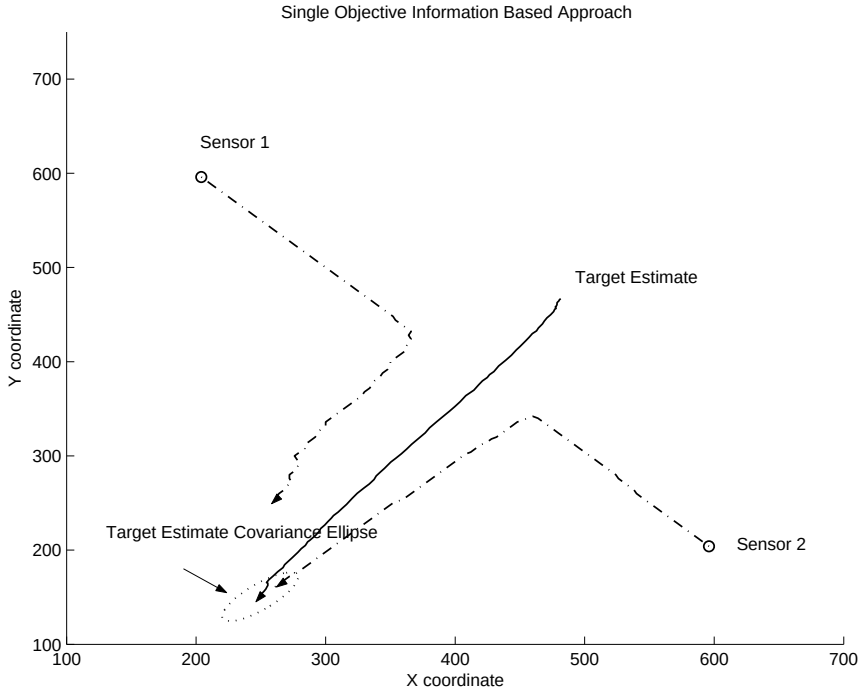

FIGURE 4.6: Example result of greedy information-based observer control.

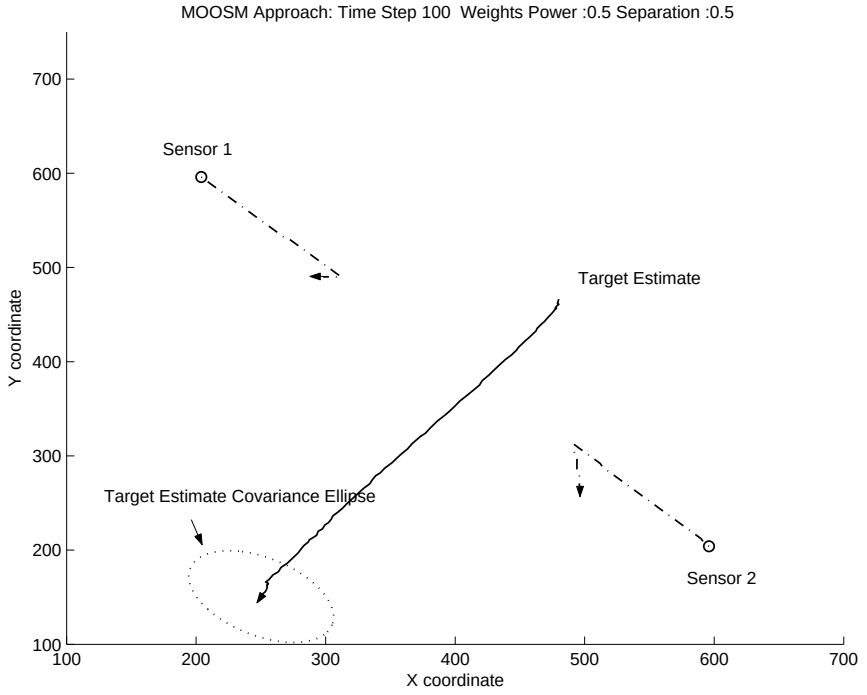

FIGURE 4.7: Example result of multiple-objective observer control with objective weights $\{Power : 0.5, Separation : 0.5\}$.

for the sensor separation objective. In this case the sensors are driven away from each other once the tracking performance measure has been satisfied (hence the algorithm

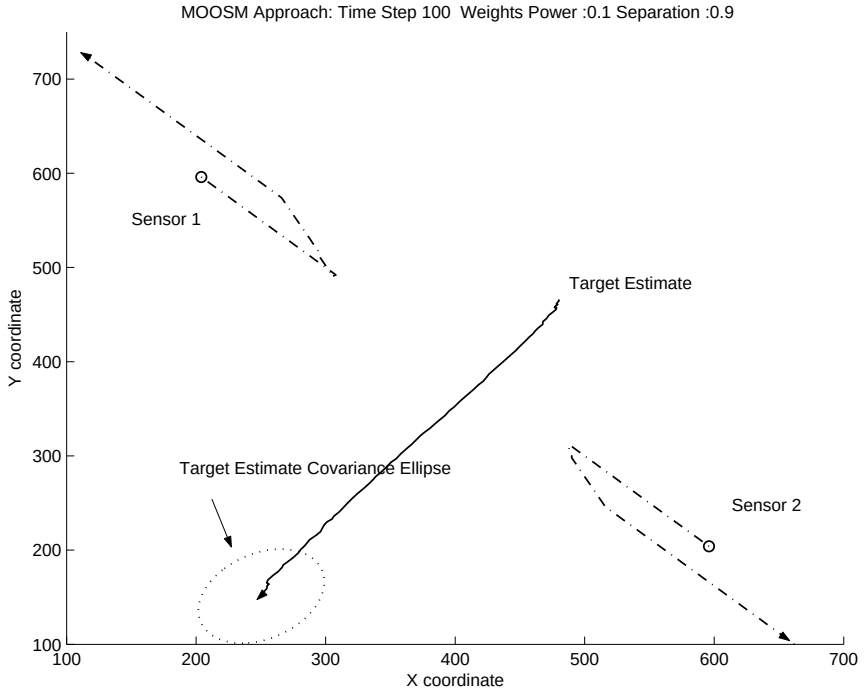

FIGURE 4.8: Example result of multiple-objective observer control with objective weights $\{Power : 0.1, Separation : 0.9\}$.

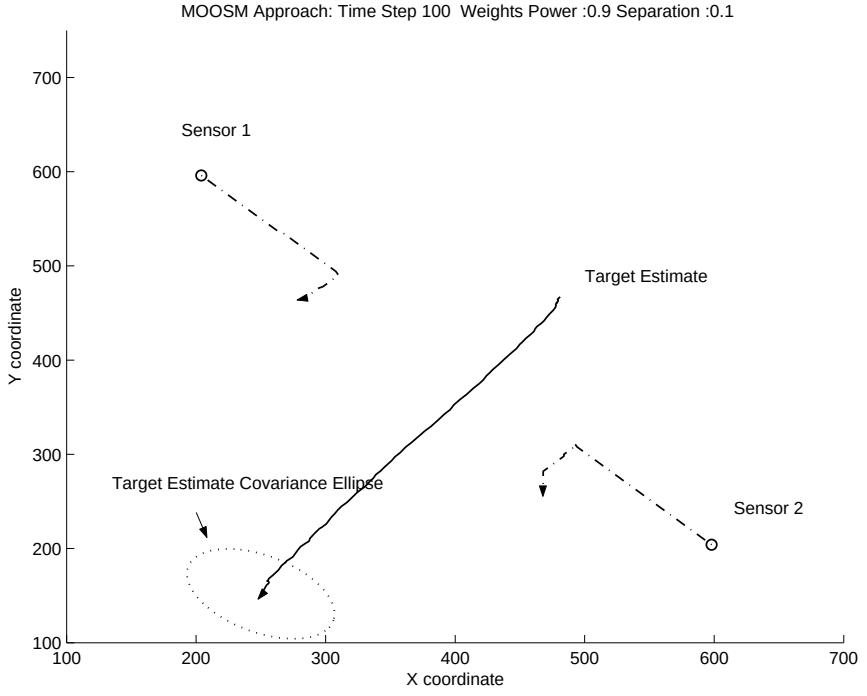

FIGURE 4.9: Example result of multiple-objective observer control with objective weights $\{Power : 0.9, Separation : 0.1\}$.

initially drives the sensors towards the target and subsequently the trajectories reverse direction). Contrasting results arise with the secondary weights set to $\{0.9, 0.1\}$ as

depicted in Figure 4.9. In this case the pressure to separate the sensors is overwhelmed by the weighting of the power objective and thus once the tracking requirement has been met, the sensors expend the least energy possible to maintain it.

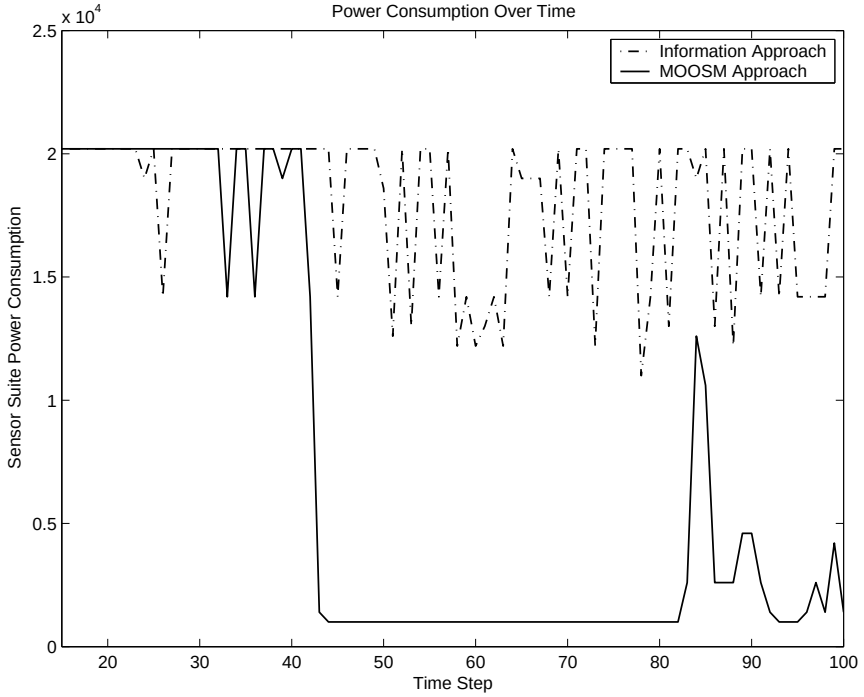

FIGURE 4.10: Comparison of power consumption between greedy approach and multiple-objective observer control approach with objective weights $\{Power : 0.5, Separation : 0.5\}$.

Figures 4.10 and 4.11 compare the power consumption and separation associated with the greedy approach and the multiple-objective approach. It is noted that the performance of both approaches in relation to these objectives is identical until approximately time-step 45. This is due to the constraint relaxation being imposed as the tracking performance is yet to be achieved. Thus the two approaches are equivalent up to this time point. Following time-step 45, when the threshold, β, has been satisfied, the multiple-objective algorithm is free to begin to account for the secondary objectives. Thus the performance with respect to the secondary objectives is significantly increased from this point forwards.

The peak in the minimax approach power consumption at approximately time-step 83 results from the algorithm moving the sensors back towards the target in order to ensure the covariance matrix threshold is satisfied.

The trade-off which is necessary in order to achieve this level of performance is a reduction in primary state estimation performance. This is illustrated in Figures 4.12 and 4.13. The state estimation performance is equivalent until time-step 45, whereupon the multiple-objective approach trades tracking performance for reduced power and increased separation.

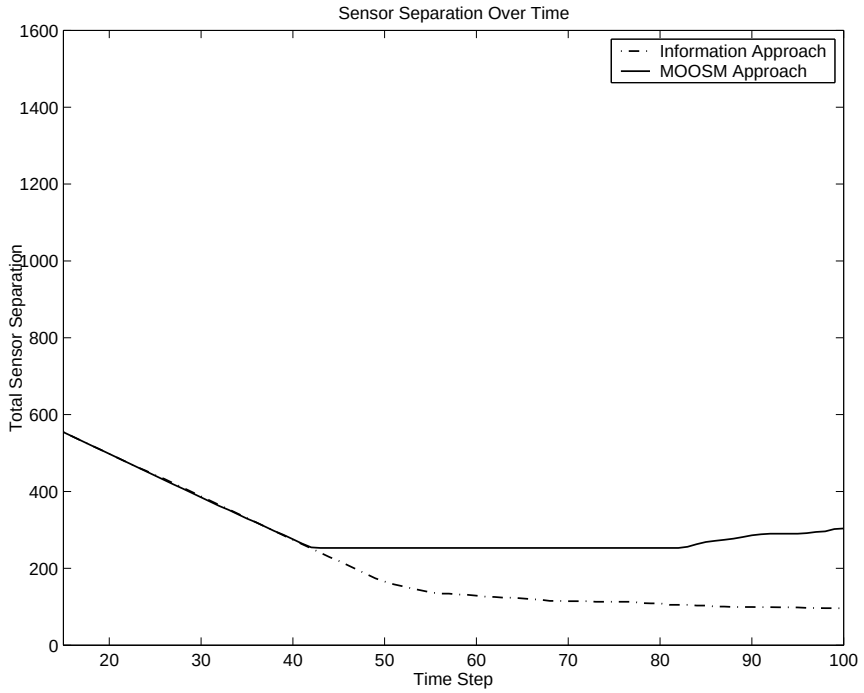

FIGURE 4.11: Comparison of sensor separation between greedy approach and multiple-objective observer control approach with objective weights $\{Power : 0.5, Separation : 0.5\}$.

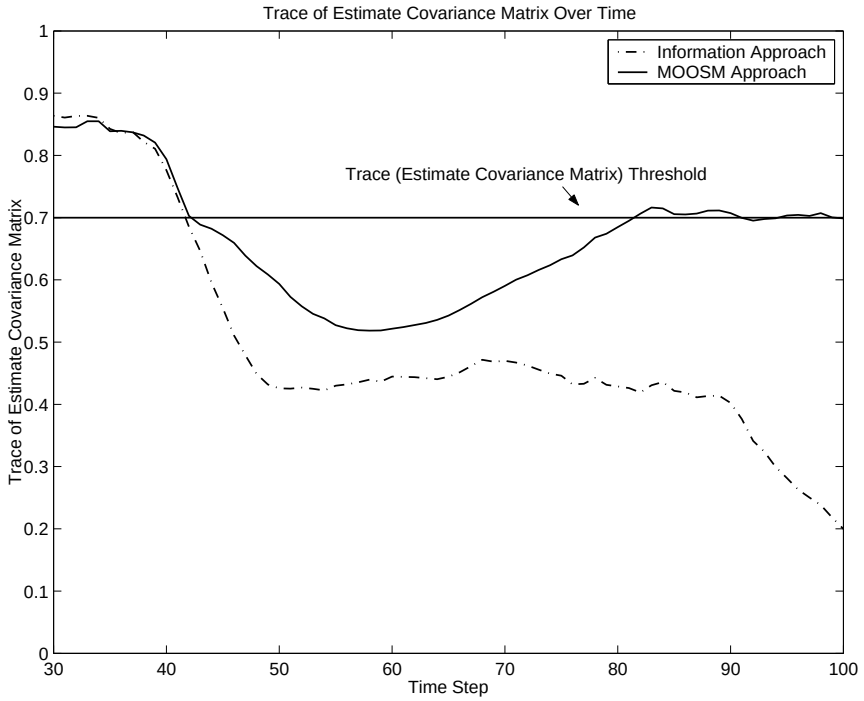

FIGURE 4.12: Comparison of state estimation covariance between greedy approach and multiple-objective observer control approach with objective weights $\{Power : 0.5, Separation : 0.5\}$.

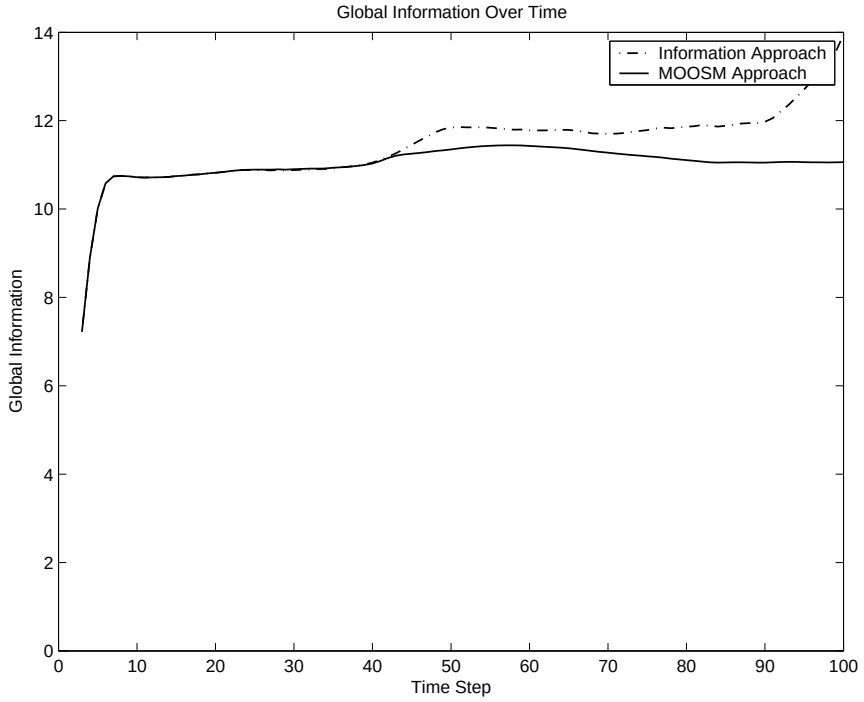

FIGURE 4.13: Comparison of global Entropic information between greedy approach and multiple-objective observer control approach with objective weights $\{Power : 0.5, Separation : 0.5\}$.

A comparison of the total weighted cost for the two approaches is highlighted in Figure 4.14. The multiple-objective approach is capable of achieving a significant reduction in overall cost, whilst satisfying the required state estimation performance levels.

One limitation of the approach is that the myopic nature of the control process can sometimes result in state estimation performance overshoot as illustrated in Figure 4.15. In this example a different target motion and significantly stronger tracking performance requirement is enforced. This highlights the tendency of the controller to hunt around the desired covariance threshold due to the myopic nature of the optimisation process. This may present problems with state estimation frameworks that are not robust or prone to divergence. However, for the state estimation framework utilised in this chapter, the algorithm was observed to offer robust performance in a range of different experiments.

In summary, the results presented above indicate that the minimax observer control algorithm proposed in this chapter is capable of offering superior performance over greedy information approaches, when the assessment accounts for additional objectives. The simulated experiments show that the algorithm is able to maintain predefined tracking performance requirements, and simultaneously minimises excessive resource utilisation in terms of objectives relating to sensor survivability. The result is expected to yield a significant increase in expected sensor lifetime, and thus is of great interest to AV

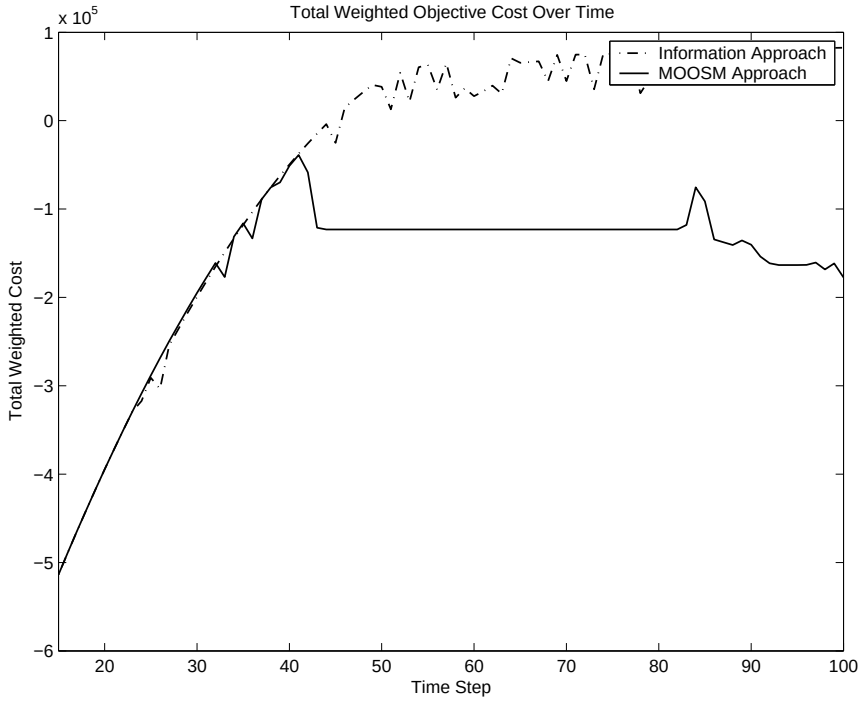

FIGURE 4.14: Comparison of total weighted cost between greedy approach and multiple-objective observer control approach with objective weights $\{Power : 0.5, Separation : 0.5\}$.

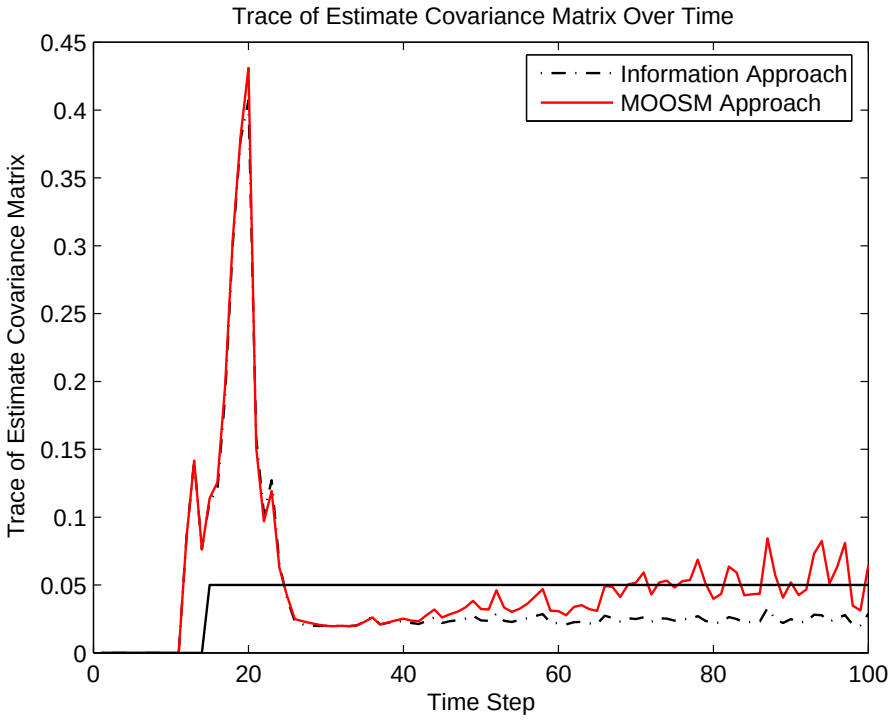

FIGURE 4.15: Overshoot in state estimation covariance induced by multiple-objective observer control.

scenarios where sensors do not have access to unlimited power and may be the subject of hostile attack.

4.8 Non-Myopic Control for Increased Power Efficiency

The state estimation metrics considered in this chapter are indirect functions of the sensor-to-target geometry. For example, it has been shown that the expected information gain for a range-bearing observation model is inversely proportional to the square of range [47] and thus control algorithms based on this measure typically drive the sensor towards the target. The expected information gain for a bearings-only observation model is related to both the range and the bearing between sensor and target [48].

Minimising sensor-target range involves expending power to move the platform. If the target is approaching the sensor platform then it is possible to minimise the expended power by predicting this geometry. Thus a non-myopic control strategy is likely to significantly improve the performance of the algorithm in terms of the power objective. An added benefit which is expected to derive from such an approach is that it may offer additional robustness against the overshoot and hunting characteristics of the algorithm outlined in this chapter. As the non-myopic strategy will consider the long-term ramifications of decisions to move the platform, the controller will mitigate against the overshoot, and provide smoother tracking performance.

4.9 Concluding Remarks

This chapter has presented an analysis of practical aspects of the observer control problem for ground-based AVs. Such platforms may operate in scenarios where there is limited access to power and are typically subject to a number of physical and dynamic constraints. In addition, the platforms may be subject to hostile action and there may be further objectives that the SM must consider in addition to the primary perceptive tasks.

This problem was approached by formulating the observer trajectory control problem in a multiple-objective optimisation framework using objectives relating to power consumption and sensor separation. These objectives are both directly linked to expected sensor suite survivability. A number of objective preference structures were considered, and it was shown that the most transparent approach to the problem is to optimise state estimation performance using a constraint, rather that using goals or weights. Motivated by these conclusions, a minimax observer control algorithm was presented which is capable of maintaining predefined tracking accuracy whilst simultaneously optimising

power and separation related objectives. An example case study was presented which demonstrated the performance of the technique.

In order to avoid some of the limitations of the technique, a non-myopic control strategy was proposed, which is expected to offer superior performance, due to its capability to predict beneficial sensor-to-target geometry and avoid overshoot characteristics.

The following chapter presents a number of contributions relating to such non-myopic techniques.

Chapter 5

Error Propagation in Non-Myopic Sensor Management

'Truth is much too complicated to allow anything but approximations.....'
(John Von Neuman, 1985).

5.1 Introduction

The SM algorithms considered in the previous chapter are *myopic* in nature, in that the optimisation of sensor control inputs is performed based on single-step (discrete) predictions of future target states. Non-myopic sensor control strategies compute optimal sequences of sensor inputs using multiple-step predictions of future target states, in the effort to hedge against the long term ramifications of the control inputs. These sensor control strategies have been demonstrated to provide increased SM performance in a variety of fusion scenarios, including target search [2], [64], [48], target tracking [66] and target classification [64].

A non-myopic decision tree is illustrated for clarity in Figure 5.1. The diagram shows the possible paths that can be taken over 3 discrete planning steps. It is trivial to generate a number of values for the sensor actions a_1 and a_2 at each time step, such that the greedy solution is inconsistent with the optimal non-myopic solution. Consider the simplest case, where a_1 for the first time step has expected utility 10, and a_2 for the first time step has expected utility 5. The greedy solution is therefore a_1. However, in the case where all the expected utility values for the path leading from a_1 are zero, and those corresponding to a_2 are greater than 5, then it is clear that in the longer term, more utility is obtained by considering the larger problem.

The increase in performance relative to myopic or greedy approaches arises from the ability to *predict* and *mitigate* against the effect of changes in the problem domain related

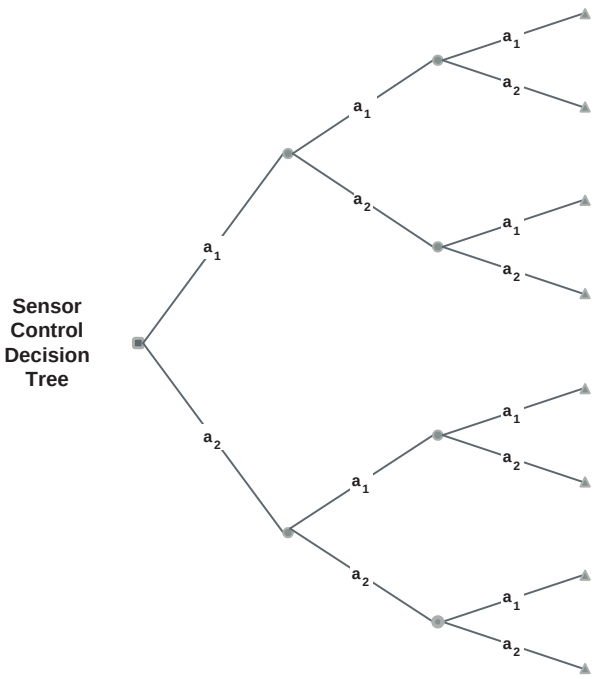

FIGURE 5.1: Sensor Management decision tree.

to mechanisms such as line-of-sight obscuration [63], constrained sensor dynamics, and intersecting targets [64]. These type of scenarios lead to problems which are amenable to the *backwards induction* approach related to Dynamic Programming (DP), where the optimal sequence of control inputs is, in general, not consistent with the greedy solution. The non-myopic SM problem is, therefore, simply a special case of the more generic sequential decision making problem in the presence of uncertainty. Due to the extreme computational requirements of solving multiple-step planning problems, they cannot be solved exactly except for in a limited number of cases (for example when the the total number of steps in the problem is very small). Typically, approximations are therefore employed to reduce the problem to one that is tractable. The approximations usually involve simplifying the problem (e.g. by solving a subset of the problem by using a shorter time horizon) or simplifying the solution, for instance by using a control parameterisation [48]. The focus of this chapter, is on limited-lookahead solutions, i.e. those solutions which solve a sub-set of the larger multiple-step problem.

The further into the future that a non-myopic strategy investigates, the more structure within the decision process is revealed, yielding (potentially) higher planning performance. Some authors have pointed out that whilst it is easy to conjecture that increasing the planning horizon leads to superior performance, there are cases where this may not necessarily be the case [2]. In many studies, this issue is simply overlooked and no analysis of the optimality of the planning horizon is conducted. To date, there has been little investigation into the relationship between the planning horizon length and the resulting performance for typical SM problems (or indeed to the author's knowledge,

for any application). This is probably due, at least in part, to the fact that solving strategies over more than a few steps is often impossible, and thus the performance of such strategies is seen as somewhat academic. To date, there has been no significant study of the choice of the horizon for SM problems.

However, there are a number of important factors which justify research into these mechanisms. Firstly, computational facilities are likely to continue to advance and, therefore, the ability to solve increasingly more complex problems of this nature will also evolve, bringing the effects of the lookahead horizon into play. Secondly, even the effect of small changes in the horizon (e.g. from 1 to 2 steps) is not yet well understood, and any insight into the underlying performance mechanisms would prove useful for those considering the trade-offs between planning performance and computational cost. Lastly, some authors have recently proposed the use of approximate strategies which capture the overall properties of full non-myopic strategies, but at significantly reduced computational cost. These strategies are capable of pushing larger lookahead horizon boundaries even with current computational capability, and it is important to understand their performance. In addition, such strategies introduce extra approximations into the control feedback-loop and are, therefore, particularly relevant to the analysis in this chapter (it will be shown later in the chapter that sources of approximation are the key drivers in understanding the effect of the optimisation horizon).

The following objectives therefore underpin the research presented in this chapter:

- the identification of the sources of error in the sensor control feedback-loop;
- the analysis of the propagation of error in non-myopic sensor control strategies and the effect of this error on sensor control performance;
- the identification of the fundamental mechanisms that explain the limits of the performance increases associated with increasing horizon length in limited-lookahead control strategies;
- and, the provision of a means to infer a suitable horizon length for a given scenario.

The above objectives are wide in scope and are important in a range of SM scenarios[1]. The analysis presented in this chapter is focussed on the trajectory optimisation problem outlined in Chapter 4.

The primary contributions in this chapter lie in the exposition of the error propagation in the sensor control feedback-loop, and the demonstration, for the first time, of one of the underlying mechanisms that limits the performance of limited-lookahead control for SM. This mechanism is due to the relationship between the size of the limited-lookahead horizon, and the inefficiency of any of the control sub-systems. It is noted that, to the author's knowledge, this concept was first identified in the open literature in [75].

[1] These objectives are also applicable to other branches of applied decision theory.

5.2 Chapter Outline

This chapter is organised as follows. Section 5.3 discusses a number of motivating SM scenarios, and identifies the primary reasons why they are susceptible to the backwards induction approach. Section 5.4 discusses the formal definition of the optimal sensor control problem as a multiple-step decision making problem with imperfect state information. Section 5.5 analyses the performance of various system state prediction techniques, based on the material presented in earlier chapters. A similar analysis of utility estimation methods and their associated performance is provided in Section 5.6. Optimal and sub-optimal control strategies are discussed in Section 5.7.

Based on notions of prediction and planning inefficiency, Section 5.8 presents an example tracking problem which illustrates how planning performance can depend on the propagation of error through the SM feedback-loop. To the author's knowledge, this is the first time that the explicit relationship between prediction inefficiency and the optimality of the planning prediction horizon has been demonstrated in practice.

Section 5.9 analyses the effect of the planning horizon on system performance at a generic level, by considering the propagation of various sources of error through the non-myopic control loop. Novel notions of Pre-Emptive Gain and Pre-Emptive Risk are offered, based on representing expected changes in utility associated with changes in horizon length. In Section 5.10 it is shown that in some architectures, system performance is a function of the planning horizon length. In this case, the horizon length should be specified in an appropriate fashion. A new Non-Myopic Risk Equilibrium is proposed which provides a basis for identifying the appropriate horizon length in different scenarios. It is asserted that an adaptive-horizon control strategy which dynamically changes the planning horizon may lead to superior performance with respect to fixed horizon approaches.

Concluding remarks can be found in Section 5.11. Finally, parallels in industrial control and economic portfolio optimisation are drawn in order to expose the wide-reaching application of the adaptive horizon approach.

5.3 Motivating Scenarios

Several important factors underly the success of such non-myopic strategies. In all of the above scenarios, multiple-step planning leads to improved choice of action due to the ability to *predict* and *mitigate against* limited utility in future steps. Without these mechanisms, non-myopic planning will not yield any performance gain over myopic strategies. This is the fundamental basis of any predictive control scheme. The following definitions are provided for the purposes of this research:

Definition 5.1. *Horizon Sensitive Control Problem* A non-myopic sensor control problem is horizon sensitive if the expected performance of the controller is *dependent* on the length of the planning horizon.

Definition 5.2. *Horizon Insensitive Control Problem* A non-myopic sensor control problem is horizon insensitive if the expected performance of the controller is *independent* of the length of the planning horizon.

It is useful at this point to consider a (non-exhaustive) list of utility-limiting mechanisms for typical fusion scenarios, to contextualise the work presented in this chapter. Broadly, they can be classified into two categories, those that *obstruct* an observation entirely (such that no observation can be made) and those that *degrade* the quality of the observation (for instance limiting the information that can be gained).

- Observation obstruction
  - Line-of-sight obstruction
  - Intersecting targets
  - Sensor failure
- Observation quality degradation
  - Maximum velocity
  - Maximum acceleration
  - Terrain constraints

These mechanisms can operate over a period of time in many scenarios, and thus it is not unreasonable to conjecture that, predicting further and further into the future will reveal multiple utility-limiting events, which can then be avoided if possible.

5.4 Non-Myopic Control System Structure

The non-myopic sensor control problem has been addressed from a number of different theoretical frameworks. A popular approach is to model the control problem as a Markov Decision Process (MDP) [45]. As the state of the process under observation is unknown, it must first be formulated as a Partially-Observed Markov Decision Process (POMDP). Fortunately, it is possible to reformulate a POMDP as a completely observed MDP by representing the state by a sufficient statistic [76], for instance the full posterior PDF. It is noted that in special cases, the posterior PDF can itself be represented by sufficient statistics, such as the mean and covariance generated by the update stage of the KF. The optimal sensor control problem can also be conveniently formulated using a MPC

framework as in Chapter 4 and [75], although this is not as common in the literature. MPC is a label used for a number of control techniques that became popular in the control community mainly due to the capacity to handle constraints and the ability to extend the theory to the multiple-variable case [77], [78]. MPC has been extensively used in robotics, the chemical industry and in medical applications using a variety of process models including state-space models (which are common in the target tracking community) [77]. Various other frameworks have been used, and most can be related back to concepts of Approximate Dynamic Programming (ADP), or sub-optimal control.

As described earlier, non-myopic SM has the potential to outperform myopic management as it reveals otherwise hidden structure within the control subspace. Unfortunately, the computational complexity of such problems can increase exponentially as the length of the horizon increases, and soon become intractable, often well before the planning horizon approaches the total number of decision steps in the problem. Typically, therefore, one has to resort to a sub-optimal control strategy such as limited-lookahead, roll-out, certainty equivalence control or Open-Loop Feedback Control (OLFC) [65] ,[76]. The links between MPC and ADP are explored in [76].

A generic sensor control architecture, which is consistent with all the theoretical approaches introduced above, can be summarised using a set of key elements: 1) multiple-step predictions of the system state over a given prediction horizon; 2) a set of performance objective functions relating to estimation performance and sensor action costs; 3) a set of estimation performance and sensor action constraints; 4) an optimisation procedure performed over the prediction horizon leading to a sequence of optimal sensor actions; and, 5) actuation of the optimal sequence (or subset of the sequence) over a predefined control horizon.

5.4.1 Formal Definition of Control Process

The analysis that follows in the remainder of this chapter assumes that the sensor control problem has been first formulated as a decision problem with imperfect information, and subsequently converted to a decision problem with perfect information (in the information state), according to the exposition in [65]. The corresponding set of elements which are required to operate under the resulting framework are:

- a set of possible states for the target;
- a set of possible states for the observer;
- a set of possible sensor control inputs or sensor actions;
- a probabilistic target model which maps the current target state to the future target state;

- a probabilistic model which maps the current target and observer state and the selected control input to the future observer state;
- a utility function which measures the value of the observer state;
- a control law or optimisation which selects the best sensor control input to be applied at each time step;
- and, a probabilistic observer state model which maps the current target state to the future target state.

These elements form the basis of the analysis of the various control sub-systems presented in the remainder of this chapter. In order to establish the scope of the analysis appropriately, only the discrete planning case is considered and it is assumed that the observer state is deterministic (i.e. there is no uncertainty with regard to the observer's state at any point).

5.5 Multiple-Step Prediction Methods

The performance measures discussed in Chapter 3 provide a mechanism for assessing the utility of different sensor control inputs. The metrics are essentially functions of the possible observer state resulting from the control input and the future target state. In the formulation in use here, the sensor dynamics are deterministic, but the future target state is a stochastic variable based on multiple-step prediction.

The question therefore arises as to how to estimate the multiple-step future target state distribution, $p(\hat{X}^t_{k+1:n}|p(\hat{X}^t_k))$, which is defined from the current time k to future time step $k+n$, conditioned on the current estimate. This is computed by repeated application of the target evolution equation:

$$p(\hat{X}^t_{k+n}|p(\hat{X}^t_k)) = p(f^{t^n}(\hat{X}^t_k|p(\hat{X}^t_k))), \tag{5.1}$$

where, with a slight abuse of notation, f^{t^n}, is taken to imply the recursive application of $f^t(\cdot)$, n times. In the simplest case, when the target evolution function is linear, and the associated noise process is white, zero-mean Gaussian, the future target distribution can be calculated analytically. Given the current posterior state PDF estimate, $p(X^t_{k|k}) = \mathcal{N}(\hat{X}^t_{k|k}, P_{k|k})$, the future target state distribution is given by $p(X^t_{k+n|k}) = N(\hat{X}^t_{k+n|k}, P_{k+n|k})$ where:

$$X^t_{k+n|k} = A^n X^t_{k|k} \tag{5.2}$$

$$P_{k+n|k} = A^n P_{k|k} {A^T}^n + \sum_{m=0}^{m=n-1} A^m Q (A^T)^m \tag{5.3}$$

In the above equation, A refers to the target process evolution matrix A^t - the superscript is omitted here for clarity. This is, of course, simply the repeated application of the KF update stage. It is interesting to note that despite the simplicity of these equations, they were not found in the literature at the time of writing. This gives some indication as to the relatively little amount of research that has been carried out into multiple-step SM planning strategies.

In the general non-linear case, the future target state distributions cannot be computed analytically. They can, however, be approximated by various methods including the use of an EKF style linearisation or an Unscented Transform approach. The EKF prediction process is, of course, identical to that derived above for the linear case, with the exception that the matrix, A, is replaced with a linearisation of the current target state.

The UKF based multiple-step prediction process proceeds in a similar fashion using the formulation outlined in Section 2.3.3.2. The *full* PDF can, of course, be estimated empirically using MC simulation. It is noted that prediction of future target state is a more complicated task for manoeuvering targets.

5.5.1 Performance of Multiple-Step Target State Prediction

The quality of the prediction (among other factors) determines the quality of the sensor action assessment and thus the identification of the optimal sensor action. It is, therefore, crucially important to analyse the performance of the prediction process. The prediction depends on the prior, which is obtained from the fusion process (i.e. the tracking filter in use) or, in the first step of a control problem, on the target state initialisation. Ultimately, the quality of the future target state estimate will, therefore, depend on the quality of the initialisation as well as the performance of the prediction process.

It is important to note the prediction process may not be identical to that used within the filtering process (e.g. a particle filter may be used to track the target, and an EKF process may be used for the multiple-step prediction).

This section considers the performance of the prediction process. The effect of the feedback between the output of the prediction process and the corresponding prior at the following prediction step is dealt with in Section 5.9 when considering the overall propagation of uncertainty within the non-myopic control structure.

5.5.1.1 Prediction Efficiency

An important property of any algorithm that is used as the basis for predictive planning is *consistency*. As described in Section 2.4.4, this ensures that the covariance of the true state prediction error is less than or equal to the apparent filter error. If the filter is

inconsistent then the value of $\hat{P}_{k|k}$ will also be underestimated[2], and thus the resulting target state prediction, $\hat{P}_{k+n|k}$ will be inconsistent, even if the prediction process itself is not[3]:

$$\hat{P}_{k+n|k} - E_{\mathcal{X},\mathcal{Z}}\{(X^t_{k+n} - \hat{X}^t_{k+n|k})(X^t_{k+n} - \hat{X}^t_{k+n|k})^T\} \geq 0. \tag{5.4}$$

If the prior estimate is consistent, and the prediction process is consistent, then the resulting predicted estimates are also consistent. However, knowing that the estimator is consistent does not guarantee that the estimate is of any use. As pointed out in Chapter 2, the value $\hat{P}_{k|k}$ may be greatly in excess of the actual covariance. In terms of the multiple-step prediction process, the following quantity should be minimised:

$$\hat{P}_{k+n|k} - E_{\mathcal{X},\mathcal{Z}}\{(X^t_{k+n} - \hat{X}^t_{k+n|k})(X^t_{k+n} - \hat{X}^t_{k+n|k})^T\}, \tag{5.5}$$

that is to say, the multiple-step prediction should be *efficient*[4]. The following simple derivation shows that prediction inefficiency is a function of n. Let $\hat{P}_{k|k}$ be the current apparent updated covariance generated by an inefficient KF, and $P_{k|k}$ be the genuine filter MSE, such that c is positive;

$$\hat{P}_{k|k} - P_{k|k} = c, \tag{5.6}$$

$$c \geq 0. \tag{5.7}$$

The derivation of prediction inefficiency then proceeds as follows:

$$\hat{P}_{k+1|k} - P_{k+1|k} = [A\hat{P}_{k|k}A^T + Q^t] - [AP_{k|k}A^T + Q^t], \tag{5.8}$$

$$= A\hat{P}_{k|k}A^T - AP_{k|k}A^T, \tag{5.9}$$

$$= A[\hat{P}_{k|k}A^T - P_{k|k}A^T], \tag{5.10}$$

$$= A[\hat{P}_{k|k} - P_{k|k}]A^T, \tag{5.11}$$

$$= AcA^T, \tag{5.12}$$

thus it follows directly that prediction inefficiency increases directly with the horizon length. This can appear to be a rather academic result of the extension of estimator performance theory, yet will be shown to be of fundamental importance to the performance of non-myopic control strategies in SM problems. Similar to (5.2), the above result also appears to be missing from the literature, and the author has yet to come across it the open domain.

A simple tracking example is offered to provide extra clarity to this discussion. Consider a basic KF problem, where there is some level of model mismatch, in that the system

[2]In practice, if the filter is inconsistent then it can place too much emphasis on the information and possibly diverge. This phenomenon was part of the motivation for the development of the UKF, see [25].

[3]This is not that unlikely - for example, if the target is linear but the observation process is highly non-linear.

[4]Technically it is possible to have a prediction process which is efficient with respect to the covariance, but inefficient with respect to higher order moments.

noise covariance matrix is incorrectly tuned. By running multiple MC simulations of the problem (10,000 in this case), it is possible to compare the apparent estimator covariance with the genuine filter error associated with the ground-truth. These simulations include randomly generating the target state, and the observation data, such that the error expectation can be taken over both $\mathcal{X}$ and $\mathcal{Z}$.

Figures 5.2 to 5.7 illustrate various aspects of this example estimation problem, in cases where the filter is both efficient and inefficient. The following parameter values were used as ground-truth: $R = 0.02$, $Q = 0.2$, $A = 1$, $H = 1$, $X_0^t \approx -0.4$. The choice of ground-truth parameters is arbitrary; the important aspect of the example is the difference between the assumed filter parameters and the ground-truth parameters, which leads to filter inefficiency. The filter was supplied with exact values for these parameters, except that, in the case with model mismatch, $\hat{Q} = 0.7$. Figure 5.2 illustrates the estimation results for this problem.

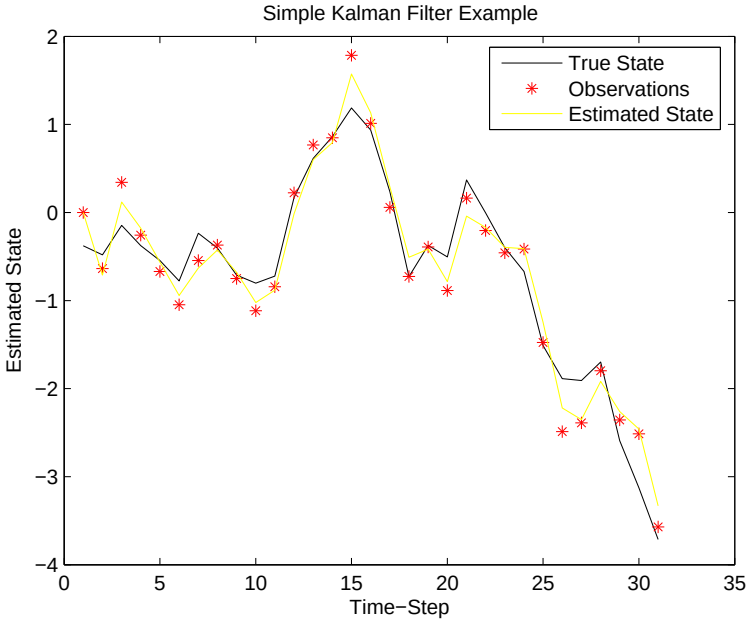

FIGURE 5.2: Example Kalman Filter demonstration.

The apparent filter covariance, ground-truth error and PCRLB for the efficient case can be observed in Figure 5.3. It is clear that all three estimates converge, indicating that any prediction based on this filter output will also be efficient.

In practice, the efficiency of a filtering process can only be assessed by examining various properties of the innovation process, as this is the only route to accounting for real data [1]. The innovation sequence should be zero-mean and white, and in the KF case, the innovation covariance should match that predicted by the filter itself. Figure 5.4 shows the squared normalised innovations for the example problem. Broadly speaking, if 95% of the innovations are observed to fall within a 2σ gate then the filter can be considered

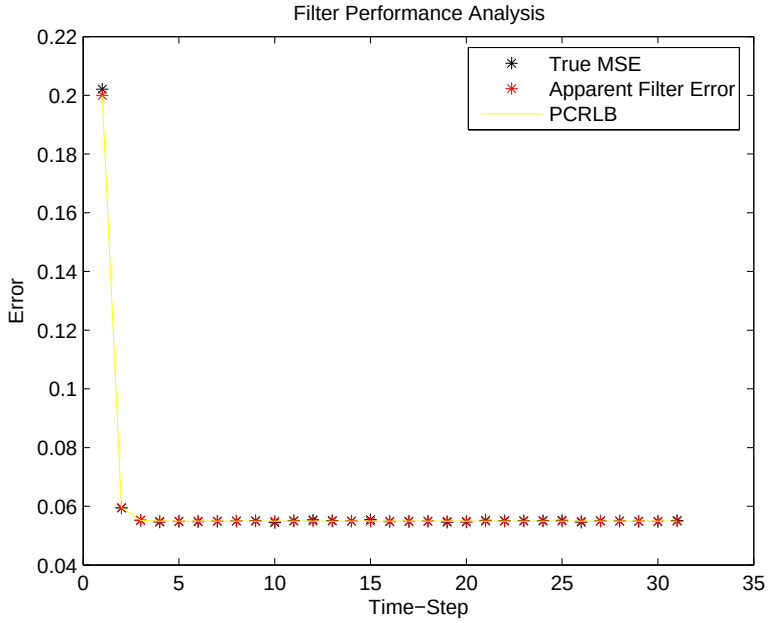

FIGURE 5.3: Efficient apparent filter covariance for estimation problem.

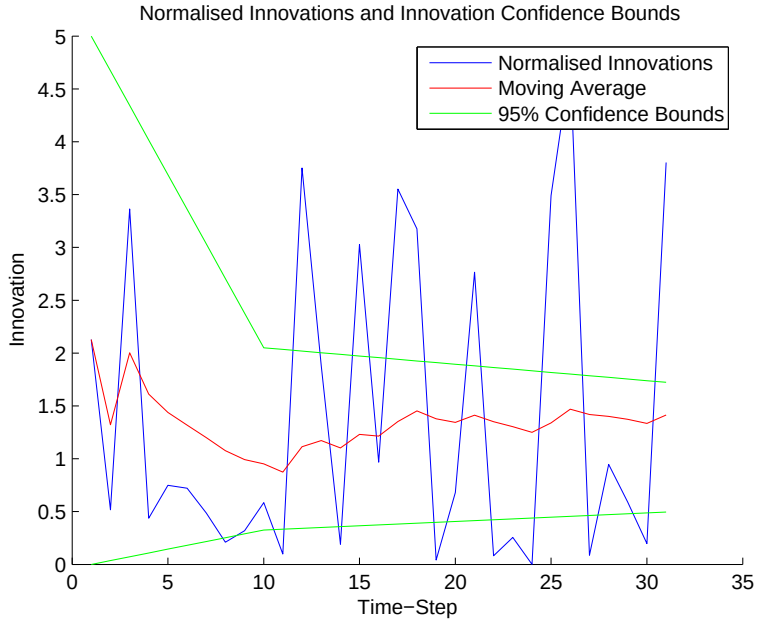

FIGURE 5.4: Normalised innovations for example efficient estimation problem.

to be performing efficiently. The innovation sequence should be white, and therefore ergodic, so the autocorrelation of the innovation sequence can also be used to analyse filter performance. If the innovation exhibits non-random periodic variations then it can be concluded that there are unmodelled dynamics in the target process or the filter is not tuned properly in relation to the observation and process noise. This subsequently

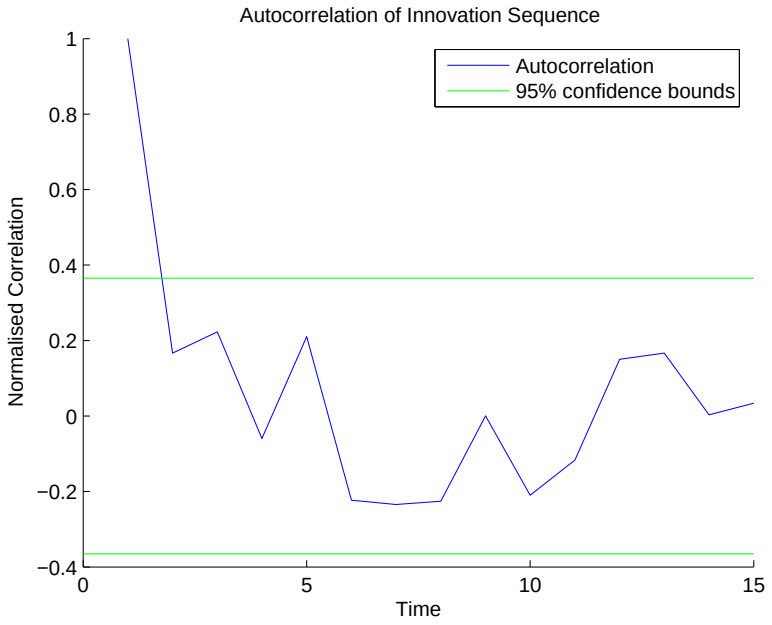

FIGURE 5.5: Autocorrelation for example efficient estimation problem.

should lead to the conclusion that the filter may not be performing efficiently and that the planning process is now operating on sub-optimal priors. Figure 5.5 illustrates that the filter output indicates that it is well matched in the efficient example.

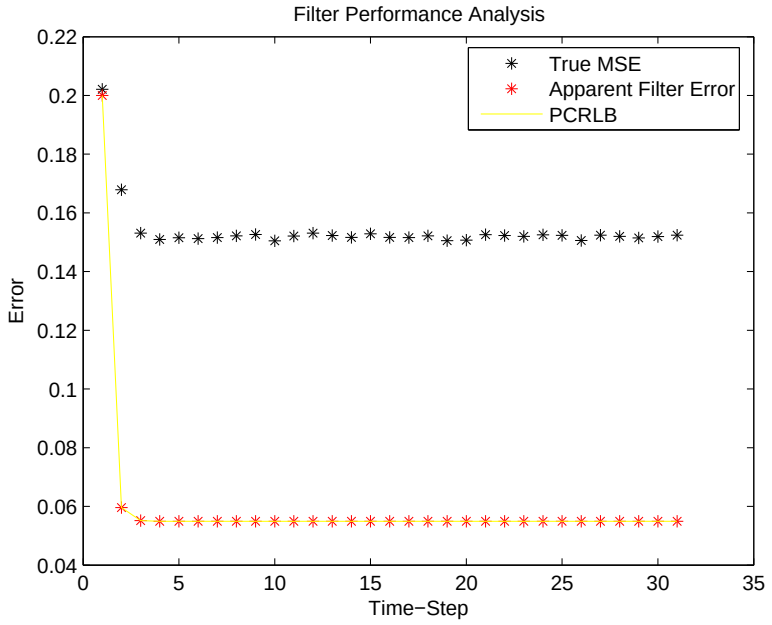

FIGURE 5.6: Inefficient apparent filter covariance for estimation problem.

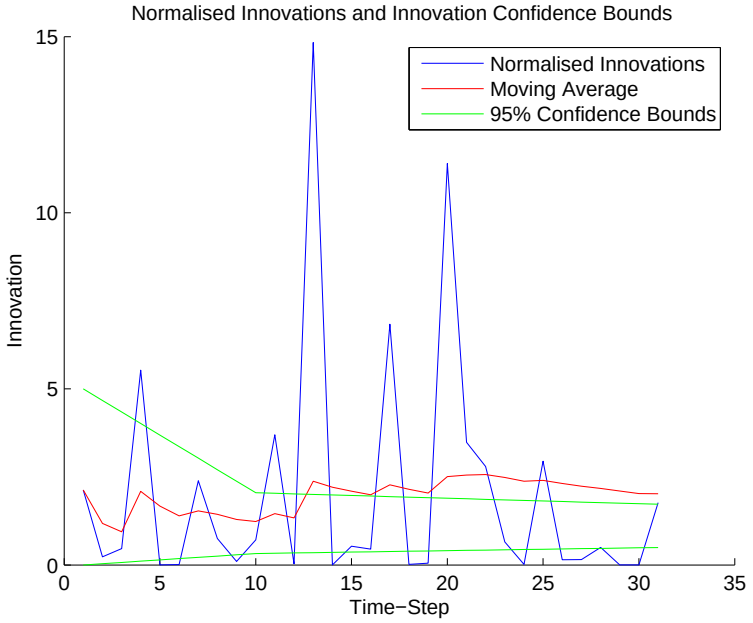

FIGURE 5.7: Normalised innovations for inefficient estimation problem.

Consider now the case where $\hat{Q} = 0.7$. This model error causes the filter to become both inconsistent and inefficient. The apparent covariance (clearly underestimated), true error and PCRLB in this case are depicted in Figure 5.6. In addition, the innovation-based online assessment illustrated in Figure 5.7 now indicates some cause for concern.

5.5.1.2 Prediction Bias

If an accurate utility estimate is to be obtained, the multiple-step prediction process must also be unbiased, i.e.:

$$\hat{X}^t_{k+n|n} \approx E_{\mathcal{X},\mathcal{Z}}\{X^t_{k+n}\}. \tag{5.13}$$

As described in Section 2.3.2.1, the KF is unbiased and thus any predictions that use a Kalman Filter prior will also be unbiased. However, the KF estimate is ensured to be unbiased if, and only if, the initialisation is unbiased:

$$\hat{X}^t_0 \approx E_{\mathcal{X},\mathcal{Z}}\{X^t_0\}. \tag{5.14}$$

If this condition is not satisfied, then the resulting filter estimate will converge to an unbiased estimate over time. During the interim period, however, any predictions based on this prior will be unbiased and thus:

$$\hat{X}^t_{k+n|k} \neq E_{\mathcal{X},\mathcal{Z}}\{X^t_{k+n}\}. \tag{5.15}$$

5.5.2 Summary of Prediction Methods and their Properties

The following generic properties for the prediction algorithms described in Section 5.5 are of key importance:

- The Kalman Filter

  The KF provides an unbiased, efficient prediction method, given that the initialisation prior is unbiased and efficient and there is no model mismatch.

- The Extended Kalman Filter

  The EKF is not generally guaranteed to provide an efficient multiple-step future target state prediction.

- The Unscented Kalman Filter

  The UKF is likely to provide a consistent multiple-step prediction, but it is not guaranteed to be efficient.

- The Particle Filter

  The PF may provide a consistent and efficient multiple-step target state prediction, depending on its structure and the number of particles used.

5.5.3 Intuitive Interpretation of Prediction Uncertainty

The state estimate that the KF provides (and therefore the linear multiple-step prediction it provides) is *exact*, assuming the model is matched. Recall from Chapter 2 that the KF covariance estimate is unconditional, and that the KF mean is conditional. Thus the KF estimate gives the state value that, on average, over all possible state evolutions and observations that could have generated the data, is closest to the ground truth.

5.6 Estimation of Utility

In addition to estimating the future target state distribution, it is also necessary to estimate the utility of various actions. In reality, the estimation of utility may derive from an approximate method, as is the case with the target predictions. Consider for instance the utility associated with bearings-only observation. Many utility function formulations will be dependent on the observation model, which in this case, is non-linear. As a result, there are several approaches to estimating the utility. It may be possible to derive the exact expectation of utility analytically. Sometimes, however, this will not be possible, and an approximate method based on linearisation or stochastic approximation will be employed. This was considered in [66], where both the EKF and UKF formulations are used to assess the utility associated with different actions.

5.6.1 Optimality and Efficiency of Utility Estimates

Utility estimates are, therefore, subject to the same consistency and efficiency analysis that was derived in Section 5.5 for prediction. An illustration of the relative performance of analytic, MC, first-order linearisaton (i.e. EKF) and UKF-type approaches to assessing the polar-Cartesian coordinate transform associated with bearings-only observation, is provided in Figure 5.8.

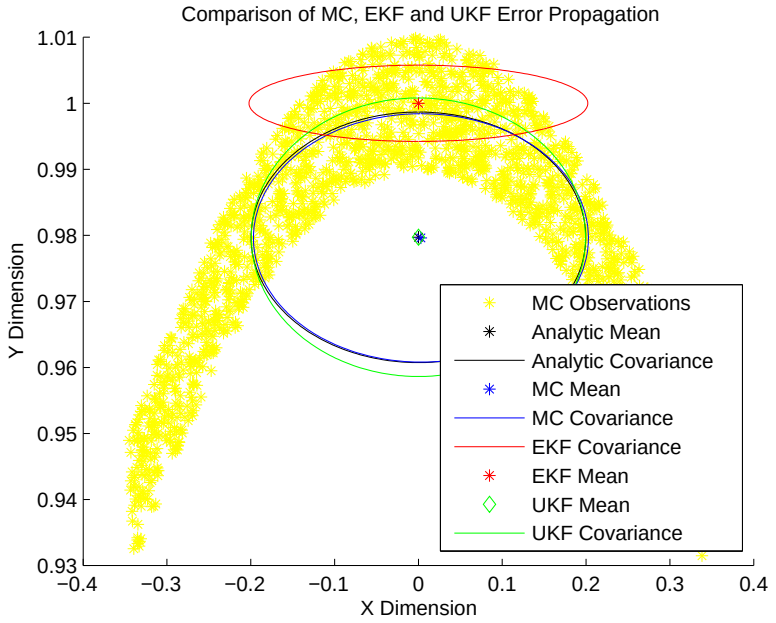

FIGURE 5.8: Comparison between EKF, UKF, and MC estimation of observation covariance.

This example is adopted from the work relating to the development of the UKF [25] ,[26], [27], and demonstrates the inferiority of basic linearisation approaches for the computation of utility associated with non-linear observation models. Ultimately, this highlights a further source of error in the non-myopic control loop. Even if an exact prediction is available, the utility estimate for different actions can still introduce significant error.

5.7 Optimal and Suboptimal Control Strategies

In the ideal case, a full feedback-control strategy with a planning horizon equal to the number of steps in the problem provides the highest level of sensor control performance. Unfortunately, due to the exponentially increasing computational requirements associated with increases in the planning horizon, it is often necessary to resort to various sub-optimal or approximate control strategies. This section describes the different levels of optimality associated with the different control strategies and the relationships between them.

5.7.1 Globally Optimal Closed-Loop Control

The highest level of optimality that could theoretically be achieved, is that which corresponds to the highest utility that can be gained over the entire problem time-line. This could only be achieved if the exact target state ground truth were known *a priori*, and the control action were selected by optimising over the entire problem horizon, $N_p = N_t$ where N_t is the total number of steps in the problem. Of course, if this were the case, the would be no sensor control problem to solve, as the fundamental problem is to obtain perfect knowledge of the target state. However, it is important to define this level of optimality formally, so its relationship with subsequent control strategies can be understood. The following definition is offered for clarity:

Definition 5.3. *Global Planning Optimality* is the level of optimality that can be reached with respect to a given utility function, if exact knowledge of the target state were known *a priori*, i.e. for $k = N_t$. The resulting derived utility function is given by:

$$\mathcal{U}_G(a_{k:N_p}) = E_{\mathcal{Z}}\left\{\mathcal{U}_f(a_{N_p}|X^t_{N_p}) + \sum_{n=k}^{N_p-1}\mathcal{U}(a_n|X^t_n)\right\}, \tag{5.16}$$

subject to

$$N_p = N_t. \tag{5.17}$$

The globally optimal sensor control sequence is then given by:

$$a^\star_G = \arg\max_{a_{k:N_p}} \quad \mathcal{U}_G(a_{k:N_p}) \tag{5.18}$$

where $\mathcal{U}_f$ is the utility function for the final step in the problem.

Another fundamental concept is the notion of Global Feedback Control Optimality. This concept takes account for the stochastic nature of the target state and defines the maximum level of optimality that can be achieved without the ground truth, and thus establishes the upper bound for any real controller. This level of optimality can only ever be achieved under the following (highly unlikely) conditions:

- an exact, complete PDF of the current target state is available;
- an exact prediction algorithm is available;
- an exact utility estimation algorithm is available;
- and, an exact optimisation process is available.

The following definition is offered for clarity:

Definition 5.4. *Global Feedback Control Optimality* is the level of optimality that can be reached with respect to a given utility function, if exact knowledge of the current state distribution and evolution of the dependent distributions were known. The true online global planning utility is, therefore, given by:

$$\mathcal{U}_{GF}(a_{k:N_p}) = E_{\mathcal{X},\mathcal{Z}}\left\{\mathcal{U}_f(a_{N_p}|p(\hat{X}^t_{N_p})) + \sum_{n=k}^{N_p-1}\mathcal{U}(a_n|p(\hat{X}^t_n))\right\}, \tag{5.19}$$

subject to:

$$p(\hat{X}^t_i) = p(X^t_i) \quad k \le i \le N_p, \tag{5.20}$$

$$N_p = N_t. \tag{5.21}$$

The true online global optimal sensor control sequence is then given by:

$$a^\star_{GF} = \arg\max_{a_{k:N_p}} \quad \mathcal{U}_{GF}(a_{k:N_p}) \tag{5.22}$$

It is noted that maximum performance is only achieved by applying the first action in the sequence and re-computing the new optimal action sequence with new observation data in the conventional receding-horizon sense.

5.7.2 Limited-Lookahead Control

Limited-lookahead control strategies alleviate the computational burden of the full feedback solutions by optimising over a restricted horizon $N_p < N_t$. In the simplest case, a horizon of length $N_p = 1$ is used, corresponding to the so-called *greedy* or myopic solution.

The associated utility function for the limited lookahead case is given by

$$\mathcal{U}_{LL}(a_{k:N_p}) = E_{\mathcal{X},\mathcal{Z}}\left\{\mathcal{U}_f(a_{N_p}|p(\hat{X}^t_{N_p})) + \sum_{n=k}^{N_p-1}\mathcal{U}(a_n|p(\hat{X}^t_n))\right\}, \tag{5.23}$$

subject to:

$$N_p < N_t. \tag{5.24}$$

The estimated global optimal sensor control sequence is given by:

$$a^\star_{LL} = \arg\max_{a_{k:N_p}} \quad \mathcal{U}_{LL}(a_{k:N_p}). \tag{5.25}$$

This solution would be optimal if, and only if, there were N_p decision steps left in the problem, and the estimated state distribution was exact, and all other processes involved in estimating the utility were perfect. The key difference between this formulation and

the Global Feedback Control Optimality is the limited horizon, and the fact that the condition $p(\hat{X}_i^t) = p(X_i^t)$ may not hold in real cases. If the condition does hold, however, then increasing the lookahead brings the performance of the controller closer to the upper bound defined in the previous section.

5.7.2.1 Roll-Out Algorithms

Roll-out algorithms are special cases of limited-lookahead or restricted-horizon control. Roll-out approximates the final cost-to-go with a base policy (e.g. a certainty equivalence control base policy). It has been shown that the use of roll-out always improves on the base policy itself [65]. The computational complexity of roll-out is intermediate between the greedy solution and the full feedback control solution. More specifically, the extra computational requirements over the greedy solution are proportional to the requirements of the base policy.

5.7.2.2 Receding Horizon Algorithms

Receding horizon algorithms are also special cases of limited-lookahead control, where the cost-to-go is set to zero. The resulting policy is therefore only optimal if the remaining horizon was of length N_p and no terminal cost is present. It has been noted in [2] and [65] that whilst it is tempting to conjecture that increasing the size of the lookahead leads to superior performance, this is not necessarily the case. This conjecture hides subtle assumptions regarding the structure of the decision space.

An example of a case where decreasing the horizon length yields superior performance can be found in [65]. Another example is provided where the optimal horizon length is invariant to the length of the problem (i.e. a Horizon-Insensitive problem).

5.7.3 Open-Loop Feedback Control

Open-loop Feedback Control is based on the assumption that no future measurements will be taken at future time-steps. It has been shown that OLFC performs at least as well as open-loop control [65]. More specifically, OLFC performance is bounded by that of open-loop control. Unfortunately, it is difficult to assess how closely OLFC performance matches the full feedback controller performance. Partial Open-Loop Feedback Control (POLFC) is intermediate between OLFC and the full feedback controller, and assumes that only a subset of the measurements will be taken in the future.

5.8 Example Illustration of Error Propagation

Motivated by the concepts of prediction and utility estimate efficiency proposed in the previous section of this chapter, this section presents an example illustration of the propagation of error through a limited-lookahead SM control loop. The example is based on the extension of the simple tracking problem first outlined in Section 5.5.1.1.

Firstly, multiple-step prediction and utility estimation is demonstrated empirically in both efficient and inefficient cases. Subsequently, a surrogate utility function is derived which embodies a utility-limiting mechanism, so that the integration of the two competing mechanisms that are at the heart of this chapter can be demonstrated. It is shown that when the planning process is inefficient, optimal planning performance is achieved with a finite-horizon length.

5.8.1 Example Multiple-Step Prediction

Using the example tracking scenario outlined in Section 5.5.1.1, a multiple-step target prediction process was implemented using repeated application of the KF update equations. In this case the following settings were used: $F = 1$, $H = 1$, $Q = 5$, $R = 4$, $N_p = 15$. Figure 5.9 illustrates the resulting comparison between the empirical prediction error at each step in the future, and the analytical prediction error based on (5.2), when the filter is efficient (i.e. there is no mismatch in the model). Ten-thousand MC simulations were used to generate the results. This result validates the multiple-step

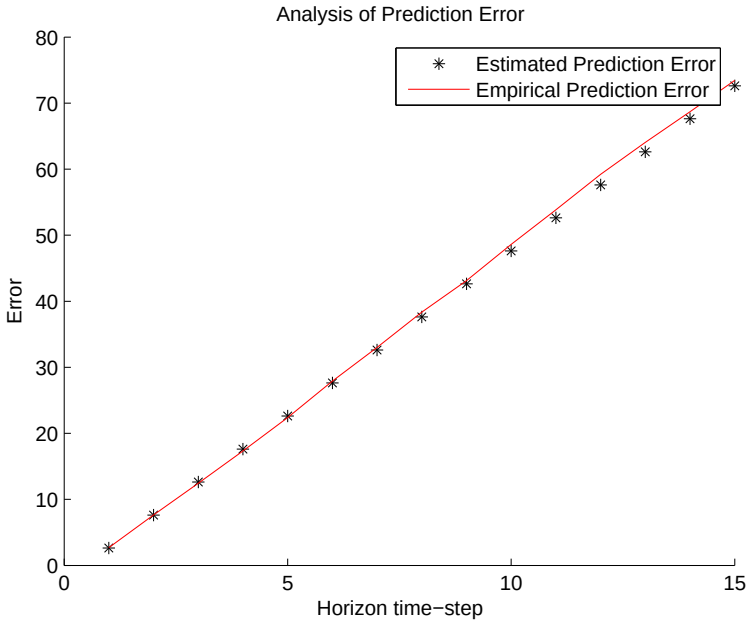

FIGURE 5.9: Multiple-step target state error for efficient filter.

prediction process outlined in this chapter. Figure 5.10 presents the same result, but now it is assumed that there is a significant model mismatch in the KF, meaning that the multiple-step prediction prior is now inefficient. The result is that the apparent prediction error becomes increasingly inefficient as the number of lookaheads increases. It is noted that a large mismatch was used in order to generate results that are easy to

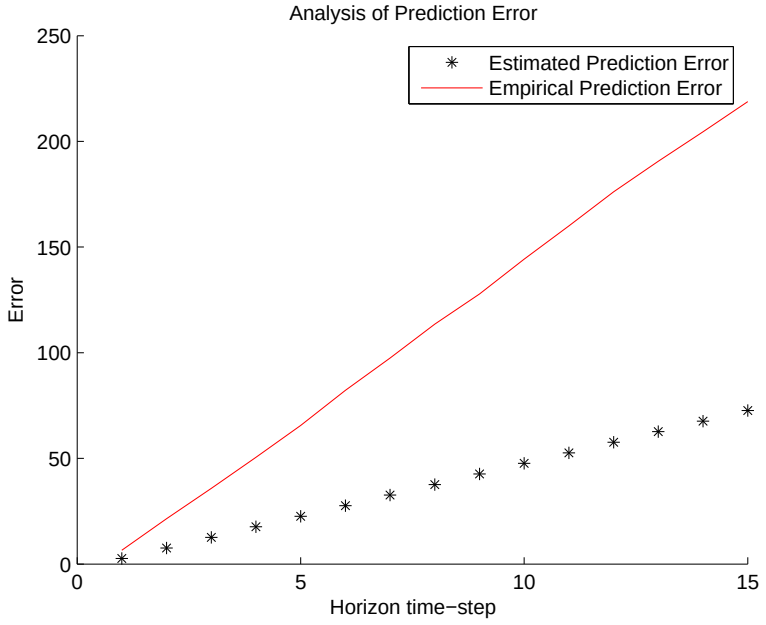

FIGURE 5.10: Multiple-step target state error for inefficient filter.

interpret ($\hat{Q} = 15$ and $\hat{R} = 9$). When a smaller mismatch is used (which is probably more relevant to real-world cases), the inefficiency is less obvious, and may in some cases bear no significant impact on the resulting control process.

5.8.2 Example Multiple-Step Utility Estimation

In the same way that it is possible to analytically determine the multiple-step prediction error in linear cases, it is also possible to determine the utility estimate error analytically for linear cases [75]. The following utility function was implemented to demonstrate this process:

$$\mathcal{U}(X_{k+n}) = \lambda X_k^t + const, \tag{5.26}$$

where λ is a multiplicative factor and *const* is a constant. In this case the utility is only dependant on the sate and not on any decision or action. The resulting distribution for this utility function is given by [75]:

$$p(\mathcal{U}(X_k)) = \mathcal{N}(0, \lambda \hat{P}_{k+n} \lambda). \tag{5.27}$$

The utility distribution is a zero-mean Gaussian with a variance which varies non-linearly with the target state prediction error. Figure 5.11 illustrates the use of the above equation to predict the utility error when the KF is efficient and, $\lambda = 4$, and $const$=10. In constrast, Figure 5.12 presents the results when the KF is inefficient. Now, the utility

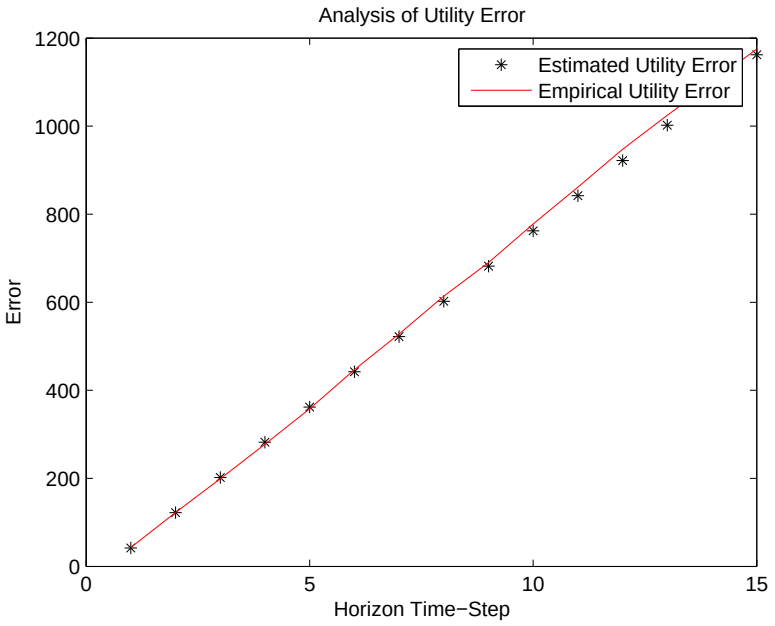

FIGURE 5.11: Multiple-step utility estimate error for efficient filter.

estimation process has become inefficient due to the propagation of error through the control architecture. If the utility function was a function of different actions, there is a danger that the wrong action might be chosen.

5.8.3 Multiple-Step Planning with Surrogate Utility Function

To demonstrate the competing effects of revealing substructure in the decision problem, and the propagation of error, a utility function which captures both mechanisms must be derived. Consider the case where at each time step there are two possible actions, a_1 and a_2. It is believed that action a_1 is always optimal when the target state is less than 100, and results in a utility of 1000. If the state is greater than 100 then action a_2 is optimal. A utility-limiting mechanism is simulated by assuming that if action a_1 is taken and the real state is actually greater than 100, then a loss of 1300 will be incurred. This is analogous to various SM scenarios. For example consider the case where the SM algorithm mistakenly chooses to move a vehicle into a certain location because it believes it will lead to optimal tracking performance. If the target is not actually in the location, the system is then prevented from observing it again until after some time-period. This might be because the exit route from the location takes much longer to navigate than the entry route.

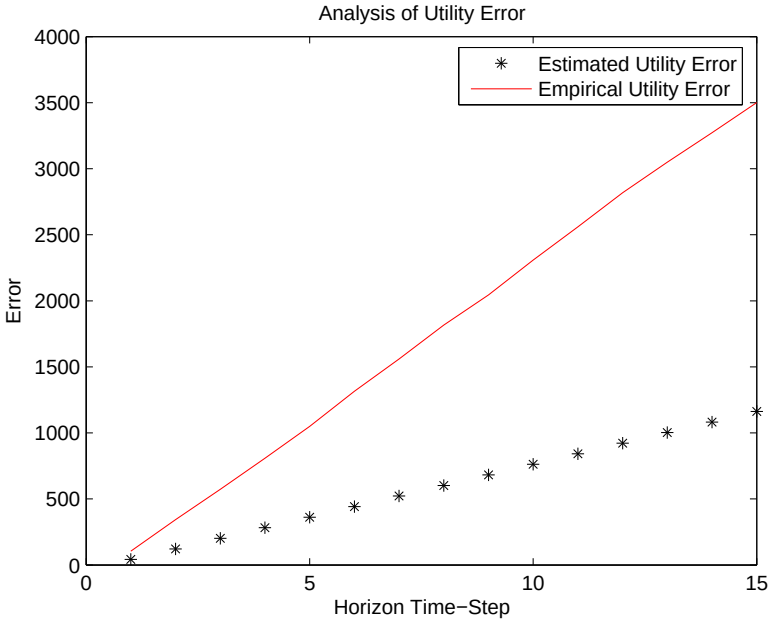

FIGURE 5.12: Multiple-step utility estimate error for inefficient filter.

5.8.4 Impact of Planning Horizon Length

Returning to the inefficient filtering case outlined previously; Figure 5.13 presents the mean estimated future target state (again using 10,000 MC simulations) and the actual future target state. The parameters used in this case were: $F = 1.2$, $\hat{F} = 1.0$, $H = 1$, $Q = 5$, $R = 4$, $\hat{R} = 17$, and $N_p = 15$. Due to the filter inefficiency, the system believes that the target state will remain less than 100 for all 15 time-steps in the horizon. The blue marker indicates the target value where the optimal action switches. To examine the impact of the prediction inefficiency, the cumulative utility estimate is plotted in Figure 5.14 over a range of different optimisation horizons, N_p. Once again, multiple MC simulations are used and the results are averaged over the repetitions. The results in Figure 5.14 clearly demonstrate that the maximum utility is actually achieved when a horizon length of approximately 12 is chosen, despite the fact that the system can predict up-to 15 steps into the future. In contrast, when the filter is efficient, the optimisation horizon can be increased as far as possible, as illustrated in 5.15 (note that different parameters were used in the generation of this figure). In summary, the basic but informative experiment presented above has demonstrated the competition between the two mechanisms which are the focus of this chapter. In the following sections, these mechanisms will be discussed in a more general context.

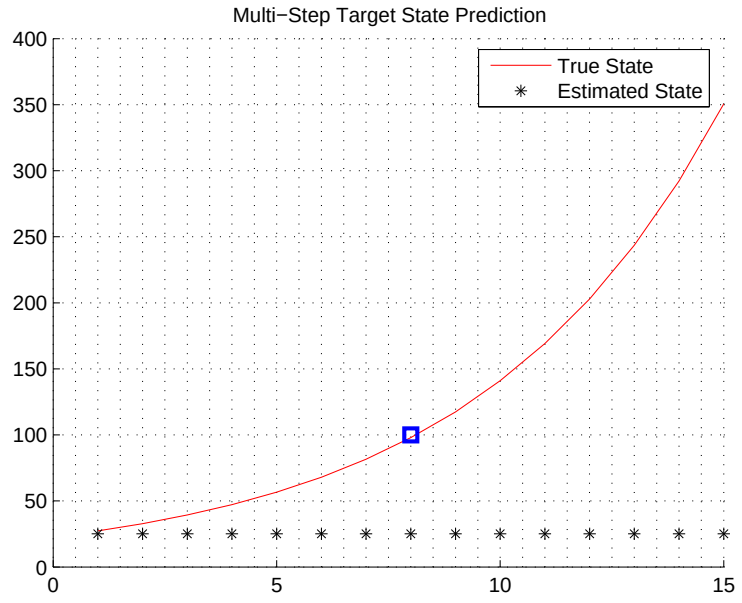

FIGURE 5.13: Multiple-step target state prediction results for inefficient filer.

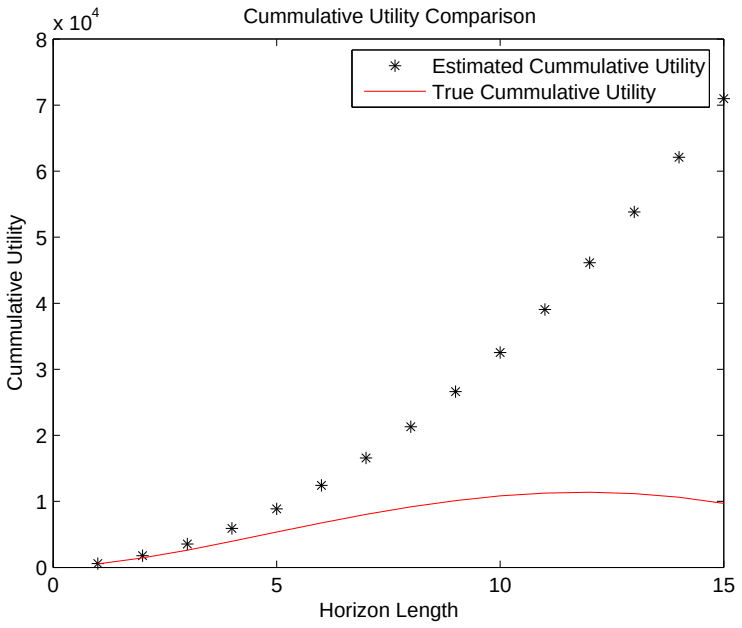

FIGURE 5.14: Cumulative planning utility in inefficient case.

5.9 Quantifying the Effect of the Planning Horizon

This section discusses the two mechanisms which relate the length of the planning horizon to system performance in limited-lookahead horizon control strategies in more depth.

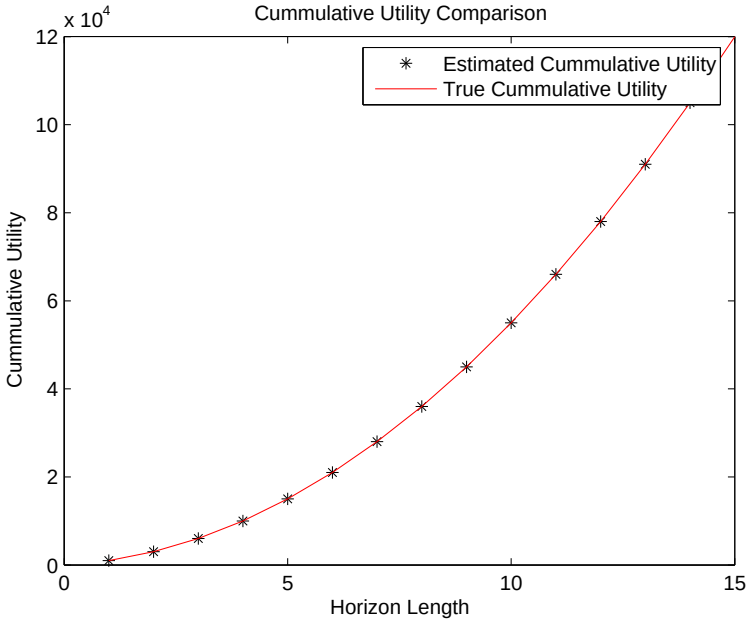

FIGURE 5.15: Cumulative planning utility in efficient case.

These mechanisms will subsequently be used to propose a control strategy that utilises a dynamic optimisation horizon.

5.9.1 The Propagation of Error

Consideration of the propagation of error through the SM feedback loop has so far provided important insight into the performance of limited-lookahead control. The material presented in this section brings together all of the possible sources of error in the sensor control feedback loop. The sources of error and their associated causes are as follows:

1. Deviations from exact target initialisation prior distribution

   Incorrect prior information

2. Deviations from exact target estimate distributions

   Deviations from exact target initialisation prior distribution

   Inexact state estimation method[5]

3. Deviations from exact future target estimate distributions

   Deviations from exact target estimate distributions

   Inexact prediction method

[5]It is noted that DF methods and partial information structures can induce the same effect as modelling errors.

4. Deviations from exact utility estimate distributions
    - Deviations from exact future target estimate distributions
    - Inexact utility estimate
    - Incorrect utility function[6]
5. Deviations from optimal control sequence
    - Deviations from exact utility estimate distributions
    - Incorrect utility function
    - Sub-optimal control optimisation
6. Deviations from exact target state update distribution
    - Deviations from optimal control sequence
    - Inexact state estimation method
    - Sub-optimal control optimisation

This list represents a hierarchy of error sources which might contribute to the propagation of error around the control loop and they are further summarised as follows. Deviations from an exact initialisation prior can lead to bias and inefficiency even in simple linear tracking using the KF. An exact initialisation prior does not guarantee an exact state estimate if the filter in use is not capable of providing one. The multiple-step prediction process is of fundamental importance in the problem formulation. If the estimates of the future target state are inaccurate then the evaluation of different control actions will also be inaccurate. This problem is compounded through the computation of derived utility which may also be inaccurate and thus the optimal control input sequence that is identified may be far from optimal. In addition, the resulting derived utility may be computed as a function of the *estimated* state, and thus the state update can also be incorrect.

5.9.1.1 Error Propagation Elements

The following list encapsulates the error sources, and an additional set of measures that can be derived to model the propagation of error in the non-myopic control loop. These measures include the efficiency of the prediction process (as derived in Section 5.2) and measures of the efficiency of the planning process.

[6] This refers to modelling assumptions.

Description	True Quantity	Apparent Quantity
Initialisation prior distribution	$p(X_0^t)$	$p(\hat{X}_0^t)$
Current target state distribution	$p(X_k^t)$	$p(\hat{X}_k^t)$
Current target state	X_k^t	$\hat{X}_k^t$
Current target estimate error	$\tilde{X}_k^t$	$\hat{\tilde{X}}_k^t$
Current target estimate efficiency	$\varrho(\hat{X}_k^t)$	$\hat{\varrho}(\hat{X}_k^t)$
Future target state distribution	$p(X_{k+n}^t)$	$p(\hat{X}_{k+n\|k}^t)$
Future target estimate bias	$b(\hat{X}_{k+n\|k}^t)$	$\hat{b}(\hat{X}_{k+n\|k}^t)$
Future target estimate efficiency	$\varrho(\hat{X}_{k+n\|k}^t)$	$\hat{\varrho}(\hat{X}_{k+n\|k}^t)$
Utility for action sequence	$\mathcal{U}(\hat{X}^t, a_{k:n})$	$\hat{\mathcal{U}}(\hat{X}^t, a_{k:n})$
Bias of estimate of utility for action sequence	$b(\hat{\mathcal{U}}(\hat{X}^t, a_{k:n}))$	$\hat{b}(\hat{\mathcal{U}}(\hat{X}^t, a_{k:n}))$
Utility for action sequence efficiency	$\varrho(\hat{\mathcal{U}}(\hat{X}^t, a_{k:n}))$	$\hat{\varrho}(\hat{\mathcal{U}}(\hat{X}^t, a_{k:n}))$
Optimal action sequence	$a^\star_{k:k+N_p}$	$\hat{a}^\star_{k:k+N_p}$
Max potential expected utility	$\mathcal{U}_k^\star$	$\hat{\mathcal{U}}_k^\star$
Error in max potential expected utility	$\tilde{\mathcal{U}}_k^\star$	$\hat{\tilde{\mathcal{U}}}_k^\star$
Max potential expected utility efficiency	$\varrho(\tilde{\mathcal{U}}_k^\star)$	$\hat{\varrho}(\hat{\tilde{\mathcal{U}}}_k^\star)$
Derived utility from applied action	$\mathcal{U}_d(X^t, a_k)$	$\hat{\mathcal{U}}_d(\hat{X}^t, a_k)$

Thorough derivation of all of the above measures is the subject of ongoing future work. It is noted that there may be some ground to be made in deriving measures of the innovation of the utility process, much in the same way as the innovations can be used to assess filter efficiency online.

5.9.1.2 Real and Apparent Utility

State updates are based upon estimated quantities for measurement information contribution[7] and thus are incorrect if any error has propagated through the control loop. The *effect* of this phenomenon is dependent on the particular measure. Consider the case where the chosen metric is MI as utilised in the observer control optimisation algorithm presented in Chapter 4. Mutual Information is computed as follows for Gaussian distributions:

$$\mathrm{I}_j = \frac{1}{2}\log\left[\frac{\mathrm{y}_{k|k-1} + \mathrm{I}_j}{\mathrm{y}_{k|k-1}}\right]. \tag{5.28}$$

In cases with non-linear observation processes, the matrix I_j will be a function of estimated variables, depending on whether an EKF linearisation or a polar-Cartesian transform is used. I_j is computed by:

$$\mathrm{I}_j = {H^j}^T R^{-1} H^j, \tag{5.29}$$

[7] irrespective of which performance measure is being used.

where, in the EKF case, the observation matrix H is a function of the estimated target state, and in the polar-Cartesian transform case, R is a function of the estimated state. Once an optimal action sequence has been identified, the action is performed, the observation is made and the following update process is performed:

$$\mathrm{i}_j = H^{j^T} R^{-1} z_k^j \tag{5.30}$$

$$\hat{\mathrm{y}}_{k|k} = \hat{\mathrm{y}}_{k|k-1} + \mathrm{i}_j \tag{5.31}$$

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + \mathrm{I}_j \tag{5.32}$$

The information state vector, i_j, is updated as a function of the observation, the value of which depends on the *actual* distance between the sensor and the target, and the information matrix, $P_{k|k}^{-1}$, is updated according to the *estimated* distance between the sensor and the target. In the case of coordinate transformations, the estimated state error manifests itself through the measurement covariance matrix, R. The state estimate error is, therefore, fed-back into the tracking process at every stage. This is the final stage in the error propagation cycle.

5.9.1.3 Pre-Emptive Planning Risk

The sources of error that are highlighted in the previous sections contribute to a mechanism which is described here as Pre-Emptive Planning Risk. This is the risk that is introduced into the control process by any inefficiency or other source of epistemological error that propagates around the SM loop.

For example, it was shown earlier in this chapter, that prediction inefficiency is a function of the limited-lookahead horizon length, and thus in some cases, the optimality of the control optimisation may decrease as the horizon is increased. This is the fundamental result that is in direct opposition with the natural benefits associated with increased lookahead. Another interpretation of this concept is that, for every increase in lookahead horizon, there may be an associated *decrease* in planning performance due to the effects of unknown error, which are amplified by the horizon length.

A proposed method to model this effect is to compute the expected decrease in performance that increasing the horizon length will induce, under the Global Feedback Control strategy presented in Section 5.7.1, through a surrogate non-myopic risk measure, P_{nmr} relating to the apparent utility efficiency:

$$P_{nmr}(N_p) = \hat{\varrho}(\mathcal{U}_{LL}(a^{\star}_{k:N_p})). \tag{5.33}$$

It is noted that this quantity can be calculated for a range of pairs of lookahead horizons (i.e. the above measure could be considered a special case of a generalised measure of risk which is defined using any pair of horizon lengths). In practice, this is an elusive quantity

and would have to be generated through extensive MC simulation and subsequent utility analysis. However, the proposed measure represents a novel approach to considering the relationship between the performance associated with limited-lookahead strategies which use difference horizon lengths.

5.9.2 Modelling Decision Process Sub-Structure

The increases in planning performance associated with increased lookahead are well established in the decision theory literature. As the horizon is increased, the limited-lookahead strategy converges towards the notion of global optimality defined in Section 5.7.1, as more sub-structure in the decision space is revealed. However, to the author's knowledge, there is no generally accepted formal approach to capturing this mechanism in the open literature.

5.9.2.1 Pre-Emptive Planning Gain

In line with the measure of Pre-Emptive Planning Risk outlined above, a complementary measure of Pre-Emptive Planning *Gain*, P_{nmg}, is proposed below. This measure is related to the expected performance *increases* that are associated with increasing the lookahead horizon:

$$P_{nmg}(N_p) = E\{\mathcal{U}_{LL}(a^{\star}_{k:N_p}) - \mathcal{U}_{LL}(a^{\star}_{k})\}. \tag{5.34}$$

This measure can be computed empirically through multiple MC simulations of the planning problem. This would allow the analysis of the gain associated with increasing the lookahead, without any prior knowledge of the substructure of the decision process, or any utility-limiting mechanisms that it hides. Of course, if this kind of prior knowledge is available, it could be used in the analysis; indeed the creation or identification of an example problem in which the Pre-Emptive Gain and Risk can be modelled analytically (at least partially) is recommended for future research.

5.10 Optimising the Planning Horizon

In the previous sections of this chapter, it was shown that the performance of a control scheme, which is based on sub-optimal representations of the problem-space, is dependent on the length of the optimisation horizon. It was shown that there are two primary competing mechanisms which control this dependence - the exposure of the substructure of the decision space (which permits the identification of action sequences which yield the most utility), and the propagation of risk though the control feedback loop, thus limiting the accuracy of the estimate of the decision-space structure.

Methods for modelling these mechanisms in an empirical fashion have been proposed above, such that their relative contribution to overall performance can be assessed. This leads directly to the notion of an 'optimal horizon length'[8]. This concept relates to a horizon length which is chosen based on knowledge of the two key effects that are related to it. This is in contrast to many approaches which have not accounted for the negative effects that a longer horizon can induce, and instead just pick the longest horizon that provides a tractable problem.

The ultimate goal of this kind of research is to identify algorithms which can compute the optimal horizon length, preferably online. As discussed in [75], a measure of the state estimation efficiency, for example, could be one factor used in such a method. Selecting the optimal horizon length is a non-trivial task for a number of reasons, all of which relate to the difficulties associated with estimating the effect of the horizon. In many cases, the non-myopic gain and non-myopic risk will be time-variant, and thus, rather than a fixed (but optimised) horizon length, a dynamic or adaptive horizon length must be used. Although there are many related concepts in the control literature, and indeed some authors have hinted at such a method [2], to the author's knowledge this concept was explicitly suggested for the first time in [75].

5.10.1 Non-Myopic Risk Equilibrium

The two mechanisms outlined above, Pre-Emptive Planning Risk, and Pre-Emptive Planning Gain, can be used to develop a notion of *equilibrium*, whereby the two competing mechanisms are balanced. The proposed Non-Myopic Risk Equilibrium, that corresponds with the optimal horizon length, $N_p^\star$, is that which balances the expected gain with the impact of the expected risk:

$$N_p^\star = \arg\min_{N_p} E_{\mathcal{X},\mathcal{Z}}\{P_{nmg}(N_p) - f_u(P_{nmr}(N_p))\}, \tag{5.35}$$

where f_u, is a function which maps the Pre-Emptive Planning Risk into changes in expected utility. It is the identification of this function that is likely to prove most challenging in this type of approach. An alternative approach would be to use a 'regularised' optimisation structure, where large horizon sizes are penalised, thus avoiding the necessity to find a route to quantifying the pre-emptive measures outlined in this chapter.

5.10.2 Adaptive Horizon Planning Strategy

A generic, high-level adaptive horizon planning strategy can now be defined as follows:

[8]The use of the term optimisation horizon throughout this work leads to the somewhat convoluted term, 'optimal optimisation horizon'.

Algorithm 2 Adaptive Horizon Planning Strategy

loop

Compute the Pre-Emptive Planning Gain

Compute the Pre-Emptive Planning Risk

Identify the Non-Myopic Risk Equilibrium and the associated optimal lookahead horizon, $N_p^\star$

Compute the optimal sensor action sequence using $N_p^\star$

Actuate the sensor action

Observe the new data

end loop

5.11 Concluding Remarks

This chapter has considered a number of aspects of the performance of non-myopic control strategies for SM problems. A large number of potential error sources were identified, relating to prediction, utility and control strategies. It was shown that these error sources can contribute towards a degradation in planning performance when using limited-lookahead strategies. For instance, it was shown that multiple-step prediction efficiency is a function of the horizon length, and thus any utility estimates and subsequent control optimisation processes based on inefficient prediction will degrade further as the horizon increases. This effect is in direct opposition of the natural benefits of increased lookahead, which arise from the ability to identify paths through the decision space that yield more utility.

Two measures, Pre-Emptive Planning Risk, and Pre-Emptive Planning Gain, based on empirically quantifying these mechanisms where proposed, and it was subsequently suggested that these measures can provide the basis for a novel Non-Myopic Risk Equilibrium, which is associated with an optimal lookahead horizon. A novel adaptive horizon control strategy was also proposed, which notes that the optimal horizon size is likely to change online. It is noted that these issues are most likely to arise in tracking scenarios which deal with a high degree of uncertainty, for instance when tracking manoeuvering targets with uncertain dynamics.

Identifying the optimal horizon length is a very complex task and may prove extremely challenging for many applications for various reasons (including the inability to compute it analytically, and a potential lack of empirical data with which to estimate it). This has much to do with the difficulty of quantifying the performance of receding horizon control strategies in general, which Bertsekas points out is extremely challenging, due to the complex feedback loops and stochastic dependencies in the problem structure [65].

Despite the introductory nature of this conceptual framework, it represents a notable contribution to the field, by providing at the bare minimum, a number of potential

methods for assessing such strategies. In addition, the demonstration of the relationship between horizon size and multiple-step prediction efficiency, followed by its impact on the optimality of different length optimisation horizons, is considered to be completely novel despite its apparent simplicity.

5.12 Parallels in Other Fields

The concepts outlined in this chapter are applicable to a wide range of sequential decision making problems where there is uncertainty about the variables on which optimal decisions depend. Such problems are often encountered in fields such as control, decision theory, and economics. For example, the use of MPC for controlling industrial plants shares many similar attributes to the SM problems analysed in this chapter. In this case, the uncertainty may lie in the knowledge of the plant (or system model) which may be derived through system identification or other similar methods. Multiple-step predictions of plant-state can, therefore, suffer from the same amplification of risk as is described in this chapter.

A similar scenario arises in the economic case where predictive methods are used as the basis for portfolio optimisation. Here, the predictions relate to the behaviour of the market, and will clearly be non-exact and be based on significant approximations. Thus the accuracies of the predictions and assessment of return may decrease as the strategies move further into the future.

The key differentiator between the control theory and economics examples will lie in the ability to establish anything about the competing mechanisms, and how they might be balanced. It is not unreasonable to expect that some ground might be made in the control example, if sufficient analysis and experimentation can be conducted. In the economic example, however, it may be completely unfeasible to model the mechanisms to any degree of accuracy, due to the sheer level of uncertainty and approximation that is present. However, the analysis presented in this thesis still provides a starting point for identifying similar concepts in such fields.

Chapter 6

Global Optimisation for Optimal Sensor Control

6.1 Introduction

Many SM optimisation problems, including those posed in Chapters 4 and 5 exhibit characteristics that make solving them relatively hard for conventional gradient-based optimisation methods. For instance, the sequential nature of the decision making problems posed in Chapter 5 may cause the resulting objective landscapes to be non-smooth and induce multiple local minima. These effects have been observed in other non-myopic SM studies, for example in [2], where solutions to receding horizon control problems are found by seeding the initial optimisation with solutions to associated subproblems. A more general approach to solving these classes of optimisation problem is to use a stochastic search algorithm.

As discussed in Chapter 4, in many scenarios there will be multiple performance objectives relating to both perceptive and non-perceptive tasks. In purely autonomous SM contexts the relative importance of the various objectives will be derived from a higher-level mission planning system and therefore an appropriate preference formulation can be derived. However, there may also be cases where it is of benefit to directly compute or approximate the Pareto-optimal solution set, as defined in Chapter 4. This is especially relevant in situations where a human (e.g. a commander) interacts with the fusion system and wishes to examine the trade-offs between various objectives throughout the mission[1].

A modern sensor network may be very diverse. As a result, the nature of the decision variable space can also vary. Some classes of sensor may have discrete operating modes[2]

[1] It is not unreasonable to conjecture that an entirely autonomous system may also make use of knowledge of the attributes of the Pareto-optimal set.

[2] In the simplest case, ON and OFF, and in more complex cases, Radar waveform modes.

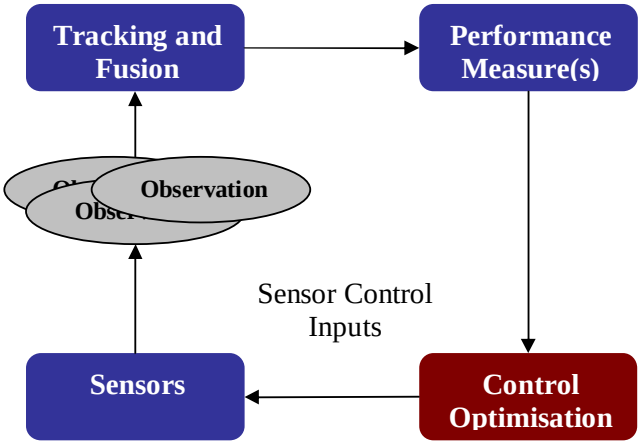

FIGURE 6.1: Optimisation aspect of Sensor Management architecture.

and some may have continuous operating parameters such as pointing angle and velocity. The decision variable constraints are also likely to contribute to the diversity evident in the problem structure.

A key aspect of the online SM problem is that the entire optimisation problem structure can be dynamic. As the mission evolves, both the objective space may change in nature (i.e. in dimensionality and cardinality of the set of objective functions). In addition, as sensor assets come online or become damaged or retired assets go offline, the decision variable space and associated constraints may also change rapidly. A practical solver must therefore be robust to changes in the problem space. Robustness in this context relates to the solver's overall ability to provide 'good' solutions to a variety of different classes of problems, and also to the sensitivity of the performance with respect to the algorithmic tuning parameters.

As an illustrative example, consider the case where a single airborne sensor platform is searching a ROI for a (friendly) lost target. During the search a number of hostile targets are detected and a number of additional heterogenous sensor platforms join the network. The overall mission objectives and, therefore, the optimisation objective functions and the decision variable space may alter drastically. As a result it is vital that the optimisation algorithm employed reacts gracefully.

The generic SM problem can, therefore, have dynamic decision variable spaces, dynamic decision variable constraints, and dynamic multiple-objective functions. In addition, the objective functions and constraints can be:

- non-smooth;

- non-linear;
- discontinuous;
- and, non-differentiable[3]

Optimisation problems exhibiting many of the above characteristics are not unusual and stochastic search techniques such as GAs, Simulated Annealing and Swarm Optimisation have been applied successfully in a number of fields. There has also been a large amount of research conducted into modifying these algorithms, in particular the GA, so that they may be applied to constrained, multiple-objective problems, such as those described above. A few of many examples include [73], [79], and [80].

Repeated Weighted Boosting Search is a guided search or meta-heuristic global optimisation algorithm that was recently proposed as an alternative to the more widely used techniques [81]. RWBS is essentially a multiple-start local-search technique, where the local optimiser is based on an iterative, adaptive[4], weighted convex combination. The convex combination is analogous to the crossover operator in a GA. In conjunction with a reflection operator, the convex combination generates new solutions in a manner similar to the Nelder-Mead simplex method [83]. The advantages of RWBS are its ease of implementation, limited number of tuning parameters, and levels of performance comparable with conventional techniques [81]. While RWBS is a very useful optimisation algorithm, its application in original form is restricted to unconstrained, single-objective optimisation problems with continuous search spaces.

The remainder of this chapter presents a number of novel extensions to the RWBS algorithm to facilitate its application to a more generic optimal sensor control context. The proposed extensions are aimed at increasing the flexibility of the algorithm while retaining its attractive features, which make it suitable for the optimisation problems encountered in optimal sensor control. Ultimately, the work presented in this chapter summarises a number of investigations conducted to assess whether the RWBS algorithm could be successfully extended in these directions. The primary contributions derived from this chapter are a number of extensions to the RWBS algorithm, in addition to a more thorough assessment of its relative performance, and an analysis of its properties for a more general class of optimisation problem.

6.2 Chapter Outline

The remainder of this chapter is organised as follows. In Section 6.3, necessary background regarding the RWBS algorithm is presented. In Section 6.4, an analysis of the

[3]When operating as a sub-system of a larger overall fusion system, the objective functions and constraints may be 'inherited' or passed down without gradient information.

[4]The adaptive weight update process is a modified Boosting technique [82].

performance of the RWBS algorithm is provided, including the results of a number of benchmarking experiments, whereby the performance is compared to that of more common techniques. A number of parameter sensitivity experiments are also presented in Section 6.4.3 which lead to various insights into parameter tuning, which are highlighted in Section 6.4.4.

The computational complexity of the RWBS algorithm is analysed in Section 6.5. This is followed by a method to adapt the algorithm to operate in mixed search spaces in Section 6.6, which includes benchmarking experiments.

A Pareto-RWBS technique is presented in Section 6.8, along with the results of a number of convergence comparison experiments based on a number of well established test problems. Concluding remarks are offered in Section 6.9.

6.3 Repeated Weighted Boosting Search

Consider the general global optimisation problem:

$$\min_{\mathbf{u}\in U} J(\mathbf{u}) \tag{6.1}$$

where $\mathbf{u} = [u(1), u(2), \ldots, u(n)]^T$ is the n-dimensional vector to be optimised, U is the feasible set of $\mathbf{u}$ and $J(\mathbf{u})$ is the cost function. The original RWBS algorithm proceeds as follows [81]: Define a population of P_s points: $\mathbf{u}_i \in U$ for $1 \leq i \leq P_s$ - these points are initially chosen at random. Let $\mathbf{u}_{best} = \arg\min J(\mathbf{u})$ and $\mathbf{u}_{worst} = \arg\max J(\mathbf{u})$; next, a $(P_s + 1)$-th point is generated by performing a convex combination of $\mathbf{u}_i$:

$$\mathbf{u}_{P_s+1} = \sum_{i=1}^{P_s} \delta_i \mathbf{u}_i, \tag{6.2}$$

where the weights satisfy $\delta_i \geq 0$ and

$$\sum_{i=1}^{P_s} \delta_i = 1. \tag{6.3}$$

The point $\mathbf{u}_{P_s+1}$ is always within the convex hull defined by $\mathbf{u}$. A mirror image of $\mathbf{u}_{P_s+1}$ is then generated with respect to $\mathbf{u}_{best}$ and along the direction defined by $\mathbf{u}_{best} - \mathbf{u}_{P_s+1}$ as

$$\mathbf{u}_{P_s+2} = \mathbf{u}_{best} + (\mathbf{u}_{best} - \mathbf{u}_{Ps+1}). \tag{6.4}$$

If $\mathbf{u}_{P_s+1}$ or $\mathbf{u}_{P_s+2}$ are outside U, they can be projected back to U. The best of $\mathbf{u}_{P_s+1}$ and $\mathbf{u}_{P_s+2}$ replaces $\mathbf{u}_{worst}$ according to their cost function values. This process is repeated

until the population converges. The convergence can be assumed, for example, if

$$||\mathbf{u}_{P_s+1} - \mathbf{u}_{P_s+2}|| < \xi_B, \tag{6.5}$$

where the small positive scalar, ξ_B, is the chosen accuracy. The weights, δ_i, are adapted according to the Boosting technique [81]. In the first iteration, the weights are distributed uniformly:

$$\delta_i(0) = \frac{1}{P_s}, 1 \leq i \leq P_s. \tag{6.6}$$

In subsequent iterations the weights are updated using the following procedure. First, the cost function values are normalised:

$$\bar{J}_i = \frac{J_i}{\sum_{i=1}^{P_s} J_i}, 1 \leq i \leq P_s, \tag{6.7}$$

then a weighting factor $\beta(t)$ is computed:

$$\eta(t) = \sum_{i=1}^{P_s} \delta_i(t-1)\bar{J}_i, \quad \beta(t) = \frac{\eta(t)}{1-\eta(t)}. \tag{6.8}$$

The weights are then updated for $1 \leq i \leq P_s$:

$$\delta_i(t) = \left\{ \begin{array}{ll} \delta_i(t-1)\beta(t)^{\bar{J}_i} & \text{if } \beta(t) \leq 1 \\ \delta_i(t-1)\beta(t)^{1-\bar{J}_i} & \text{if } \beta(t) > 1 \end{array} \right\}, \tag{6.9}$$

and normalised:

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{i=1}^{P_s} \delta_i(t)}, 1 \leq i \leq P_s. \tag{6.10}$$

The weighted Boosting search procedure described above is repeated multiple times with random population initialisation in order to search for a global optimum. Each repetition is termed a generation and an elitist based initialisation is used (i.e. the best solution found in the previous generation is kept). The global optimisation algorithm can then be described as follows [81].

Specify the following algorithmic parameters: P_s - population size; N_g - number of generations in the repeated search; N_B - number of iterations in the weighted Boosting search; ξ_B - accuracy for terminating the weighted Boosting search.

- **Outer loop : for generations** $g = 1 : N_g$
- – *Generation initialisation*: Initialise the population by setting $\mathbf{u}_1^{(g)} = \mathbf{u}_{best}^{(g-1)}$ and randomly generating the rest of the population members $\mathbf{u}_i^{(g)}, 2 \leq i \leq P_s$ where $\mathbf{u}_{best}^{(g-1)}$ denotes the solution found in the previous generation. **If** $g = 1$**,** $\mathbf{u}_1^{(g)}$ is also randomly chosen.

– Weighted Boosting search initialization: Assign the initial distribution weights, $\delta_i(0) = \frac{1}{P_s}, 1 \leq i \leq P_s$, for the population, and calculate the cost function value of each point:

$$J_i = J\left(\mathbf{u}_i^{(g)}\right), 1 \leq i \leq P_s.$$

- **Inner loop : weighted Boosting search** for $t = 1 : N_B$
- *Step 1*: Boosting

1. Find:

$$i_{best} = \arg \min_{1 \leq i \leq P_s} J_i$$
$$i_{worst} = \arg \max_{1 \leq i \leq P_s} J_i.$$

Denote $\mathbf{u}_{best}^{(g)} = \mathbf{u}_{i_{best}}^{(g)}$ and $\mathbf{u}_{worst}^{(g)} = \mathbf{u}_{i_{worst}}^{(g)}$

2. Normalise the cost function values:

$$\bar{J}_i = \frac{J_i}{\sum_{i=1}^{P_s} J_i}, \quad 1 \leq i \leq P_s.$$

3. Compute a weighting factor $\beta(t)$ according to:

$$\eta(t) = \sum_{i=1}^{P_s} \delta_i(t-1)\bar{J}_i, \quad \beta(t) = \frac{\eta(t)}{1-\eta(t)}.$$

4. Update the distribution weights for $1 \leq i \leq P_s$:

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta(t)^{\bar{J}_i} & \text{for } \beta(t) \leq 1 \\ \delta_i(t-1)\beta(t)^{1-\bar{J}_i} & \text{for } \beta(t) > 1 \end{cases}$$

and normalise them:

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{j=1}^{P_s} \delta_i(t)}, 1 \leq i \leq P_s.$$

- *Step 2*: Parameter Updating

1. Construct the $(P_s + 1)$-th point using the formula:

$$\mathbf{u}_{P_s+1} = \sum_{i=1}^{P_s} \delta_i(t)\mathbf{u}_i^{(g)}.$$

2. Construct the $(P_s + 1)$-th point using the formula:

$$\mathbf{u}_{P_s+2} = \mathbf{u}_{best}^{(g)} + (\mathbf{u}_{best}^{(g)} - \mathbf{u}_{Ps+1}).$$

3. Compute the cost function values $J(\mathbf{u}_{P_s+1})$ and $J(\mathbf{u}_{P_s+2})$ for these two points, and find:

$$i_* = \arg \min_{i=P_s+1,P_s+2} J(\mathbf{u}_i).$$

4. The pair $\{\mathbf{u}_{i_*}, J(\mathbf{u}_{i_*})\}$ then replaces $\{\mathbf{u}_{worst}^{(g)}, J_{i_{worst}}\}$ in the population.

- – If $||\mathbf{u}_{P_s+1} - \mathbf{u}_{P_s+2}|| < \xi_B$, exit **Inner loop**
  – End of **Inner loop** The solution found in the g-th generation is $\mathbf{u}_{best}^{(g)}$
- **End of Outer loop** This yields the solution $\mathbf{u}_{best}^{(N_g)}$

6.4 Optimisation Performance Analysis

Introductory RWBS performance testing was reported in [81]. The following section provides a more thorough analysis of RWBS in terms of its convergence to global optimality, and the effect of the tuning parameters on its performance. The performance is also benchmarked against standard stochastic search and other algorithms on a number of test problems.

6.4.1 Convergence to Global Optimality

There are two main factors which affect the global convergence performance of RWBS. In the simplest terms, RWBS is guaranteed to reach global optimality due to the stochastic search component in Equation (6.3). Since there is always a population member generated using a uniform distribution over the decision space domain, then the algorithm will, in the limit, converge to the global minimum with probability one (assuming that the global minimum lies within the chosen domain). The performance of RWBS is, therefore, lower bounded by that of pure stochastic search:

$$C_{RWBS} \geq C_{RS}, \tag{6.11}$$

where C_{RWBS} and C_{RS} are the convergence rates of RWBS and pure random search (with the same generating distribution), respectively.

The convergence rate of RWBS is, however, expected to exceed that of random search due to the local search component in Equation (6.2). The nature of a successful local search technique dictates that it should converge to the local optimum very rapidly and

thus the overall performance of RWBS will, in part, hinge on the combined convergence rate of the convex combination and reflection operators. However, it will be shown that the definition of 'locality' that is appropriate for RWBS is different from that associated with conventional local search such as steepest-descent.

The steepest-descent algorithm is designed to find the local minimum which lies within the (convex) 'basin of attraction' of a single data point. It works by computing the gradient of the objective function at a point, and moving in the direction of steepest gradient. The weighted Boosting technique in RWBS, however, is a *population*-based technique designed to find a local minimum within the convex hull defined by the population $\mathbf{u}$. In many cases, this region will represent a larger subspace than the basin of attraction around a single point, especially if the search space is large and multi-modal. It should also be noted that due to the reflection operator in Equation (6.4), the size of the convex hull that is assessed by the RWBS inner loop can increase if $J(\mathbf{u}_{P_s+1}) > J(\mathbf{u}_{P_s+2})$, and the solution found at the end of the inner search loop can in fact be outside of the convex hull defined by the population at the beginning of the inner loop. The local search in RWBS is, therefore, inherently less restrictive than a steepest-descent or hill-climbing technique as it is able to explore a larger subspace. RWBS appears to exhibit a *smoothing* or *low-pass* effect which guides the optimisation towards the optimum of a smoothed version of the function. Thus, in general, it is conjectured that RWBS is expected to perform well in cases were the cost function is approximately globally convex. A rigorous proof of this concept is the subject of further work.

RWBS shares a number of similarities with a GA (with elitism). Both are population-based techniques which combine a local search based on current members of the population (i.e. convex combination in RWBS and crossover in a GA), and a stochastic search component (the outer loop in RWBS and mutation in a GA), designed to prevent the algorithm from converging towards local optima. The performance of the two techniques can therefore be expected to be similar in general.

6.4.2 Benchmark Convergence Experiments

This section presents a number of benchmark convergence experiments which analyse the performance of RWBS in more depth than the presentation in [81]. More specifically, its performance is compared to random search, a multiple-start gradient-based search (MSGS) and a GA. A number of well known test functions with different attributes are utilised.

The first test function is a two–dimensional version of the widely used Ackley function:

$$J(\mathbf{u}) = -20\exp\left(-0.2\sqrt{\tfrac{1}{2}\sum_{i=1}^{2} u_i^2}\right) - \exp\left(\tfrac{1}{2}\sum_{i=1}^{2}\cos(2\pi u_i)\right) + 20 + e, \tag{6.12}$$

where $\mathbf{u} = [u_1, u_2]'$. This function is illustrated in Figure 6.2. For points outside of the interval $\{-10 \leq u_j \leq 10\}, j = 1, 2$, the cost function was assigned a value of 100. The

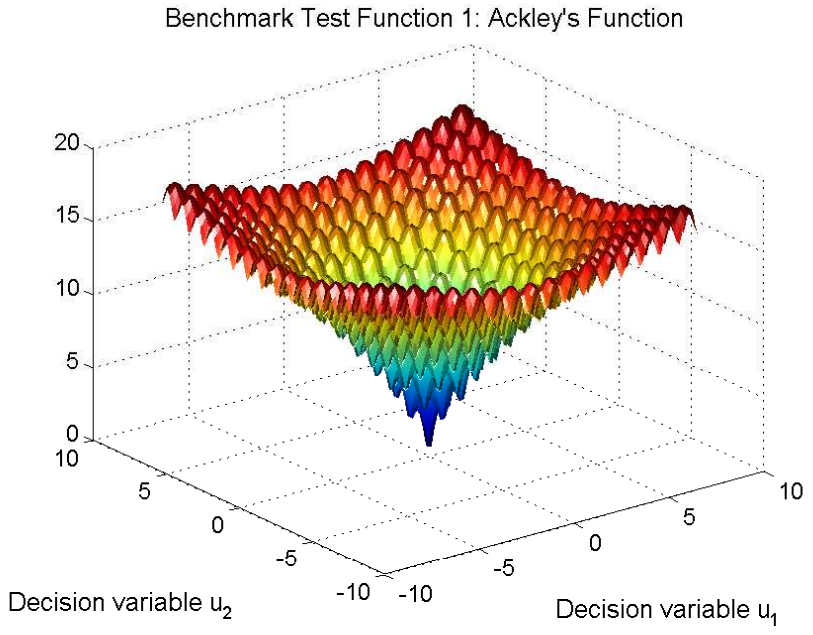

FIGURE 6.2: Two-dimensional Ackley function - in this problem there are multiple local minima and one global minimum at $\mathbf{u} = [0, 0]', (J(\mathbf{u}) = 0)$.

second test function used is the well known Rastrigin function:

$$J(\mathbf{u}) = 20 + u_1^2 + u_2^2 - 10(\cos 2\pi u_1 + \cos 2\pi u_2). \tag{6.13}$$

The surface of this function is illustrated in Figure 6.3. For points outside of the interval $\{-5 \leq u_j \leq 5\}, j = 1, 2$, the cost function was again assigned a value of 100. These functions were chosen as they are standard benchmark functions for assessing the performance of optimisation algorithms. Both are non-smooth and have multiple local optima which should present difficulties to MSGS. RWBS and GAs are generally expected to perform better in these types of problems.

Thirdly, a simpler function with just two local minima and one global minimum is tested:

$$\begin{aligned} J(\mathbf{u}) = 1.5(1 - \exp^{(-(u_1-1)^2-(u_2+1)^2)}) + \\ 1 - \exp^{(-(u_1+2)^2-(u_2-2)^2)} + \\ 1 - \exp^{(-u_1^2-(u_2-1)^2)}. \end{aligned} \tag{6.14}$$

The surface of this cost function is illustrated in Figure 6.4. For points outside the interval $\{-4 \leq u_j \leq 4\}, j = 1, 2$, the cost function was again assigned a value of 100. This function is smooth and the global optimum has a large basin of attraction, thus it should not present much difficulty to the MSGS method. This function was chosen

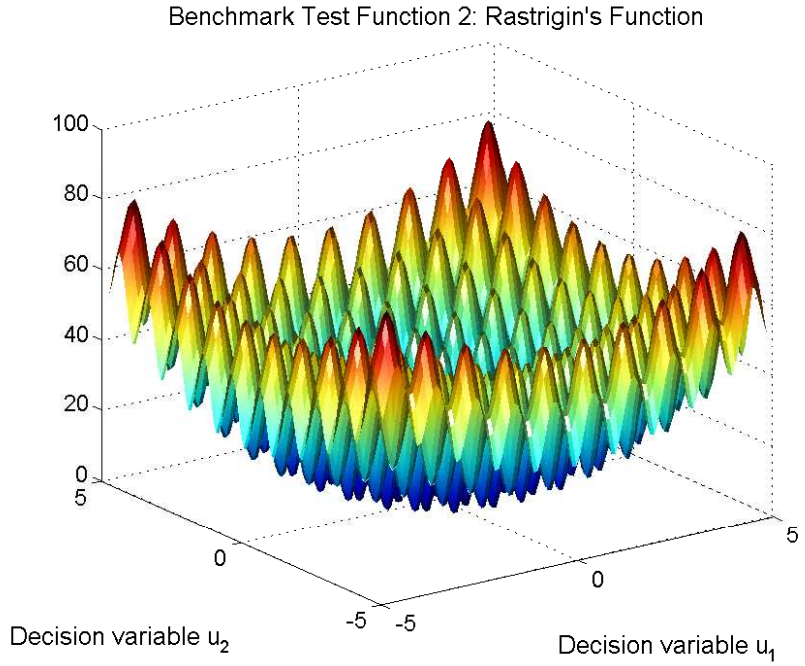

FIGURE 6.3: Two-dimensional Rastrigin's function - in this problem there are multiple local minima and one global minimum at $\mathbf{u} = [0, 0]', (J(\mathbf{u}) = 0)$.

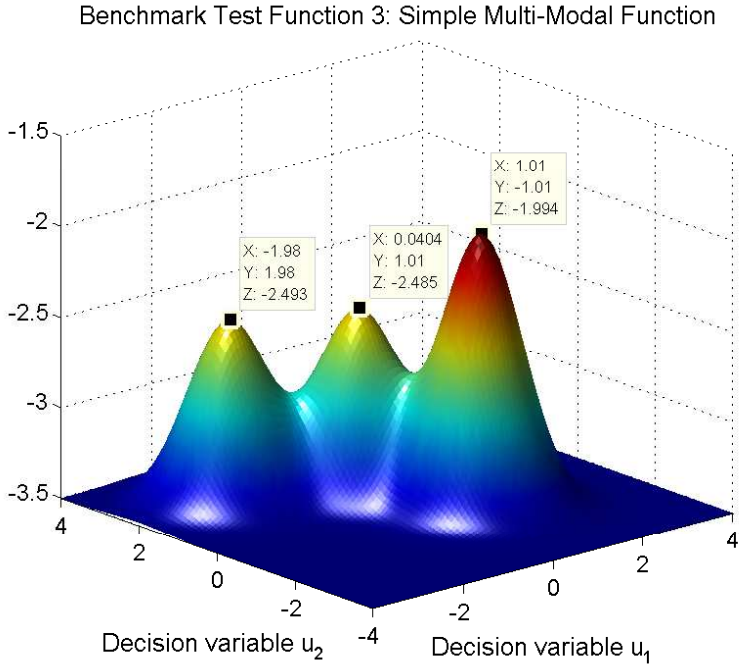

FIGURE 6.4: Two-dimensional benchmark cost function (plot inverted for clarity) - in this problem there are two local minima corresponding to $\mathbf{u} \approx [0, 1]', (J(\mathbf{u}) \approx 2.5)$ and $\mathbf{u} \approx [-2, 2]', (J(\mathbf{u}) \approx 2.5)$. The single global minimum is at $\mathbf{u} \approx [1, -1]', (J(\mathbf{u}) \approx 2.0)$.

in order to demonstrate the relative performance of RWBS on simpler problems where techniques such as MSGS are known to perform well.

In order to create a basic MSGS algorithm, single samples from a uniform distribution over the predefined variable intervals are repeatedly used to seed the MATLAB Optimisation Toolbox function *fminunc* [84]. As the gradient of the cost function is not passed to the function, it automatically uses a medium-scale algorithm based on the Broyden, Fletcher, Goldfarb, and Shannon (BFGS) Quasi-Newton method with a mixed quadratic and cubic line search procedure [84]. This is an industry standard gradient-based search technique. The random search method employed was also based on repeated uniform sampling over the respective interval. A GA was created using the MATLAB function *ga* from the GA and Direct Search Toolbox [85]. This high-performance GA is based on the following components: scattered crossover, zero-mean Gaussian mutation, stochastic uniform selection, double-vector real-coding, migration, and a relative rank-based fitness scaling process.

Figure 6.5 presents the results of the performance comparison on Ackley's function. The number of cost function evaluations is plotted against the mean best cost function value found so far, averaged over 100 MC experiments of each algorithm. In this case the RWBS algorithm was used with the following settings: $P_s = 11, N_g = 150, N_B = 15, \xi_B = 0.02$. These settings were chosen using very rapid coarse tuning experiments. The GA settings were default except for the initial population range which was set appropriately, and the population size was set to 100; these parameters were also chosen using coarse tuning experiments. The remaining default parameters are: crossover fraction = 2, elitism count = 2, migration interval = 20 and these remained constant throughout all the experiments.

Default settings for the maximum number of iterations and termination tolerances were used for the MSGS algorithm as they were deemed appropriate for the problem.

It is interesting to note that a number of rule-of-thumb-based approaches have been proposed for tuning GAs, for example in [86]. One approach for choosing a population size is to simply use approximately 10 times the number of dimensions (corresponding to a population size of 20 in this case); this method has been shown to provide good performance in a number of cases. However, the experiments presented herein were repeated with a population size of 20 and the GA results were, in general, found to be inferior to those generated using the described parameters. One possible explanation for this observation is that the rule-of-thumb may make assumptions about the nature of the GA (e.g. regarding coding, crossover and mutation operators, and fitness scaling mechanisms etc) which are not justified in this case.

Additional rules-of-thumb for selecting GA parameters such as crossover rate, mutation rate, and number of generations have also been proposed. Investigating the relative performance of RWBS against the GA tuned using appropriate rules is, therefore, an interesting area of future work. All of the results presented in this section are based on direct trial and error parameter tuning for RWBS, the GA, and MSGS, rather than

using rules-of-thumb. The comparison outlined in this chapter is, therefore, considered to be fair.

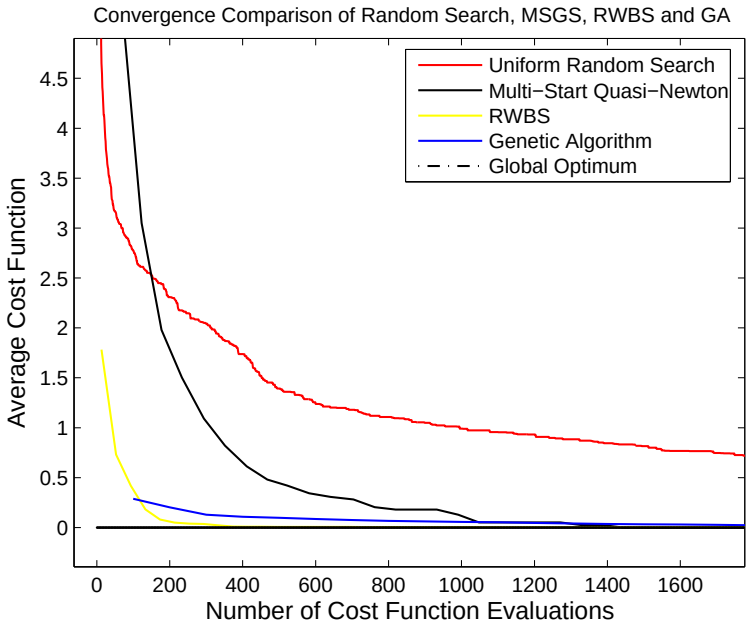

FIGURE 6.5: Convergence performance comparison of RWBS, random search and MSGS on test problem (6.12) - Ackley's function.

RWBS performs significantly better than the other algorithms on this test function with only brief tuning of the two major parameters (P_s and N_B). On average, RWBS finds a solution within 0.05 of the global optimum after only 253 cost function evaluations. The GA requires 1200 evaluations in order to reach the same performance. MSGS requires in the order of 1327 evaluations and the random search procedure remains at least 0.6 away from the global optimum after 2000 cost function evaluations.

Figure 6.6 shows the results generated with Rastrigin's function. In this case the RWBS algorithm was used with the following settings which were found to be appropriate: $P_s = 11, N_g = 150, N_B = 15, \xi_B = 0.02$. Once again, default GA settings were used except for the initial population range which was set appropriately and the population size was set to 100. Default settings for the maximum number of iterations and termination tolerances were used for the MSGS algorithm. In this example, the performance offset, in terms of rate of change of average cost function between RWBS and the GA, is smaller. Both algorithms find a value within 0.05 of the global optimum within 1900 cost function evaluations (GA-1900, RWBS-1773). As expected, MSGS requires significantly more evaluations (>6900), as does random search.

Figure 6.7 illustrates the performance comparison using the simpler function defined in Equation (6.14). In this case the RWBS algorithm was used with the following settings which were found to be appropriate: $P_s = 6, N_g = 150, N_B = 5, \xi_B = 0.02$. The GA population size was once again set to 100. Default settings for the maximum number of

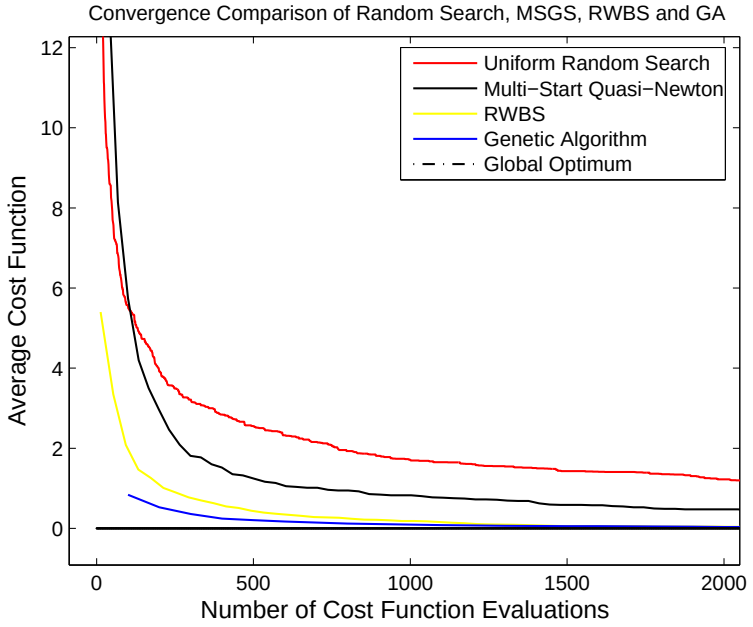

FIGURE 6.6: Performance comparison of RWBS, random search and MSGS on test problem (6.13) - Rastrigin's function.

iterations and termination tolerances were used for the MSGS algorithm. It is clear from

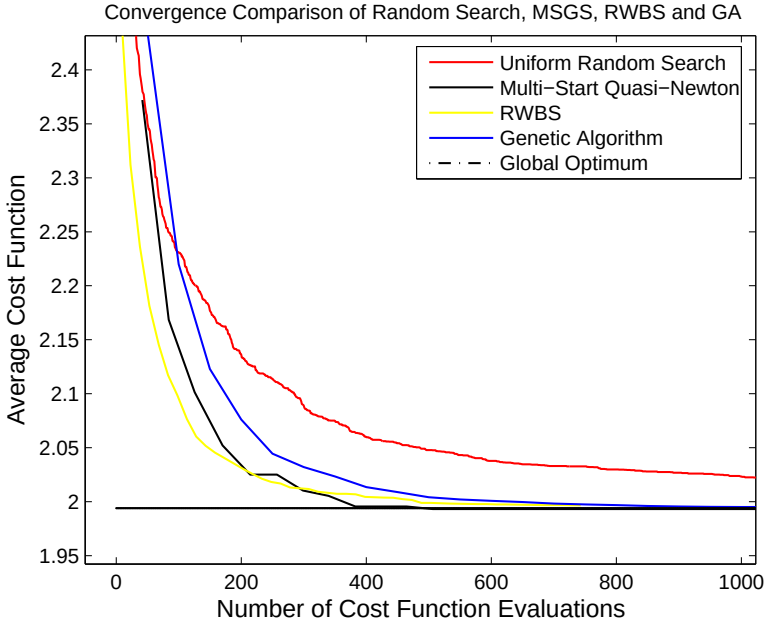

FIGURE 6.7: Performance comparison of RWBS, random search and MSGS on test problem (6.14) - Simple Multi-Modal function.

Figure 6.7 that neither the RWBS or GA algorithms significantly outperforms MSGS. This is expected as the basin of attraction for the global minimum is very large compared to that of the global minimum of Ackley's or Rastrigin's function. MSGS is, therefore, likely to find the global minimum quite rapidly. RWBS performs at least as well as

MSGS and the GA in this case. Results show that RWBS finds a solution within 0.05 of the global minimum within 173 evaluations, GA within 300 and MSGS within 214.

TABLE 6.1: Results of convergence performance comparison - mean number of function evaluations required to identify solution within 0.05 of the global optimum

Algorithm / Test Function	Ackley	Rastrigin	Simple Multi-Modal
MSGS	1327	> 6900	214
RWBS	**253**	**1773**	**173**
Random Search	> 2000	> 6900	> 2000
GA	1200	1900	300

Table 6.1 summarises the performance comparison for the various test functions. In each case, RWBS converges to within 0.05 from the true global minimum faster than all of the other algorithms (as highlighted by the bold font in the table). This provides additional support to the conjecture that RWBS is a very promising algorithm for black-box optimisation problems, especially considering the very minor tuning requirements involved and the ease of implementation. In most cases examined herein the RWBS algorithm was found to perform well with the key parameters set to $5 \leq Ps \leq 15$ and $5 \leq N_b \leq 20$. Coarse tuning within these ranges can be used to fine tune the performance very quickly.

6.4.3 Convergence Parameter Sensitivity

The performance of RWBS on a given problem is dependent on the cost function to be optimised, and on the selected RWBS parameters. In order to make efficient use of RWBS, therefore, the robustness of the algorithm's performance to parameter changes is investigated. In this section, each of the benchmark cost functions tested in the previous section is optimised using a range of RWBS parameters. As the original RWBS algorithm has a total of five tuning paramaters - population size, number of boosts, number of outer loops, and the termination threshold, the parameter trade-off space is five-dimensional. If each parameter is tested over N discrete values, then the total number of optimisation runs extends to N^5. However, in practice the real parameters of interest are the population size, and the number of boosts. Thus the algorithms are tested over a range of these parameters. It is informative to perform this process on benchmark functions offline in order to provide some pragmatic suggestions for good parameter choice. In all cases, the algorithm was tested over the parameter ranges $3 \leq P_s \leq 50$, and $3 \leq N_b \leq 30$, and the results were averaged over 50 MC simulations. In each case, the diagrams indicate the minimum number of cost function evaluations required to maintain the performance level described in the previous section.

The number of generations and the termination threshold were not varied. The number of generations was set very high to ensure that the algorithm had sufficient time to

converge, and the termination threshold was kept constant. In practice, the number of generations must be chosen to be large enough to allow convergence, in a similar way to which the number of generations used in a GA is chosen. Varying the number of generations in this context would not provide significant insight as it will simply effect whether or not the algorithm finds a solution which satisfies the threshold during the course of the simulation. The number of boosts is therefore of more direct interest with regards to the convergence rate (rather than convergence success). Varying the termination threshold would, however, provide additional insight into the rate of convergence and how it changes as the algorithm proceeds. This is the subject of future work.

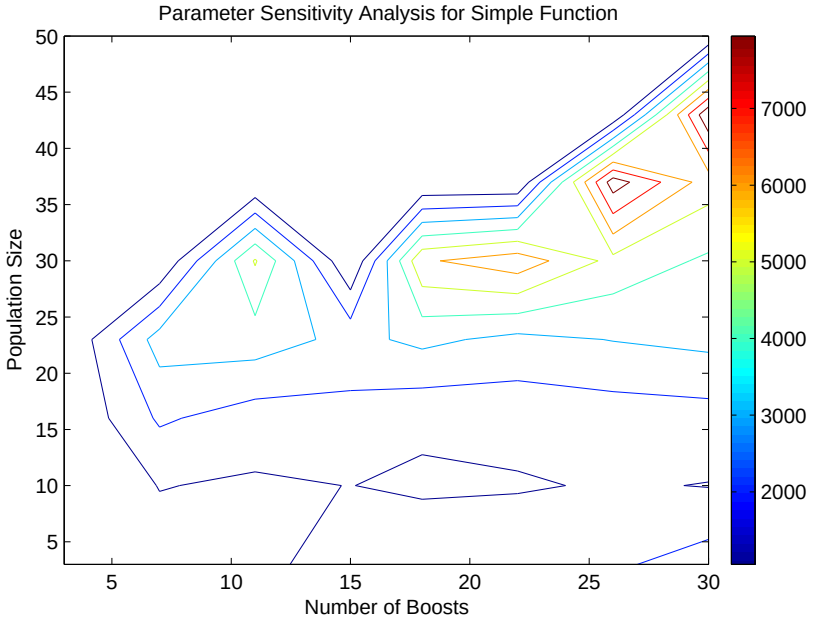

FIGURE 6.8: Parameter sensitivity contour plot for RWBS applied to simple function.

Figures 6.8 and 6.9 present the parameter sensitivity analysis for RWBS applied to the simple multi-modal test problem in the previous section. It is noted that the areas of the surface that appear to require zero evaluations correspond to those simulations where a solution of the required accuracy was not found. It is immediately noticeable from Figures 6.8 and 6.9, that, in general, the larger the population size, the larger the number of evaluations required. This provides justification for small population sizes. It is interesting to note that the optimum performance appears to occur when a population size of around 10 is chosen. As the population size increases, the algorithm requires more evaluations to converge. These results also indicate a level of insensitivity to the number of boosts, and provide evidence for the assertion that the population size is the key parameter. Figures 6.10 and 6.11 present the parameter sensitivity analysis for RWBS applied to Rastrigin's function. A different trade-off surface is observed, in that the algorithm now exhibits more sensitivity to the number of boosts for small and large population sizes. However, it is noted that if the population size is set to

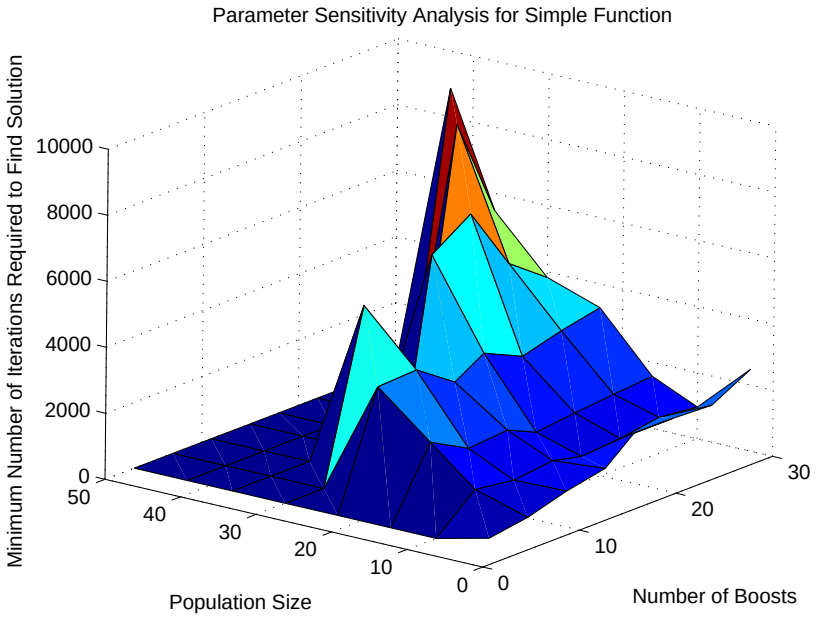

FIGURE 6.9: Parameter sensitivity surface for RWBS applied to simple function.

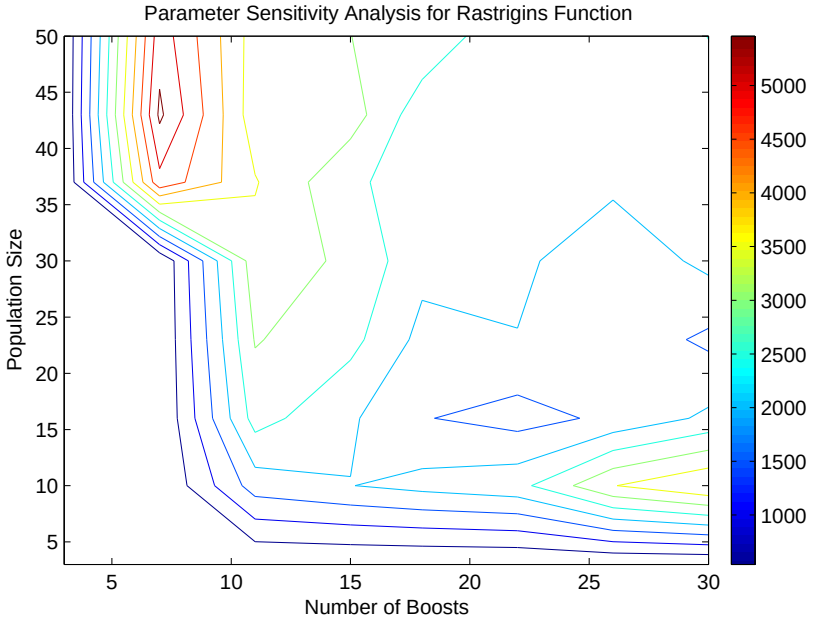

FIGURE 6.10: Parameter sensitivity contour plot for RWBS applied to Rastrigin's function.

an appropriate value (sizes less than 20 seem to offer reasonable performance), there is still a level of insensitivity to the number of boosts. Figures 6.12 and 6.13 present the parameter sensitivity analysis for RWBS applied to Ackley's function. Similar results are observed in regard to the insensitivity to the number of boosts, and the levels of performance achieved with low population sizes. In summary, therefore, the parameter

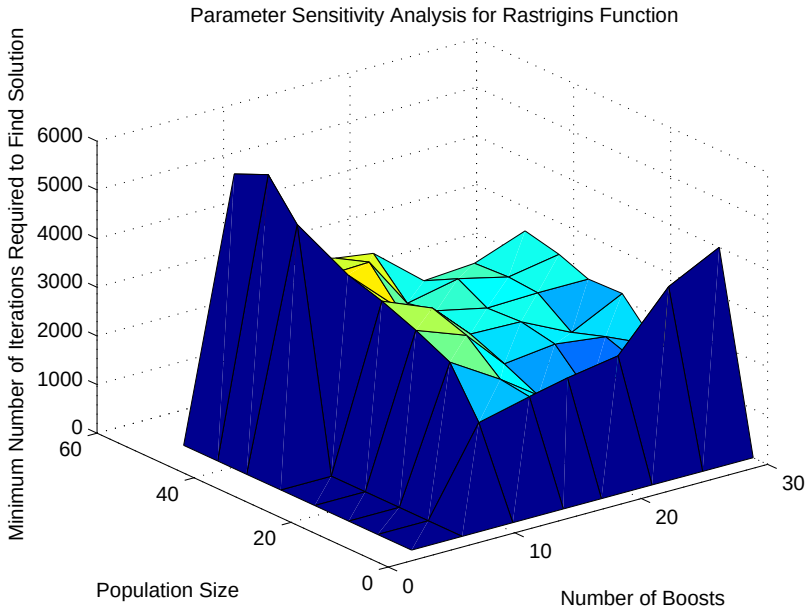

FIGURE 6.11: Parameter sensitivity surface for RWBS applied to Rastrigin's function.

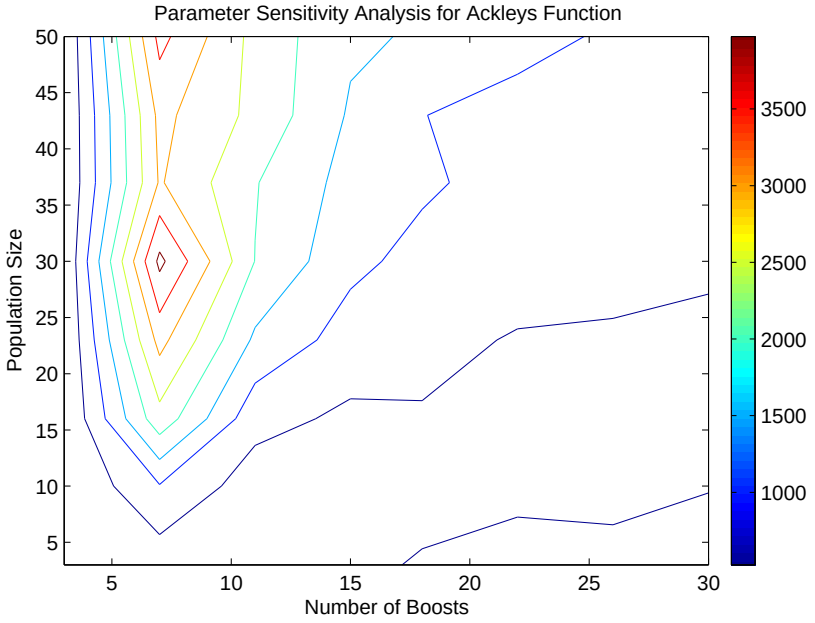

FIGURE 6.12: Parameter sensitivity contour plot for RWBS applied to Ackley's function.

sensitivity results indicate that the key parameter of importance is the population size, P_s. If an appropriate choice of this parameter is made, the algorithm exhibits a level of insensitivity to the number of boosts. In addition the results demonstrate that small population sizes are capable of providing good levels of performance across a range of different cost functions, further strengthening the potential of the RWBS algorithm for

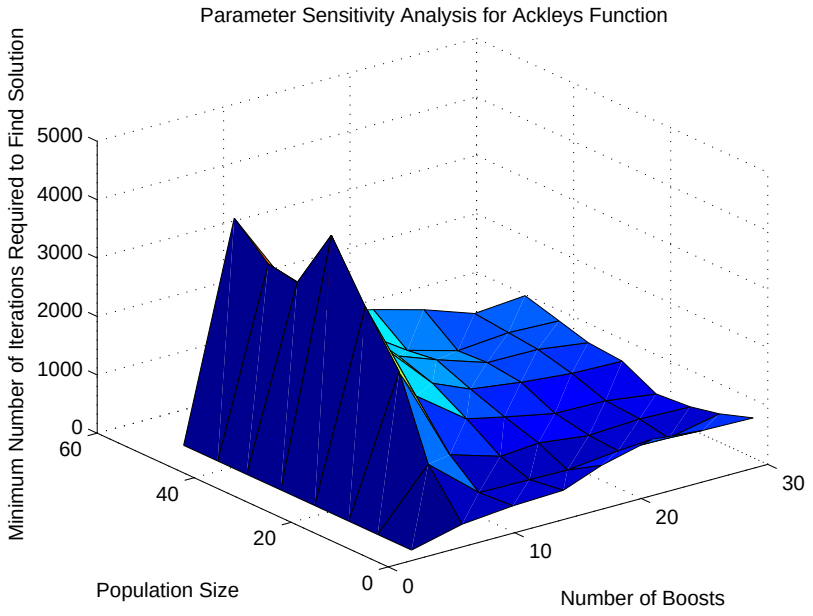

FIGURE 6.13: Parameter sensitivity surface for RWBS applied to Ackley's function.

black-box optimisation, in problems such as those relevant to autonomous SM.

6.4.4 Parameter Tuning

Like most stochastic search algorithms, RWBS must be tuned to each particular problem to achieve the best results. The main trade-off in relation to convergence *success* is between the number of generations, N_g, and the number of boosts, N_b, which balances the stochastic component of the algorithm with the deterministic local search component. Appropriate choice of parameters depends on whether the cost function is suspected to contain multiple local minima. As the population size increases, the effect of each Boosting iteration is attenuated. A larger population is likely to cover a larger subspace and thus it takes longer for the Boosting process to focus on a particular region of interest. Experiments suggest that larger populations, therefore, demand higher values for N_b. The advantage of a larger population is that the probability of encompassing the basin of attraction of the global minimum is increased. If the function is also relatively convex then the probability of locating the global minimum in a shorter time period is also increased. Initial experience with the algorithm suggest that smaller populations often provide more efficient results for non-smooth cost functions. The results outlined in the previous section also indicate that if the population size is chosen appropriately, the algorithm will exhibit insensitivity to the number of boosts.

6.5 Computational Complexity Analysis

The computational complexity of RWBS can be analysed in following manner. Each inner loop iteration, of which there are N_b, consists of $\mathcal{O}(P_s)$ operations, where the $\mathcal{O}$ symbol represents 'order of'. Each outer loop iteration, or 'generation', of which there are N_g, therefore consists of $\mathcal{O}(N_b P_s)$ operations. The total computational complexity of RWBS is therefore $\mathcal{O}(N_g N_b P_s)$.

6.6 Mixed Weighted Boosting Search

The original RWBS algorithm described in Section 6.3 is only applicable to problems where the decision variable is defined over a continuous search space. This is because the weighted convex combination in the local search inner loop (Equation (6.2)) is a continuous function. However, in many problems, the decision variable search space may be a discrete space, or a mixture of continuous space and discrete space. The latter class of problems are sometimes known as mixed problems. It should also be noted that in some cases, the cost function may only be defined or evaluated on the discrete points and so methods based on continuous optimisation followed by discretisation are not always applicable. Consider the case where the search space is purely discrete, such as that illustrated in Figure 6.14. In this case the search space is also regular and all feasible points are located on multiples of 0.25. In order to 'convert' the algorithm to operate

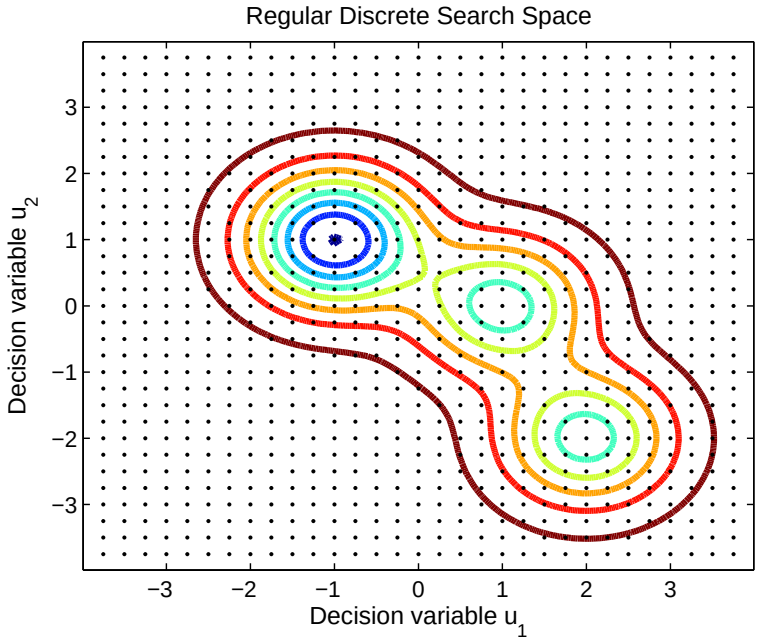

FIGURE 6.14: Example regular discrete decision variable space overlayed with contour plot of cost function (6.14). Black markers correspond to possible decision variable values.

over discrete search spaces, the convex combination operator must be constrained in

some way to generate discrete solutions. A modified parameter update step is proposed here and the resulting algorithm is termed Nearest-Neighbour RWBS (NNRWBS). This new algorithm retains the original convex combination but modifies the generated points so that they are assigned to a feasible value in the discrete space. It is assumed that the underlying search space is a regular grid in n-dimensional space.

6.6.1 Nearest-Neighbour Parameter Update

A simple and efficient method to generate a new discrete point from a weighted combination of points in discrete space is as follows: firstly, generate an intermediate point, $\mathbf{v}_1$, by computing the weighted convex combination as in (6.2); a second intermediate point, $\mathbf{v}_2$, is then generated by reflection using Equation (6.4); next, assign or 'snap' the new points to the nearest discrete points according to some distance measure (e.g. Euclidean distance). The resulting NNRWBS algorithm is identical to the original outlined in Section 6.3, with the exception of the parameter update stage which is replaced with the following (this particular example uses the Euclidean norm as the distance metric):

- *Step 2*: NNRWBS Parameter Update:

1. Construct the $(P_s + 1)$-th point - first generate the intermediate point using:

$$\mathbf{v}_1 = \sum_{i=1}^{P_s} \delta_i(t)\mathbf{u}_i^{(g)}, \tag{6.15}$$

   then assign the new point to the discrete grid using:

$$\mathbf{u}_{P_s+1} = \arg\min_{\mathbf{u}\in U} ||\mathbf{u} - \mathbf{v}_1|| \tag{6.16}$$

2. To construct the $(P_s + 2)$-th point, first use the formula

$$\mathbf{v}_2 = \mathbf{u}_{best} + (\mathbf{u}_{best} - \mathbf{v}_1{}^5), \tag{6.17}$$

   then assign the new point to the discrete grid using:

$$\mathbf{u}_{P_s+2} = \arg\min_{\mathbf{u}\in U} ||\mathbf{u} - \mathbf{v}_2|| \tag{6.18}$$

3. Compute the cost function values $J(\mathbf{u}_{P_s+1})$ and $J(\mathbf{u}_{P_s+2})$ for these two points, and find:

$$i_* = \arg\min_{i=P_s+1,P_s+2} J(\mathbf{u}_i). \tag{6.19}$$

[5] One could equally use $\mathbf{u}_{P_s+1}$.

4. The pair $(\mathbf{u}_{i*}, J(\mathbf{u}_{i*}))$ then replaces $(\mathbf{u}^{(g)}_{worst}, J_{i_{worst}})$ in the population.

Note that the 3^{rd} and 4^{th} steps of the new parameter update stage remain in original form, but the first two steps are now modified. The NNRWBS parameter update step for the $(P_s + 1)$-th point is illustrated in Figure 6.15. The green markers indicate the

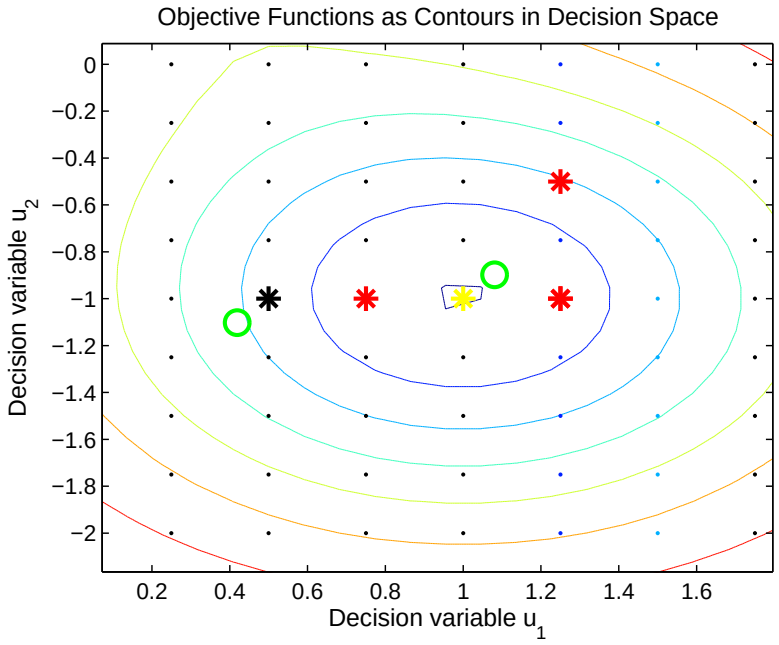

FIGURE 6.15: Illustration of the NNRWBS parameter update on the Simple Multi-Modal function (6.14) using a regular discrete search space with feasible points located on multiples of 0.25 as in Figure 6.14.

two intermediate points, v_1 and v_2. The red markers illustrate the original members of the population and the yellow and black markers indicate the (P_s+1)-th and (P_s+2)-th points respectively[6].

6.6.2 Embedded Hull Parameter Update

While the convex combination in the nearest-neighbour parameter update stage can only generate intermediate points within the Euclidean hull defined by the population, the $(P_s + 1)$-th point can, of course, be outside of the Euclidean hull, as indicated in Figure 6.15. Despite this, use of the Euclidean convex operator can be considered restrictive as it does not account for the differences between notions of continuous and discrete convexity. When discrete spaces are modelled with cell complexes, a number of different types of convexity can be defined, each with different properties [87]. The embedded convex hull is particularly useful in this context due to its simple shape. By identifying the embedded convex hull defined by the population, it is possible to relax

[6]Note that if the search space is a regular grid then the $(P_s + 1)$-th point is guaranteed to be within the convex hull defined by u.

the parameter update so that it operates over a larger subspace, potentially leading to superior performance. An illustration of this concept can be found in Figure 6.16. Some

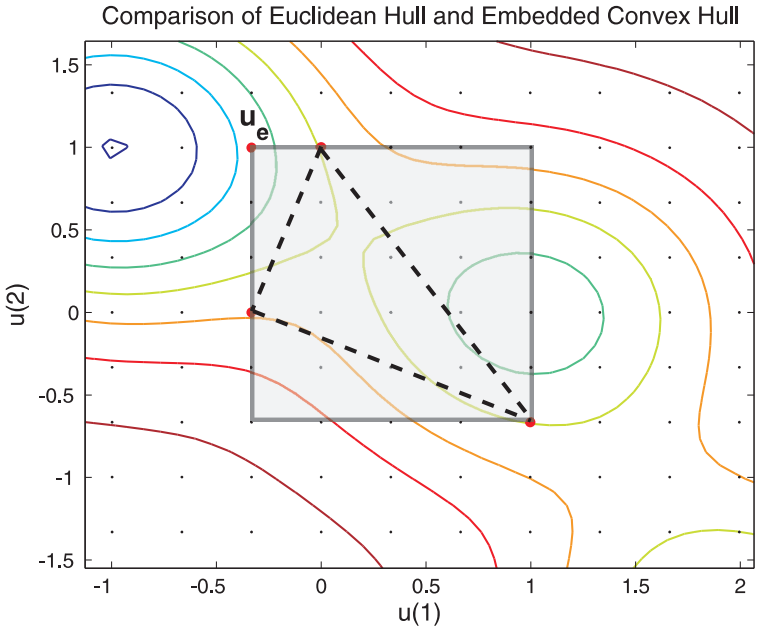

FIGURE 6.16: Comparison of Euclidean convex hull and discrete space embedded convex hull. The shaded grey area is the discrete embedded convex hull and the Euclidean hull is depicted by the dashed trace.

of the points in the embedded convex hull are unreachable through use of the nearest-neighbour parameter update, for example $\mathbf{u}_e$. By augmenting the population with a set of new points, such that the overall population forms a Euclidean hull that covers the embedded hull, all points become reachable. An alternative discrete space parameter update stage and corresponding Embedded Hull RWBS (EHRWBS) algorithm operates in the following manner: 1) identify the embedded discrete convex hull defined by the current population; 2) find a sparse set of new points such that the overall population creates a convex hull in Euclidean space which completely covers the discrete embedded hull; 3) perform the nearest-neighbour convex combination algorithm on the resulting population (retaining P_s population members after replacement).

Generating a sparse set of points to enclose the discrete embedded hull can be achieved in various ways. A simple method exploits the fact that the embedded convex hull will always be a n-dimensional hyper-cuboid. To find a set of points which form a Euclidean hull equivalent of the embedded hull, all that is required is to find the intersection of the lines defined by the minimum and maximum points in each decision variable dimension. The number of points required to represent this hull is $n+1$, although in many cases the number of points that needs to be added to the population, m, will be less than $n+1$, as some of the nodes of the hyper-cuboid will already be members of the population.

The resulting EHRWBS algorithm is, like NNRWBS, identical to the original RWBS algorithm apart from the parameter update step. The EHRWBS update step proceeds as follows:

- *Step 2*: EHRWBS Parameter Update

1. Identify the minimum and maximum points in all dimensions.
2. Identify the m nodes of the embedded hull hyper-cuboid that are not already members of the population, and augment the population with these points creating an intermediate population of size $P_s + m$.
3. Construct the $(P_s + 1)$-th point - first generate the intermediate point using:

$$\mathbf{v}_1 = \sum_{i=1}^{P_s+m} \delta_i(t)\mathbf{u}_i^{(g)}, \tag{6.20}$$

   then assign the new point to the discrete grid using:

$$\mathbf{u}_{P_s+1} = \arg\min_{\mathbf{u}\in U} ||\mathbf{u} - \mathbf{v}_1||. \tag{6.21}$$

4. Construct the $(P_s + 1)$-th point using the formula

$$\mathbf{v}_2 = \mathbf{u}_{best} + (\mathbf{u}_{best} - \mathbf{v}_1). \tag{6.22}$$

   then assign the new point to the discrete grid using:

$$\mathbf{u}_{P_s+2} = \arg\min_{\mathbf{u}\in U} ||\mathbf{u} - \mathbf{v}_2||. \tag{6.23}$$

5. Compute the cost function values $J(\mathbf{u}_{P_s+1})$ and $J(\mathbf{u}_{P_s+1})$ for these two points, and find:

$$i_* = \arg\min_{i=P_s+1,P_s+2} J(\mathbf{u}_i). \tag{6.24}$$

6. The pair $(\mathbf{u}_{i_*}, J(\mathbf{u}_{i_*}))$ then replaces $(\mathbf{u}_{worst}^{(g)}, J_{i_{worst}})$ in the population.
7. Remove the m worst remaining points so that the size of the population is P_s.

6.6.3 Benchmark Convergence Experiments

In this section the performance of the NNRWBS algorithm is assessed by comparison with the original RWBS algorithm operating on an equivalent continuous search space. The NNRWBS algorithm is applied using a regular discrete search space with feasible points centered on multiples of 0.01 in both dimensions and uses the same parameter

values as the RWBS simulation. Figure 6.17 shows the results of this comparison on the simple multi-modal function defined in Equation (6.14) and Figure 6.18 shows the results generated using the Ackley function (6.12). In both cases the performance of NNRWBS is very similar to that of the continuous space algorithm indicating that the modified parameter update step offers a reasonable approach to apply RWBS in discrete space cases. In addition, if the search space is regular then the quantisation step in Equation (6.16) can be computed relatively easily.

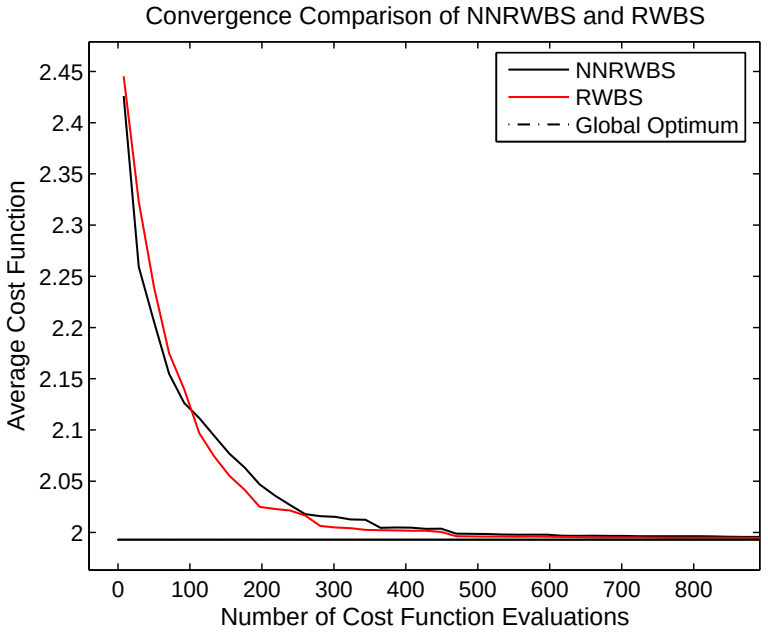

FIGURE 6.17: NNRWBS and RWBS convergence comparison results on test problem (6.14) - Simple Multi-Modal function.

Application in mixed cases can be achieved by applying the NNRWBS parameter update step in the appropriate dimensions, whilst retaining the original update step in any continuous dimensions.

6.6.4 Computational Complexity Analysis

The computational complexity of NNRWBS only differs from that of conventional RWBS due to the extra quantisation step within each inner loop (and only in discrete dimensions). If the structure of the discrete space is known then the quantisation can, in some cases, be performed relatively easily. For instance in a regular Euclidean space, a simple rounding operation is sufficient. In such cases, the extra computation involved is a constant, and the overall algorithmic complexity remains as $\mathcal{O}(N_g N_b P_s)$. In general, however, the complexity will depend on the difficulty involved in assigning the intermediate points to a feasible point. If a highly irregular search space is used, a grid-based look up table may be used to identify candidate feasible points and an exhaustive search employed to select the new point.

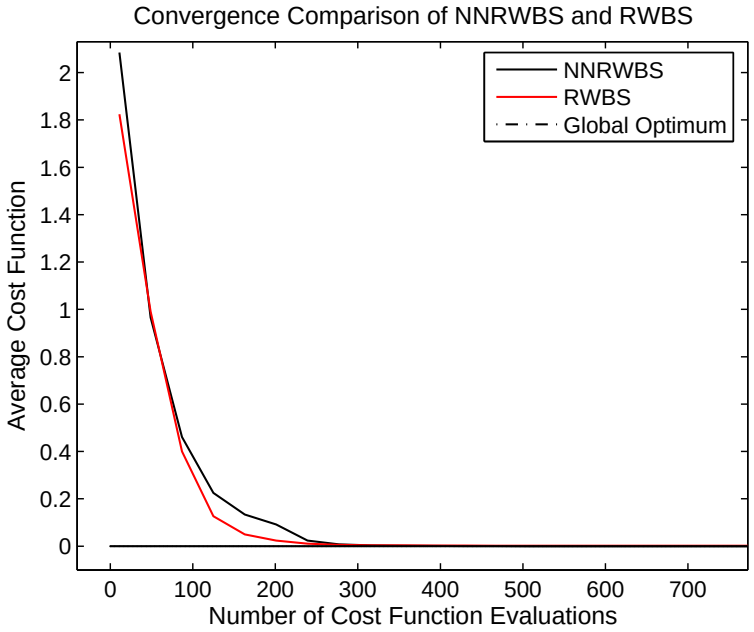

FIGURE 6.18: NNRWBS and RWBS convergence comparison results on test problem (6.12) - Ackley's function.

6.7 Constrained Weighted Boosting Search

There have been a wide range of studies investigating methods for incorporating constraints into stochastic search processes, in particular for evolutionary algorithms [88], [89]. One of the simplest methods to handle constraints in stochastic search technique is to simply reject any infeasible solutions which are generated (this is sometimes known as the death-penalty method). The most popular approach, however, is the application of less drastic penalty functions [90]. These techniques transform the original constrained problem into an unconstrained problem, where the penalty function is added to the original cost function. Sometimes, the penalty is made proportional to the total constraint violation [88]. Various kinds of penalty functions have been investigated, and they can be broadly classified according to whether they are static or dynamic, and whether they are adaptive or non-adaptive. Dynamic penalty functions reduce the effect of the constraints as the optimisation process evolves. The main criticism of the penalty function approach is that it is very difficult to tune the penalty functions. However, the authors of [89] have shown that by searching both the original decision space and an associated Lagrange-multiplier subspace, that some of the problems associated with tuning penalty functions can be avoided.

Both the death-penalty and penalty function methods can be used directly with RWBS. However, it is noted that experimentation suggests that the reflection operator in the algorithm often yields infeasible solutions, and thus the question arises as to how to deal with such points, since simply deleting them questions the fundamental basis of the algorithm. An alternative approach is to use a repair method, whereby infeasible

points are assigned a value in decision space which places them at the boundary of the constraint. Application of such methods is the subject of future work.

6.8 Pareto-Repeated Weighted Boosting Search

It was shown in Chapter 4 that if a priori information regarding the relative importance of different optimisation objectives is available, the multiple-objective optimisation problem can be reformulated as a single-objective problem (such as in the simple weighting method). Techniques which operate in this context are often termed 'Non-Pareto Methods' as they search for solutions to surrogate problems. However, if preference information is not available, or indeed the nature of the Pareto-frontier is of direct interest, then a useful optimisation algorithm must generate a *set* of solutions; more specifically a set of Pareto-optimal solutions. Ideally, the solutions should be well distributed across the Pareto-frontier[7]. These methods are normally termed 'Pareto Methods'. The solution set can then be used to consider which solution is most appropriate for the particular problem (and thus implicitly infer some relative importance of the objectives). Several methods have been proposed to adapt common population based stochastic search techniques, in particular GAs, to generate Pareto-optimal sets. An excellent introduction to this literature can be found in [80].

There are two main aspects to designing an efficient algorithm for Pareto-optimisation. Firstly, the algorithm needs to embody a mechanism which drives solutions towards the Pareto-frontier, and secondly, there needs to be a mechanism which ensures that there is a good distribution of solutions across the frontier. Typically, a form of Pareto-ranking or Pareto-sorting is used to guide the optimisation towards the frontier [80]. These techniques effectively modify the cost value or fitness value for a solution depending on its performance *relative* to other solutions in the set (in contrast to the absolute notion of optimality used in conventional optimisation). Solutions which are non-dominated or mildly dominated (i.e. only dominated by a limited number of other solutions) are attributed a higher fitness or lower cost than those which are strongly dominated. This promotes the generation of more non-dominated solutions. Distribution of solutions across the Pareto-frontier is commonly achieved using 'sharing' or 'niche methods' [80], [73]. Sharing methods distribute an individual's fitness depending on how many solutions are nearby it[8], thus encouraging spread and avoiding the problems associated with genetic drift. The difficulty with sharing techniques is that the user must define the so-called 'sharing parameter' [80]. In general, manual fixing of the sharing parameter requires knowledge of the objective function and adds to the tuning complexity of the optimisation algorithm. However, the authors of [70] has shown that if the sharing process

[7] What is meant by 'well distributed' in this context is often problem specific but a reasonably uniform distribution over the Pareto-front may often be suitable.

[8] Sharing can take place either in the decision space or the fitness space, although in some cases decision space sharing is preferable [91].

is considered from a density estimation viewpoint, the sharing parameter is analogous to a smoothing parameter, and thus the sharing process can be automated by modelling the population density using a technique such as Parzen Window density estimation [92]. The authors further show that with appropriate choice of kernel function, it is possible to create a parameterless sharing process. It should be noted, however, that the density estimator uses recommended heuristics for its internal parameters. In contrast, a distance based measure is used in [79] which is completely parameterless.

As RWBS is a population-based method, it can be readily adapted to the Pareto case. Aside from the addition of a Pareto-ranking process and a mechanism which encourages distribution, there are a number of additional modifications required. These include modifying the elitism process so that a larger set of solutions is retained.

6.8.1 Modified Elitism

In order to identify a suitable set of Pareto-optimal solutions, a record of potential solutions must be retained during each generation. To achieve this, the elitism process enforced by Equation (6.3) is extended so that a larger proportion of the current population is kept between each generation. This introduces a new parameter, P_e, the 'elitism count', which specifies how many population members are kept between generations.

6.8.2 Pareto-Ranking, Distribution, and Cost Mapping

All population members are ranked relatively in terms of Pareto-dominance according to the 'Fast-Non-Dominated-Sort' procedure proposed in [79]. To encourage a good spread across the Pareto-frontier, the resulting Pareto-rank, R_i, is then re-adjusted according to a scaling parameter, P_r, and the mean distance from all other points, D_i:

$$\hat{J}_i = \frac{P_r R_i}{D_i}, 1 \leq i \leq P_s \tag{6.25}$$

6.8.3 Pareto-Repeated Weighted Boosting Search Algorithm

The proposed Pareto-RWBS algorithm is constructed as follows. Specify the following algorithmic parameters: P_s - population size; N_g - number of generations in the repeated search; N_B - number of iterations in the weighted Boosting search; ξ_B - accuracy for terminating the weighted Boosting search, P_r - Pareto-ranking scaling, P_e - Elitism count.

- **Outer Loop : generations** For $g = 1 : N_g$

- – *Pareto Generations initialisation*: Initialise the population by setting $\mathbf{u}_{1:P_e}^{(g)} = \mathbf{u}_{1:P_e}^{(g-1)}$ and randomly generating rest of the population members $\mathbf{u}_{P_e:P_s}^{(g)}, P_e \leq i \leq P_s$ where $\mathbf{u}_{1:P_e}^{(g-1)}$ denotes the best P_e solutions found in the previous generation. **If** $g = 1$**,** $\mathbf{u}_{1:P_e}^{(g)}$ are also randomly chosen.
  – Weighted Boosting search initialisation: Assign the initial distribution weights $\delta_i(0) = \frac{1}{P_s}, 1 \leq i \leq P_s$ for the population.

- **Inner Loop : weighted Boosting search** For $t = 1 : N_B$

- *Step 1*: Pareto Boosting

1. Calculate the cost function values of each point and for each objective[9]

$$J_{i,o} = J_o\left(\mathbf{u}_i^{(g)}\right), 1 \leq i \leq P_s, 1 \leq o \leq N_o$$

   where N_o is the number of objective functions.

2. Calculate the Pareto-rank for each member of the population

$$R_i = FastNonDominatedSort(J), \tag{6.26}$$

   using the method proposed in [79].

3. For each member of the population, compute the mean Euclidean distance to all other points in decision space:

$$D_i = \frac{1}{P_s}||\mathbf{u}_i^{(g)} - \mathbf{u}_j^{(g)}||, 1 \leq i \leq P_s, i \neq j \tag{6.27}$$

   and use it to compute the distance and rank adjusted cost:

$$\hat{J}_i = \frac{P_r R_i}{D_i}, 1 \leq i \leq P_s. \tag{6.28}$$

4. Normalise the adjusted cost values:

$$\bar{J}_i = \frac{\hat{J}_i}{\sum_{j=1}^{P_s} \hat{J}_j}, 1 \leq i \leq P_s.$$

5. Compute a weighting factor $\beta(t)$ according to

$$\eta(t) = \sum_{i=1}^{P_s} \delta_i(t-1)\bar{J}_i, \quad \beta(t) = \frac{\eta(t)}{1-\eta(t)}.$$

[9]Note that the cost function evaluation has moved into the inner loop because it must be re-evaluated for every boost as the evaluation is a relative measure rather than an absolute measure.

6. Update the distribution weights for $1 \leq i \leq P_s$

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta(t)^{\bar{J}_i} & \text{for; } \beta(t) \leq 1 \\ \delta_i(t-1)\beta(t)^{1-\bar{J}_i} & \text{for; } \beta(t) \leq 1 \end{cases}$$

and normalise them:

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{j=1}^{P_s} \delta_i(t)}, 1 \leq i \leq P_s.$$

- *Step 2*: Pareto Parameter Update

1. Construct the $(P_s + 1)$-th point using the formula

$$\mathbf{u}_{P_s+1} = \sum_{i=1}^{C} \delta_i(t)\mathbf{u}_i^{(g)}.$$

2. Construct the $(P_s + 1)$-th point using the formula

$$\mathbf{u}_{P_s+2} = \mathbf{u}_{best}^{(g)} + (\mathbf{u}_{best}^{(g)} - \mathbf{u}_{Ps+1}).$$

3. Compute the new cost function values, $J_{i,o}$, Pareto-rankings, R_i, distance estimates, D_i, and distance and rank adjusted cost, $\bar{J}_i$, for each member of the augmented population $\tilde{\mathbf{u}} = [\mathbf{u}', \mathbf{u}'_{P_s+1}, \mathbf{u}'_{P_s+2}]'$, and find:

$$i_w = \arg \max_{1 \leq i \leq P_s+2} \bar{J}_i.$$

4. The pair $(\mathbf{u}_{i_w}, J_{i_w})$ is then removed from the population.

- – End of **inner loop**
- **End of outer loop** This yields the solution set $\mathbf{u}^{(N_g)}$

6.8.4 Benchmark Convergence Experiments

In order to evaluate the performance of Pareto-RWBS, its performance is compared with the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) on a number of test problems. NSGA-II algorithm is a well known state of the art multiple objective algorithm which has been shown to produce very good results on a wide range of problems [79]. The NGSA-II implementation used here utilises real-coding, binary tournament selection, binary crossover with probability 0.9, polynomial mutation with probability $1/n$ (where n is the number of dimensions), and non-dominated sorting in conjunction with a crowding operator.

The following examples are the results of individual simulations rather than multiple MC simulations. The first test function that is analysed is the 'SCH' function taken from [79], which exhibits a simple convex Pareto-frontier:

$$\begin{aligned} J_1(u) =& u_1^2 \\ J_2(u) =& (u_1 - 2)^2. \end{aligned} \tag{6.29}$$

The decision variable in this case can lie in the interval $\{-1 : 1\}$. The following settings were used for Pareto-RWBS; $P_s = 25$ $N_b = 10$, $N_g = 100$, $P_e/P_s = 0.8$, $P_r = 10$ and $\xi_b = 0.05$. These parameters were found to produce the best results based on trial and error. The settings used for NSGA-II were a population size of 30 and 50 generations. As with the Pareto-RWBS settings, these were tuned using trial and error to provide the best performance. It is interesting to note here that good performance is observed with a population size of approximately two to three times the number of dimensions, in contrast to the settings which were found to yield high performance for the single-objective GA implementation used earlier in this chapter. The results for this test function are illustrated in Figure 6.19, which shows the resulting objective space solutions. In this diagram, and the remainder of the objective space diagrams in this section, red markers indicate the feasible solutions generated by multiple MC simulations based on random sampling in the decision space (these also help to visually locate the Pareto-frontier), blue markers indicate the candidate solutions generated by NSGA-II, and black markers are the Pareto-RWBS candidate solutions. It is noted

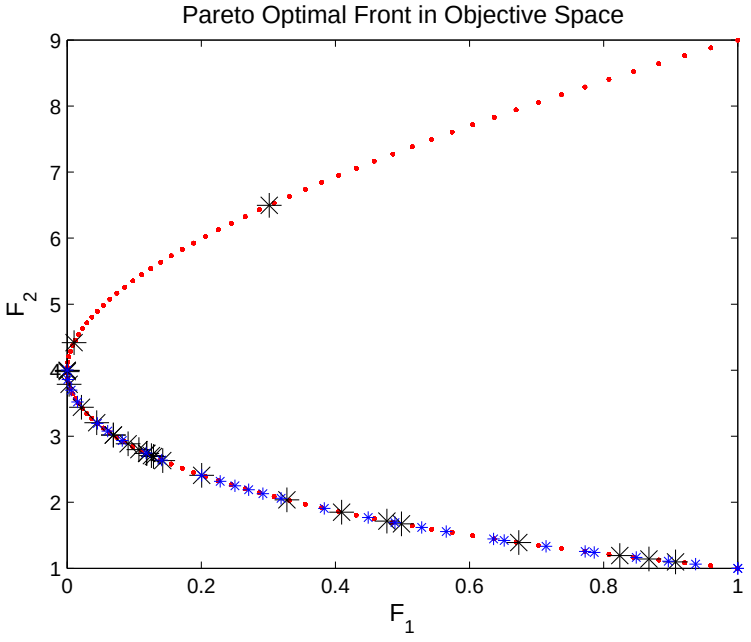

FIGURE 6.19: Objective space performance comparison of NSGA-II and Pareto-RWBS on the convex test problem in (6.29). Red markers indicate the feasible solutions (generated by MC simulation), blue markers indicate the candidate solutions generated by NSGA-II, and black markers are the Pareto-RWBS candidate solutions.

immediately that the Pareto-RWBS algorithm is capable of finding solutions across the Pareto-frontier. However, the distribution of the solutions is sub-optimal with respect to the NSGA-II results in that it is less uniform.

The next test function, 'KUR', is again taken from [79], and is an example of a problem where the Pareto-frontier is non-convex:

$$J_1(u) = \sum_{i=1}^{n-1}(-10\exp^{(-0.2\sqrt{u_i^2+u_{i+1}^2})}) \tag{6.30}$$

$$J_2(u) = \sum_{i=1}^{n}(|u_i|_a^{0.8} + 5\sin u_i^3). \tag{6.31}$$

In this case there are two decision variable dimensions, and the decision variables can lie in the interval $\{-5:5\}$. The following settings were used for Pareto-RWBS; $P_s = 25$, $N_b = 10$, $N_g = 100$, $P_e/P_s = 0.8$, $P_r = 10$, and $\xi_b = 0.05$. The population size and number of generations for NSGA-II were 30 and 50 respectively. As in the first test problem, these parameters were chosen through trial and error. The results for this test function are illustrated in Figures 6.20, 6.21 and 6.22. Similar results are observed with

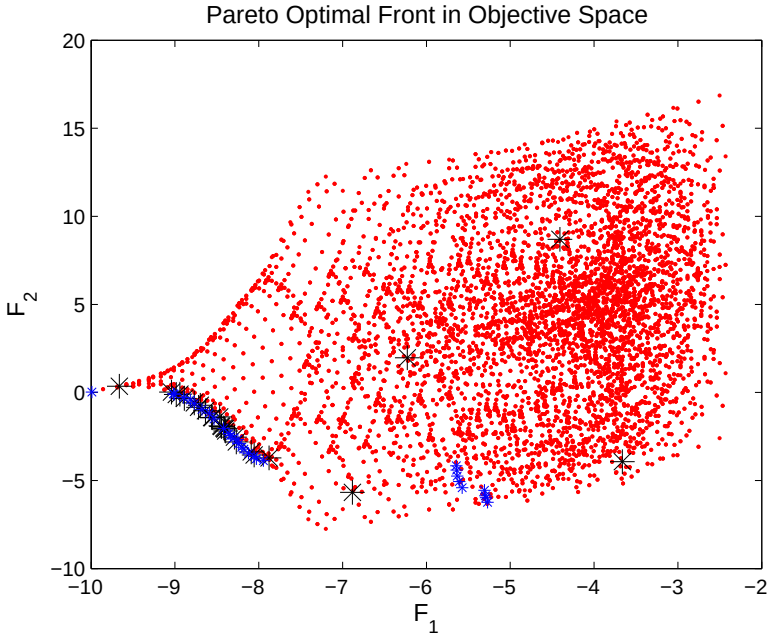

FIGURE 6.20: Objective space performance comparison of NSGA-II and Pareto-RWBS on the non-convex test problem in (6.30). Red markers indicate the feasible solutions (generated by MC simulation), blue markers indicate the candidate solutions generated by NSGA-II, and black markers are the Pareto-RWBS candidate solutions.

this test function; both the NSGA-II algorithm and the Pareto-RWBS algorithm are focussed on the same convex region of the Pareto-frontier. A close-up of the objective space in the region where the majority of the solutions are located, is illustrated in Figure

6.21. This reveals that the Pareto-RWBS algorithm approaches the Pareto-frontier successfully and is distributed across a similar region as the NSGA-II candidate solutions. The distribution across this range, however, is somewhat poorer (i.e. less uniform), as observed with the 'SCH' function. The decision variable space results, illustrated in

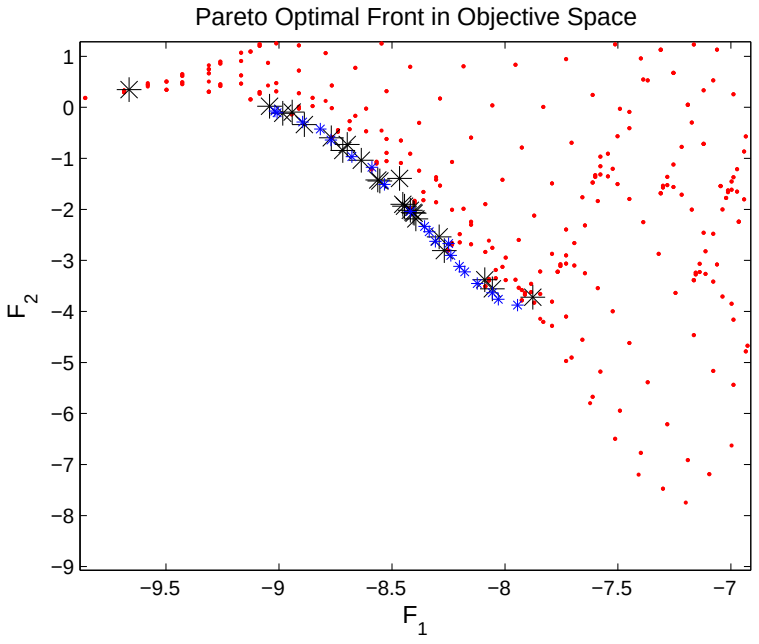

FIGURE 6.21: Close-up of objective space performance comparison of NSGA-II and Pareto-RWBS on the non-convex test problem in (6.30). Red markers indicate the feasible solutions (generated by MC simulation), blue markers indicate the candidate solutions generated by NSGA-II, and black markers are the Pareto-RWBS candidate solutions.

Figure 6.22, and all subsequent decision variable space diagrams in this section can be interpreted as follows: the overlaid contour function represents the objective functions; blue markers indicate NSGA-II candidate solutions; and, red markers indicate Pareto-RWBS candidate solutions. Figure 6.22 indicates that the Pareto-optimal solutions lie in a very small region of the decision variable space and that the Pareto-RWBS algorithm has identified a very similar region to NSGA-II. However, the Pareto-RWBS solutions are slightly more spread out in the decision space indicating that the Pareto-ranking and cost adjustment process could be improved.

Performance in cases where the Pareto-frontier is multi-modal is examined using a test function adopted from [91]:

$$J_1(u) = u(1) \tag{6.32}$$

$$g(u(2)) = 2.0 - \exp^{-\left(\frac{u(2)-0.2}{0.004}\right)^2} - 0.8\exp^{-\left(\frac{u(2)-0.6}{0.4}\right)^2}$$

$$J_2(u) = \frac{g(u(2))}{u(1)}. \tag{6.33}$$

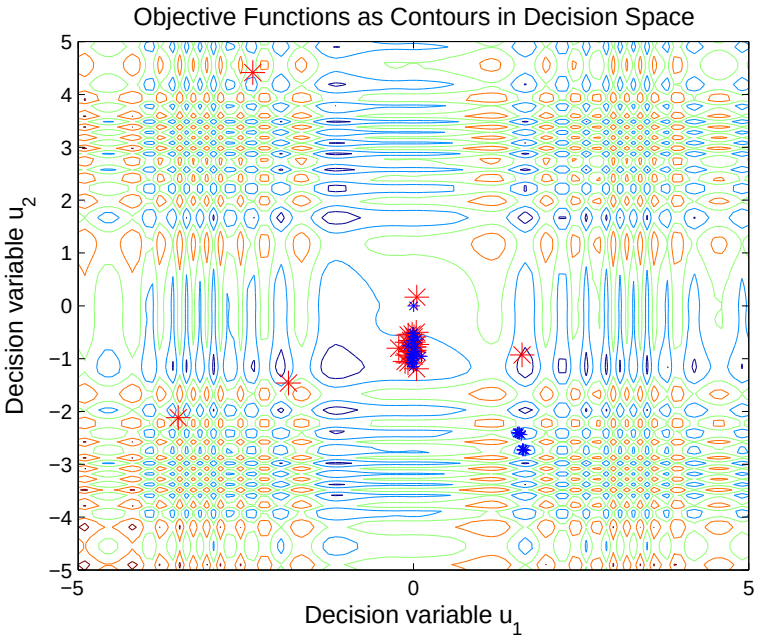

FIGURE 6.22: Decision space comparison of NSGA-II and Pareto-RWBS on the discontinuous test problem in (6.30). The overlaid contour function represents the objective functions, blue markers indicate NSGA-II candidate solutions, and red markers indicate RWBS candidate solutions.

The decision variables can lie in the interval $u(1) = \{0.1 : 1\}$ and $u(2) \in \{0, 1\}$. As in earlier examples, the following settings were used for Pareto-RWBS; $P_s = 25$ $N_b = 10$, $N_g = 100$, $P_e/P_s = 0.8$, $P_r = 10$ and $\xi_b = 0.05$. The settings used for NSGA-II were a population size of 30 and 50 generations. The results for this test function are illustrated in Figures 6.23 and 6.24. This cost function has multiple modes, an attribute which is known to cause difficulties for many multiple-objective optimisation methods. In this case, the RWBS-based algorithm demonstrates the ability to identify a range of modes, and in some regions of the frontier, outperforms the NSGA-II algorithm. Once again, a reasonable area of the Pareto-frontier is identified, but the distribution is less uniform than with NSGA-II. The relative positions of the candidate solutions in decision variable space, as depicted in Figure 6.24, are not as informative in this case and it is difficult to infer any insight into the operation of Pareto-RWBS or NSGA-II from them. However, armed with a priori knowledge regarding the true Pareto-optimal region of the decision space, it may be possible to gain a deeper understanding, and this is an interesting area for future research. The following two-dimensional test function is an example of a problem where the Pareto-frontier is discontinuous [91]:

$$\begin{aligned} J_1 &= u(1) \\ g(u(2)) &= 1 + 10u(2) \\ J_2 &= g(u(2)) \left[1 - \left(\frac{J_1}{g(u(2))} \right)^{\alpha} - \frac{J_1}{g(u(2))} \sin(2\pi q J_1) \right], \end{aligned} \tag{6.34}$$

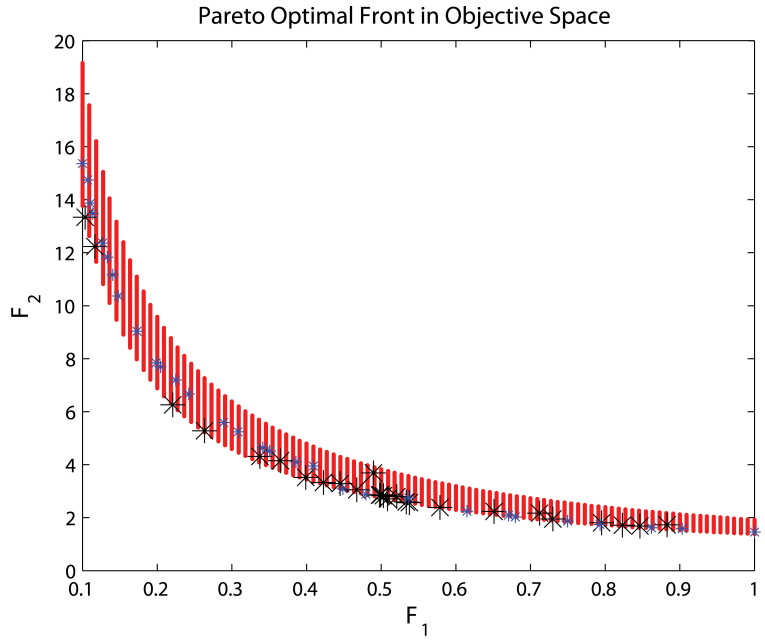

FIGURE 6.23: Objective space performance comparison of NSGA-II and Pareto-RWBS on the multi-modal test problem in (6.32). Red markers indicate the feasible solutions (generated by MC simulation), blue markers indicate the candidate solutions generated by NSGA-II, and black markers are the Pareto-RWBS candidate solutions.

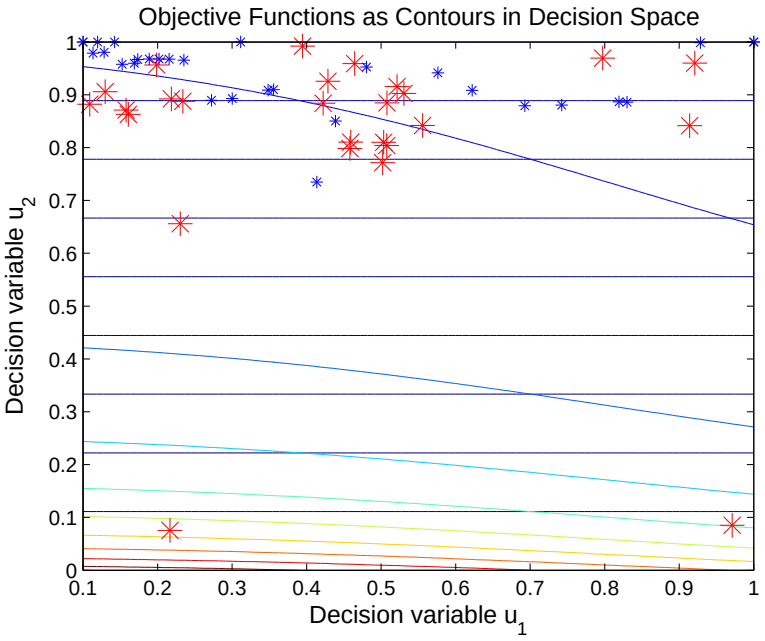

FIGURE 6.24: Decision space comparison of NSGA-II and Pareto-RWBS on the multi-modal test problem in (6.32). The overlaid contour function represents the objective functions, blue markers indicate NSGA-II candidate solutions, and red markers indicate RWBS candidate solutions.

where, in this case, $\alpha = 2$, $q = 4$, and the decision variables both lie in the interval $\{0 : 1\}$. The following settings were found to provide the best results for Pareto-RWBS; $P_s = 50$, $N_b = 20$, $N_g = 100$, $P_e/P_s = 0.8$, $P_r = 10$, and $\xi_b = 0.05$. NSGA-II was

simulated using the same settings as in the previous problems. The results for this test function are illustrated in Figures 6.25 and 6.26. In relation to Pareto-RWBS, this

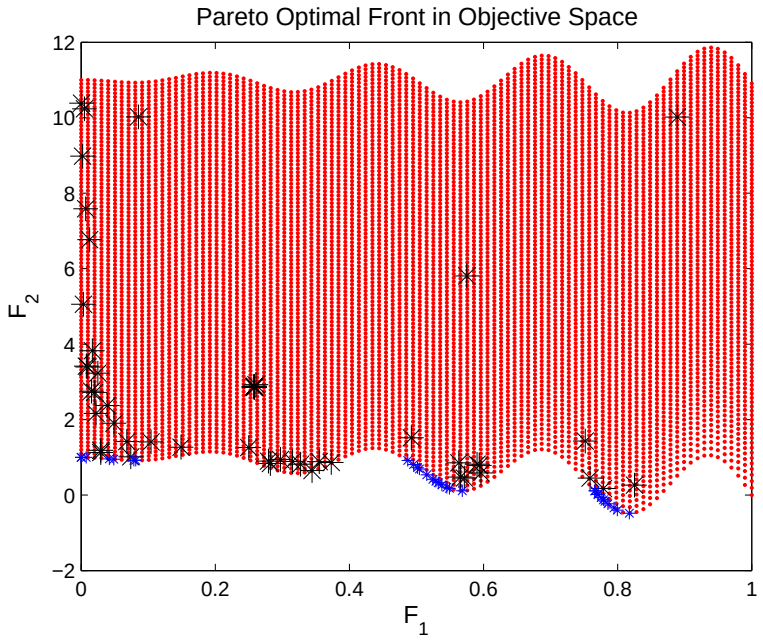

FIGURE 6.25: Objective space performance comparison of NSGA-II and Pareto-RWBS on the discontinuous test problem in (6.34). Red markers indicate the feasible solutions (generated by MC simulation), blue markers indicate the candidate solutions generated by NSGA-II, and black markers are the Pareto-RWBS candidate solutions.

particular problem required a larger population size and a larger number of boosts per iteration, most likely due to the challenging nature of the problem. This cost function has a discontinuous Pareto-frontier, a further attribute which is known to challenge multiple-objective optimisation techniques. It is observed that Pareto-RWBS converges towards four of the primary Pareto-optimal regions, in comparison with the NSGA-II algorithm which only identifies three of the regions in this particular simulation. The performance of the NSGA-II algorithm within the located regions is, however, superior to that of Pareto-RWBS as in earlier test cases. Figure 6.26 offers a similar intuition as to the performance of Pareto-RWBS as the decision space illustration for the discontinous test problem. Pareto-RWBS is observed to identify a large area of the Pareto-frontier (in multiple modes in this case), but the solutions are located further from the frontier than the NSGA-II solutions. This further justifies the assertion that the Pareto-RWBS exhibits promising general performance characteristics, but that the Pareto-ranking and cost mapping method could be improved in order to exploit them.

In summary, the Pareto-RWBS algorithm proposed in this chapter demonstrated clear potential for being developed into a flexible, high-performance multiple-objective optimisation technique. The algorithm has been shown to converge reliably towards the Pareto-frontier in a range of test problems with various challenging attributes. The

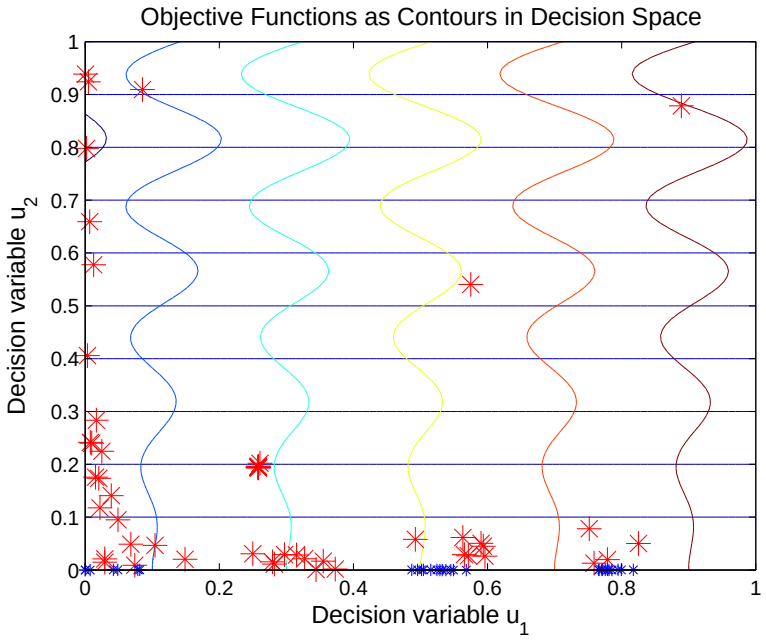

FIGURE 6.26: Decision space comparison of NSGA-II and Pareto-RWBS on the discontinuous test problem in (6.34). The overlaid contour function represents the objective functions, blue markers indicate NSGA-II candidate solutions, and red markers indicate RWBS candidate solutions.

algorithm was observed to identify a large area of the Pareto-frontier in each case, comparable with NSGA-II, a well known state of the art GA variant, and provided superior performance in terms of the localisation of discontinuous regions of the Pareto-frontier. There is scope, however, for the algorithm to be improved both in terms of the distribution of its solutions along the Pareto-frontier, and the accuracy of the solutions in terms of their distance to the Pareto-frontier. The convergence accuracy hinges on the Pareto-ranking process used in this particular Pareto-RWBS variant, and the impact it has on the performance of the convex combination operator. A method for improving the performance of Pareto-RWBS in this regard is outlined in the following section.

6.8.5 Selective Combination

As Pareto-RWBS generates new members through the convex operator in Equation (6.2), the standard approaches to Pareto-ranking procedures, such as the one used in this chapter, may be sub-optimal. For instance, ranking methods which simply assign all non-dominated candidates a similar rank will usually work reasonably well with a GA as the combination operator will simply select a number of members of the population to combine together and the rank value is not used directly in the local search operator (i.e. during crossover). RWBS, however, *weights* the combination according to rank, so all members of the same front will receive equal weighting in the combination (ignoring any sharing or other distribution related mechanisms). The efficiency of the convex operator may, therefore, be reduced, which suggests that a larger number

of boosts have to be used per generation. Additionally, RWBS only generates a single new member in each inner loop Boosting stage, unlike a GA, which can create several new members at the local search stage. With population sizes of the order required for many multiple-objective optimisation problems, a large number of boosts may be required. An alternative approach to that outlined above, therefore, is to use a selection operator to select which members are used in a *set* of convex combinations at each stage (similar to the way a GA proceeds). This would create a number of new individuals in each generation as well as reducing the number of solutions in each Boosting stage, thus reducing the required number of boosts. It is hypothesised that this approach would help to improve the algorithms performance in terms of the accuracy with which the Pareto-frontier is localised.

6.9 Concluding Remarks

This chapter has presented a number of extensions to the RWBS optimisation algorithm in order to facilitate its use in a wider class of optimisation problems. The extensions permit the use of RWBS in problems that have discrete or mixed search spaces, and multiple-objectives. In the discrete case, it was shown that a simple extension to the convex operator renders the algorithm capable of successful optimisation over discrete search spaces.

Significant benchmark convergence experiments were presented, which demonstrate the algorithms respectable performance across a range of well-known test problems, and a number of parameter sensitivity experiments provide indications for tuning procedures. The convergence experiments and the tuning analysis help to strengthen confidence in the algorithm's performance and operation, therefore rendering it more appropriate for application to real-world optimisation problems.

Multiple-objective problems were addressed by integrating a Pareto-ranking scheme and a sharing process into the RWBS algorithm. The resulting Pareto-RWBS algorithm was shown to be capable of finding Pareto-optimal solution sets in a variety of test problems, and was observed to offer superior performance over a GA variant on a problem with a discontinuous Pareto-frontier. All of the extensions proposed in this chapter retain the attractive properties of the original RWBS algorithm outlined in [81], i.e. simplicity, ease of implementation, and a small number of tuning parameters. At the same time the extensions facilitate its application to a more general class of optimisation problems. This contributes towards making RWBS a powerful tool for solving general complex optimisation problems with minimal tuning and programming effort.

In addition, the research presented in this chapter further justifies the applicability of the algorithm to the generic AV SM case due to the performance and flexibility of the algorithm, and, in particular, its insensitivity to some tuning parameters.

Chapter 7

Conclusions and Future Work

This chapter comprises a critical summary of the research presented in this thesis. The principal contributions of the research are summarised and analysed in Section 7.1. The theoretical and practical significance of the work, including aspects of how and when the contributions may have an impact on real systems and other fields, is analysed in Section 7.2. Conclusions regarding the importance of various facilitating assumptions are drawn in Section 7.3. Finally, a number of routes for future work are proposed and explored in Section 7.4, for each of the primary contributory areas.

7.1 Summary of Contributions

This thesis has presented a number of contributions to the SM and optimisation fields. More specifically, the contributions relate to the manner in which observer control planning problems are formulated, and subsequently solved. The primary contributions deriving from this research are summarised as follows:

Chapter 4: Optimal Observer Trajectory Optimisation

This chapter focused on a number of practical aspects relating to observer trajectory control. The problem was formulated as a novel, multiple-objective optimisation problem with constraints. A number of methods to enforce preference relations were considered, and it was shown that, if a predefined performance requirement for state estimation can be derived, this proves a more natural approach to framing the optimisation problem than using objective weights. These contributions are summarised as:

- a demonstration of the relationship between the changes in sensor-to-target geometry induced by information gathering objectives and secondary objectives relating to sensor survivability;

- an analysis of the use of different multiple-objective optimisation formulations for UGV observer trajectory optimisation;
- and, a novel UGV observer control algorithm based on the minimax approach which permits the optimisation of secondary system objectives while simultaneously maintaining a predefined state estimation performance level.

Chapter 5: Non-Myopic Sensor Control

Chapter 5 concerned the development of non-myopic SM control strategies. Due to the computational complexities associated with these sequential decision making problems, they are typically solved using sub-optimal or approximate control strategies, such as limited-lookahead control. It was shown that, in the general case, the performance of such strategies is linked to the length of the lookahead horizon, for two primary reasons. Firstly, it is well-known that increasing the length of the horizon reveals further information about the substructure of the decision space. However, Chapter 5 demonstrated that there is a further mechanism which limits performance. This mechanism is the propagation of error through the control loop, and it was shown that increasing the lookahead can degrade performance. This effect was expressed formally for the first time, and led to the development of a novel adaptive control strategy based on a dynamic lookahead horizon. An approach for computing the optimal horizon length was proposed, based on a novel Non-Myopic Risk Equilibrium, which identifies the horizon which balances these two mechanisms.

The primary contributions of this chapter are summarised below:

- an analysis of the propagation of uncertainty in a predictive Sensor Management feedback-loop;
- the identification of a multitude of error sources which can contribute to the propagation of error, and thus planning performance;
- and, an approach to developing a novel adaptive horizon control strategy which improves on the performance of classical limited-lookahead control by increasing robustness to uncertainty.

Chapter 6: Repeated Weighted Boosting Search for Optimisation

Many SM problems resolve into complex optimisation problems which present difficulties to traditional gradient-based optimisation techniques. This is especially true when sequential decision-making strategies, such as those discussed in Chapter 5 are utilised. As a result, such problems are often solved using stochastic search methods such as GAs. The RWBS algorithm is a recently proposed alternative guided-search algorithm which offers a number of attractive features such as ease of implementation and simplicity of

tuning. These attributes make it an interesting candidate for use in AV systems for a range of different problems. However, the algorithm, in its original form, is only capable of being applied to a limited class of problems. Chapter 6 presents the results of a number of investigations in the possibility of extending the RWBS algorithm to a wider class of optimisation problems, such as those involved in SM studies.

The resulting algorithms demonstrate promising performance in terms of convergence performance and flexibility, and retain the attractive features of the original algorithm. The contributions presented in Chapter 6 are summarised below:

- an empirical convergence comparison of Repeated Weighted Boosting Search, multiple-start gradient-based descent, random search and a Genetic Algorithm;
- an analysis into the tuning parameter sensitivity of Repeated Weighted Boosting Search;
- a Repeated Weighted Boosting Search optimisation algorithm that is capable of optimising over mixed search spaces;
- a Pareto multiple-objective optimisation algorithm based on Repeated Weighted Boosting Search;
- and, a comparative performance analysis of the Pareto-RWBS algorithm and a state of the art GA equivalent.

7.2 Significance of Research

An important aspect of research driven by real-world problems is assessing the practical significance of derived results, and analysis of the way in which the algorithms and understanding can be integrated into real systems. In this section, the theoretical contributions presented in this thesis are examined according to their practical significance at the current time, and in the future, with reference to the other scientific disciplines to which they are relevant.

The control algorithms presented in Chapter 4 are based on a pragmatic, multiple-objective view of the SM problem for AVs. As discussed in Chapter 1, the AV field is still relatively immature in terms of autonomous behaviour (but not in terms of the the physical design of the platforms). As the AV field matures, more and more systems will emerge, and it is at this point that the practical issues relevant to the contributions in Chapter 4 will come to light. In particular, the use of power-limited AVs for tracking in a 'deploy and forget' context will become increasingly important, especially in the urban warfare environment and anti-terrorism scenarios. However, there is clearly a wealth of additional work involved in implementing such algorithms on a real system. The main

obstacle to this process is the difficulty with integrating system-level mission planning processes and the more intricate control processes presented in this work. One important area of focus for the SM field, therefore, is how multiple-layer planning processes can be integrated efficiently.

Multiple-step planning algorithms clearly offer superior performance in many military DF scenarios. Unfortunately, the computational requirements associated with such algorithms can be exponential in the length of the optimisation horizon. Thus the scope for practical implementation of such algorithms is extremely limited. To the author's knowledge, multiple-step planning with horizons greater than a few steps is not currently under test on any practical sensor system which is described in the open domain. This has been the key motivation for the development of ADP and related solutions such as that found in [63]. The approach taken in Chapter 5 of this thesis, is to side-step the computational problems associated with developing sensor control strategies over multiple-steps and to explore the theoretical *limits* of such planning. This approach is motivated by the assumption that computational performance will continue to increase[1]. The analysis in Chapter 5 is, therefore, aimed at answering the question, 'if computational requirements were not an issue, what are the benefits to be gained by long-term non-myopic sensor control strategies, and can these benefits be exploited in a practical system?'. While further work is required to identify the real performance gains in military scenarios (including the existence of so-called 'forecast horizons' - a term used in the control literature to identify a horizon length beyond which the first action becomes independent), the contributions in this thesis have identified, for the first time, one of the primary mechanisms which limit the planning performance, and a suggested method for finding the limit.

It is clear that the understanding derived from this aspect of the current work may not see any practical demonstration in the near future. This is really a reflection on the agility of the defence sector, rather than on the significance of the results. However, it is noted that the theoretical approach taken in this work is also applicable to the *approximate* algorithms described above, and, in fact, may actually be of *more* significance to such systems due to the inherent additional approximations on which methods rely. Thus it is concluded that the results of the contributions in Chapter 5 may become significant in a practical sense, as the SM planning field matures, and the defence market makes increasing use of the theoretical frameworks available in the academic domain. Additionally, the broad applicability of the analysis to other scientific fields such as control, operations research, and finance, suggests that this work may find useful application in other areas.

Chapter 6 presented a number of novel optimisation algorithms based on RWBS. While the primary motivation for their development was to solve the complex multiple-stage

[1] This is driven largely by non-defence markets.

optimisation problems associated with sensor control problems, they are equally applicable to any non-linear optimisation problem, and it is this observation that motivated the choice of test problems that were used to assess their performance. It was demonstrated in Chapter 6 that the performance of the RWBS algorithms is comparable to that of the GA, a tool which has already been established as a flexible optimisation method. The two primary advantages of the RWBS algorithms are their rapid implementation and basic tuning procedure. This promotes easy prototyping and flexible application. As a result, it is hoped that the potential of the RWBS technique will be examined in more depth in the future.

7.3 On Facilitating Assumptions

The application of the research presented in this thesis[2] is based on a set of facilitating assumptions regarding the SM problem. These assumptions are necessary in order to avoid complicating the analysis of the planning algorithms presented herein. The major assumptions are based around simplifications of the state estimation and control problems, and are summarised as follows:

- The data-association and registration problems are solved externally.
- The sensor platform dynamics are deterministic.
- Sensor platform control is instantaneous and perfect.
- Inter-platform communication is instantaneous and perfect.

The first assumption has an impact on the results presented in both Chapter 4 and 5. Broadly speaking, if the data -association problem is taken into account, the performance of the observer trajectory control strategies detailed in this work will degrade, as there will be additional uncertainty involved in finding the optimal sensor action. Of particular note is the effect of the data-association process on the optimality of the DF process, and as such, this will amplify the error propagation effect discussed in Chapter 5. This will render the results more significant for practical applications.

The deterministic nature of the sensor dynamics and control processes assumed in the development presented in Chapter 4 avoided unnecessary complications in analysing the observer control problems. In real systems, however, the sensor positions will be unknown to some degree, and the actuation process will not be perfect, and this will involve additional complications in the target state estimation and optimisation stages. Similar to the assumption regarding data association, the effect of sensor position and

[2]This discussion is primarily relevant to Chapters 4 and 5.

control actuation uncertainty also serves to compound the error propagation process, and further justifies the results presented in Chapter 5.

Finally, the assumption of a fully-connected communication network is also not appropriate for some practical systems, particularly due to the trend towards local communication strategies [2]. Once again, the effect of this assumption is that there will be a degradation in planning performance due to a decrease in state estimation performance; in this case, each node may only be able to fuse the data from other local nodes. As a result, the implications of this assumption yield similar conclusions to those presented above.

7.4 Future Work

Each of the principal areas of contributions outlined in this thesis has a number of associated areas of potential further development. In this section, extensions to the research are considered in regard to Chapters 4 to 6.

7.4.1 Multiple-Objective Sensor Management

One of the most challenging problems that will face the development of practical AV systems, will be finding an approach to balancing a wide range of performance objectives and tasks. These will typically include own-platform Self-Localisation and Mapping (SLAM), searching for new targets, observing, tracking and identifying existing targets, and minimising power consumption and deviation from other required mission objectives. It is now accepted that information-based performance measures provide a single integrated framework within which many of these objectives can be accommodated [45]. It is interesting, therefore, to consider the extension of the work in Chapter 4 to a wider set of both perceptive and non-perceptive optimisation objectives.

A particularly interesting avenue of further work, is to explore the benefits of non-myopic planning to secondary system objectives, particularly power. It was shown in Chapter 4 that some tracking performance objective functions induce changes in the sensor-target geometry that equate to minimising the sensor-to-target range. According to the power model adopted in this work, this is in direct conflict with the requirement to minimise power consumption. However, it is noted that in many cases, the target may be travelling in a direction towards the sensor, and thus there may be some benefit in 'waiting' for the target to approach the sensor, thus saving the need to expend power. This will require a non-myopic sensor control strategy, such as that discussed in Chapter 4. In addition, the error propagation model applied in Chapter 5 could then be extended to account for secondary objectives.

Further future work includes higher fidelity platform power modelling, and the evaluation of goal-based algorithms such as goal attainment [69].

7.4.2 Adaptive Horizon Sensor Management

The results presented in Chapter 5 provide a means to analyse the performance of multiple-step SM planning algorithms. One of the most important areas of future work associated with this work is the thorough investigation into the horizon dynamics in typical SM scenarios. In addition, the identification and demonstration of the optimal horizon length in such problems would represent a *significant* advancement in the development of non-myopic control strategies. Thus it is concluded that this is the avenue of future work that has the most significance in relation to this chapter, and perhaps for the entire thesis.

A route to achieving this aim is proposed as follows. Firstly, the introductory problem outlined in 5 should be extended to a more practical SM problem. Subsequently, the computation of the expected non-myopic risk and non-myopic gain, as defined in Sections 5.9.2.1 and 5.9.1.3 should proceed based on statistical analysis of multiple MC simulations of the scenario. This, it is hoped, will yield valuable information into different strategies for balancing these two mechanisms, and the demonstration of an adaptive horizon sensor control algorithm and the key mechanisms which underpin it. As a precursor to this study, a number of scenarios should be analysed in terms of their suitability, according to their sensitivity to the lookahead horizon; it is expected that only a subset of practical SM problems will be amenable to this approach.

In Chapter 5, a number of potential sources of error within a typical SM feedback-loop were identified based on prediction and utility analysis inaccuracies. There are a number of additional sources of error that could also be studied, including those associated with multiple-sensor fusion methods such as track-fusion [42]. An analysis into the effect of a non-zero probability of detection (as discussed in Section 7.3), may also identify further mechanisms which dictate the optimal horizon length.

A number of alternative theoretical formulations of the adaptive horizon control methodology could be developed, including a regularised sensor control planning algorithm. This algorithm is suggested as a surrogate to the full adaptive planning process, due to its simple structure. An investigation into the performance of this algorithm for various planning problems is therefore of great interest.

Another avenue of future work concerns the analysis of the applicability of the method to other applications, including, for example, financial portfolio optimisation, and chemical control processes, where the controller is based on a data-based system model.

7.4.3 Improvements to Repeated Weighted Boosting Search

There are several possible areas of future work in relation to the optimisation algorithms presented in Chapter 6. Firstly, a theoretical analysis of the convergence rate of the RWBS algorithm would provide further insight into the tuning process and ensure that the algorithm is guaranteed to converge to the optimal solution. The interested reader is referred to [93] and references therein. It is also noted that establishing a convergence guarantee, based purely on the random-search aspect of RWBS, would provide a solid basis for a full convergence analysis.

There are many ways in which the performance of the Pareto-RWBS algorithm could be improved. A more in-depth analysis of constraint handling capability would make the algorithm more useful in a large range of optimisation problems. In addition, an analysis of the performance of the proposed EHRWBS update stage, and the subsequent comparison with the NNRWBS update stage, would establish further insight into the applicability of the algorithm to mixed search spaces.

While the RWBS tuning procedure is relatively simple, as with all stochastic search algorithms, an automatic tuning procedure would yield a completely automatic optimisation procedure. This kind of algorithm may prove very useful for various AV problems, in which it is not always guaranteed that the nature of the optimisation problem will be known a priori. One manner in which this could be achieved is using 'iterative deepening', where the tuning parameters themselves are subject to a form of optimisation. A benchmark algorithm could be established by using a multi-resolution approach to selecting the parameters, based on the sensitivity analysis presented in Section 6.4.3.

One of the difficulties associated with developing Pareto-optimisation algorithms lies in thorough performance comparison. An important aspect of future work in relation to Chapter 6 is a more detailed assessment of the convergence performance, both in terms of Pareto-optimality and distribution across the efficient-frontier. A detailed quantitative analysis of the performance of the proposed Pareto-RWBS algorithm could, for example, be based on the research presented in [94].

Finally, it is noted that an interesting relationship between the results in Chapter 5 and 6 could be explored based on the development of an RWBS variant that is maximally robust to uncertainty in the objective function. This is particularly relevant for the multiple-step planning problems discussed in this work, where the utility function may be subject to some error, and, therefore, it is interesting to investigate optimisation algorithms which provide robust solutions. The interested reader is directed to extensive literature regarding dynamic optimisation using GA, and the test-cases proposed in [95].

Bibliography

[1] J. Manyika and H. F. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information–Theoretic Approach*. Prentice Hall, 1994.

[2] F. Bourgalt, *Decentralized Control in a Bayesian World*. PhD thesis, Australian Centre for Field Robotics, 2007.

[3] R. Malhotra, "Temporal Considerations in Sensor Management," in *Proc. of the IEEE National Aerospace and Electronics Conference (NAECON)*, vol. 1, pp. 86–93, 1995.

[4] M. L. Hernandez, "Efficient Data Fusion for Multi-Sensor Management," in *Proc. of The IEEE Aerospace Conference*, vol. 5, 2001.

[5] J. Denzler and C. M. Brown, "Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, 2002.

[6] N. Xiong and P. Svensson, "Multi–Sensor Management for Information Fusion: Issues and Approaches," *International Journal of Information Fusion*, vol. 3, no. 2, pp. 163–186, 2002.

[7] C. Kreucher, K. Kastella, and A. O. Hero, "Sensor Management using an Active Sensing Approach," *IEEE Transactions on Signal Processing*, Aug 2004.

[8] L. R. M. Johansson and N. Xiong, "Perception management–An Emerging Concept for Information Fusion," *Information Fusion*, vol. 4, no. 3, pp. 231–234, 2003.

[9] J. Llinas and D. L. Hall, "An Introduction to Multi-Sensor Data Fusion," in *Proc. of The IEEE International Symposium on Circuits and Systems*, vol. 2484, pp. 45–65, 1998.

[10] A. Steinberg, C. Bowman, and F. White, "Revisions to the JDL Fusion Model," in *Proc. of Sensor Fusion: Architectures, Algorithms and Applications III*, vol. 3719, pp. 430–441, 1999.

[11] S. Musick and R. Malhotra, "Chasing the Elusive Sensor Manager," in *Proc. of the IEEE National Aerospace and Electronics Conference (NAECON). Part 1 (of 2)*, vol. 1, pp. 606–613, 1994.

[12] E. Blasch and S. Plano, "JDL Level 5 Fusion Model: User Refinement Issues and Applications in Group Tracking," in *Proc. of The IEEE National Aerospace and Electronics Conference*, vol. 4729 of *SPIE*, pp. 270–279, 2002.

[13] E. Blasch, "Performance Metrics for Information Fusion," in *Proceedings of the NATO ASI on Rapid and Robust Situation and Threat Assesment*, (Albena, Bulgaria), 2005.

[14] D. T. Cole, I. H. Gktogan, and S. Sukkarieh, *Experimental Robotics*, vol. 39, ch. The Demonstration of a Cooperative Control Architecture for UAV Teams, pp. 501–510. Springer, January 2008.

[15] C. Harris, X. Hong, and Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data*. Springer, 2002.

[16] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter*. Artech House, 2004.

[17] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[18] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 141 of *Mathematics in Science and Engineering*. Academic Press, inc, 1979.

[19] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[20] J. Borrie, *Stochastic Systems for Engineers: Modelling, Estimation and Control*. Pretince Hall, 1992.

[21] P. J. Wojcik, "On-line Estimation of Signal and Noise Parameters with Application to Adaptive Kalman Filtering," *Circuits, Systems, and Signal Processing*, vol. 10, pp. 137–152, February 1991.

[22] A. Jazwinski, "Limited Memory Optimal Filtering," *IEEE Transactions on Automatic Control*, vol. 13, pp. 558–563, October 1968.

[23] M. S. Grewal and A. P. Andrews, *Kalman Filtering – Theory and Practice Using MATLAB*. Wiley–Interscience, 2001.

[24] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*. Academic Press, 1977.

[25] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[26] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, vol. 45, pp. 477–482, March 2000.

[27] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, March 2004.

[28] E. Wan and R. Van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Proceedomgs of the IEEE Symposium (AS-SPCC)*, 2000.

[29] N. J. Gordon, D. J. Salmon, and A. F. M. Smith, "A Novel Approach to Nonlinear–Non–Gaussian Bayesian State Estimation," in *IEE Proceedings*, vol. 140, 1993.

[30] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[31] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–189, 2002.

[32] N. Bergman, A. Doucet, and N. J. Gordon, "Optimal Filtering for Partial N–Guassian State Space Models – Computational and Theoretical Bounds," in *Proceedings of The International Symposium on Frontiers of Times Modelling (invited paper)*, 2000.

[33] A. Doucet, N. D. Freitas, K. Murphy, and S. Russell, "Rao–Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *Proceedings of The 16th Conference on Uncertainty in Artificial Intelligence*, pp. 176–183, 2000.

[34] Y. Bar-Shalom, *Multitarget–Multisensor Tracking: Applications and Advances*, vol. II. Artech House, 1998.

[35] M. L. Hernandez, "Optimal Sensor Trajectories in Bearings-Only Tracking," in *Proceedings of the 7th International Conference on Information Fusion*, (Stockhold, Sweden), pp. 893–900, 2004.

[36] Q. Gan and C. J. Harris, "Comparison of Two Measurement Fusion Methods for Kalman Filter Based Multisensor Data Fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, pp. 273–280, 2001.

[37] R. Chang, K.C.and Saha and Y. Bar-Shalom, "On Optimal Track-to-Track Fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, 1997.

[38] Y. Bar-Shalom, "The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, pp. 803–805, 1986.

[39] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, vol. 179 of *Mathematics in Science and Engineering*. Academic Press Inc, 1988.

[40] J. A. Roecker and C. McGillem, "Comparison of Two-Sensor Tracking Methods Based on State Vector Fusion and Measurement Fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, pp. 447–449, 1988.

[41] C. Harris and J. B. Gao, "Some Remarks on the Kalman Filter for the Multisensor Fusion," *Information Fusion*, vol. 3, no. 2, pp. 191–201, 2002.

[42] X. R. Li and K.-S. Zhang, "Optimal Linear Estimation Fusion-Part IV: Optimality and Efficiency of Distributed Fusion," in *Proceedings of the International Conference on Information Fusion*, (Montreal, QC, Canada), August 2001.

[43] S. Zahedi, E. Ngai, E. Gelenbe, D. Mylaraswamy, and M. Srivastava, "Information Quality Aware Sensor Network Services," in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, October 2008.

[44] R. L. Rothrock and O. E. Drummond, "Performance Metrics for Multiple–Sensor, Multiple–Target Tracking," in *Proc. of Signal and Data Processing of Small Targets*, vol. 4048 of *Proceedings of SPIE*, 2000.

[45] A. Hero, D. A. Castan, D. Cochran, and K. Kastella, *Foundations and Applications of Sensor Management*. No. 978-0-387-27892-6 in Signals and Communication Technology, Springer, 2008.

[46] A. Makarenko and H. F. Durrant-Whyte, "Decentralized Data Fusion and Control in Active Sensor Networks," in *The 7th International Conference on Information Fusion*, 2004.

[47] B. Grocholsky, H. F. Durrant-Whyte, and P. Gibbens, "An Information–Theoretic Approach to Decentralised Control of Mulitple Autonomous Flight Vehicles," in *Proceedings of SPIE*, vol. 4196, pp. 348–359, 2000.

[48] B. Grocholsky, *Information–Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sydney, 2002.

[49] L. Chen, A. I. El-Fallah, R. K. Mehra, J. R. Hoffman, R. P. Mahler, and M. G. Alford, "Initial Studies on Direct Sensor Management Optimization using Tracking Performance Metrics and Genetic Algorithms," in *Proc. of Signal Processing, Sensor Fusion and Target Recognition XII*, vol. 5096 of *Proc. of SPIE – The International Society for Optical Engineering*, pp. 233–244, 2003.

[50] B. Grocholsky, A. Makarenko, and H. F. Durrant-Whyte, "Information–Theoretic Coordinated Control of Multiple Sensor Platforms," in *Proc of The IEEE International Conference on Robotics and Automation*, 2003.

[51] B. Grocholsky, A. Makarenko, T. Kaupp, and H. F. Durrant-Whyte, "Scalable Control of Dencentralised Sensor Platforms," in *Proc. of Information Processing in Sensor Networks*, 2003.

[52] S. Bryson, M.T. ans Sukkarieh, "Decentralised Trajectory Control for Multi-UAV SLAM," in *Fourth International Symposium on Mechatronics and its Applications*, (United Arab Emirates), March 2007.

[53] K. Kastella, "Discrimination Gain to Optimize Detection and Classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems, Man and Humans*, vol. 27, no. 1, 1997.

[54] R. P. Mahler, "Global Posterior Densities for Sensor Management," in *Proc. of SPIE*, vol. 3365, pp. 252–263, 1998.

[55] A. El-Fallah, M. Perloff, A. Gandhe, R. Mahler, T. Zajic, and C. Stelzig, "Multisensor–Multitarget Sensor Management using Geometric Objective Functions," in *Proc of The International Conference on Integration of Knowledge Intensive Mulit–Agent Systems: Modelling, Exploration, and Engineering*, pp. 349–354, 2003.

[56] R. P. Mahler, "Objective Functions for Bayesian Control–Theoretic Sensor Management, I: Multitarget First–Moment Approximation," in *Proceedings of The IEEE Aerospace Conference*, vol. 4, pp. 1905–1923, 2003.

[57] R. P. Mahler, "Objective Functions for Bayesian Control–Theoretic Sensor Management, II: MHC-Like Approximation," *Recent Developments in Cooperative Control and Optimization*, vol. 3, pp. 273–316, 2004.

[58] M. L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom, "Multisensor Resource Deployment using Posterior Cramer–Rao Bounds," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 399–416, 2004.

[59] M. L. Hernandez and P. R. Horridge, "Advances in the Management of Multi–Sensor Systems with Associated Applications," in *Target Tracking 2004 Algorithms and Applications, IEE*, vol. 23–24, pp. 121–130, 2004.

[60] P. R. Horridge and M. L. Hernandez, "A Fast-Jet Pursuer-Evader Game: How to Track a Difficult Target," in *Proceedings of the SPIE - Signal and Data Processing of Small Targets*, vol. 5428, pp. 237–248, 2004.

[61] M. Kalandros, "Covariance Control for Multisensor Systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 4, pp. 1138– 1157, 2002.

[62] M. Kalandros and L. Pao, "Controlling Target Estimation Covariance in Centralized Multisensor Systems," in *Proc. of the American Control Conference*, vol. 5, pp. 2749–2753, 1998.

[63] C. Kreucher, A. O. Hero, K. Kastella, and D. Chang, "Efficient Methods of Non-Myopic Sensor Management for Multitarget Tracking," in *Proc. of the 43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, pp. 722–727, 2004.

[64] C. M. Kreucher, *An Information Based Approach to Sensor Resource Allcoation*. PhD thesis, University of Michigan, USA, 2005.

[65] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd ed., 2001.

[66] A. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Nonmyopic Sensor Scheduling and its Efficient Implementation for Target Tracking Applications," in *Journal on Applied Signal Processing*, vol. 2006, pp. 1–18, EURASIP, 2006.

[67] M. Cone and S. Musick, "Scheduling for Sensor Management using Genetic Search," in *Proceedings of the Aerospace and Electronics Conference*, vol. 1, (Dayton, OH, USA), pp. 150–156, May 1996.

[68] Y. Yong, G. Lei, C. Liang, and C. Yan-Fang, "Ant Colony Optimization for Sensor Management," in *Proceedings of the International Conference on Control and Automation*, pp. 576–580, June 2007.

[69] J. F. Whidborne, G. Liu, and J. Yang, *Multiobjective Optimisation and Control*. Research Studies Press LTD, 2003.

[70] C. Fonseca and P. Fleming, "Multiobjective Genetic Algorithms made Easy: Selection Sharing and Mating Restriction," in *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA)*, 414, pp. 45–52, 1995.

[71] Y. Mei, Y. Lu, Y. C. Hu, and C. S. G. Lee, "Deployment of Mobile Robots with Energy and Timing Constraints," *IEEE Transactions on Robotics*, vol. 22, pp. 507–522, June 2006.

[72] L. Pao and M. Kalandros, "The Effects of Delayed Sensor Requests on Sensor Manager Systems," in *Proc. of AIAA Guidance Navigation and Control Conference*, 1998.

[73] C. M. Fonseca, *Multiobjective Genetic Algorithms with Application to Control Engineering Problems*. PhD thesis, Department of Automatic Control and Systems Engineering, September 1995.

[74] S. F. Page, A. N. Dolia, C. J. Harris, and N. M. White, "Multiple Objective Optimization for Active Sensor Management," in *Proceedings of the SPIE: Multisensor, Multisource Information Fusion - Architectures, Algorithms, and Application* (B. V. Dasarathy, ed.), vol. 5813, (Orlando, Florida, USA), pp. 269–280, SPIE, Bellingham, WA, 2005.

[75] S. F. Page, A. N. Dolia, C. Harris, and N. M. White, "Adaptive Horizon Model Predictive Control based Active Sensor Management for Multi-Target Tracking," in *Proceedings of the American Control Conference*, (Minneapolis, Minnesota, USA), June 2006.

[76] D. P. Bertsekas, "Dynamic Programming and Suboptimal Control: A Survey From ADP to MPC," *European Jouranl of Control*, vol. 11, no. 4-5, 2005.

[77] C. Camacho, E. F. amd Bordons, *Model Predictive Control*. Springer-Verlag, 1999.

[78] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2001.

[79] K. Deb, A. Pratap, l. S. Agarwa, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NGSA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[80] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, vol. 3, pp. 1–16, Spring 1995.

[81] S. Chen, X. X. Wang, and C. J. Harris, "Experiments with Repeating Weighted Boosting Search for Optimization in Signal Processing Applications," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, August 2005.

[82] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *International Conference on Machine Learning*, pp. 148–156, 1996.

[83] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Computer Journal*, vol. 7, pp. 308–313, January 1965.

[84] The MathWorks Inc, *MATLAB Optimization Toolbox User Guide*, 3rd ed., September 2006.

[85] The MathWorks Inc, *MATLAB Genetic Algorithm and Direct Search Toolbox 2*, 2.1 ed., March 2007.

[86] K. A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975.

[87] A. Roy and J. G. Stell, "Convexity in Discrete Space," in *Proceedings of Spatial Information Theory* (W. Kuhn, M. F. Worboys, and S. Timpf, eds.), vol. 2825 of *Lecture Notes in Computer Science*, pp. 268–285, 2003.

[88] K. Deb, A. Pratap, and T. Meyarivan, "Constrained Test Problems for Multi-objective Evolutionary Optimization," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 284–298, London, UK: Springer-Verlag, 2001.

[89] B. Wah and Y. Chen, "Constrained Genetic Algorithms and their Applications in Nonlinear Constrained Optimization," in *International Conference on Tools with Artificial Intelligence*, pp. 286–293, 2000.

[90] C. M. Fonseca and P. J. Fleming, "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part I: A Unified Formulation," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, 1998.

[91] K. Deb, "Multi-objective Genetic Algorithms - Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.

[92] E. Parzen, "On the Estimation of a Probability Density Function and Mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[93] Y. Gao, "An Upper Bound on the Convergence Rates of Canonical Genetic Algorithms," *Complexity International*, vol. 5, 1998.

[94] C. M. Fonseca, V. G. D. Fonseca, and L. Paquete, *Evolutionary Multi-Criterion Optimization*, ch. Exploring the Performance of Stochastic Multiobjective Optimisers with the Second-Order Attainment Function, pp. 250–264. No. 978-3-540-24983-2, Springer, Janurary 2005.

[95] J. Yaochu and B. Sendhoff, "Constructing Dynamic Optimization Test Problems using the Multi-Objective Optimization Concept," *Lecture Notes in Computer Science*, vol. 3005, pp. 525–536, March 2004.