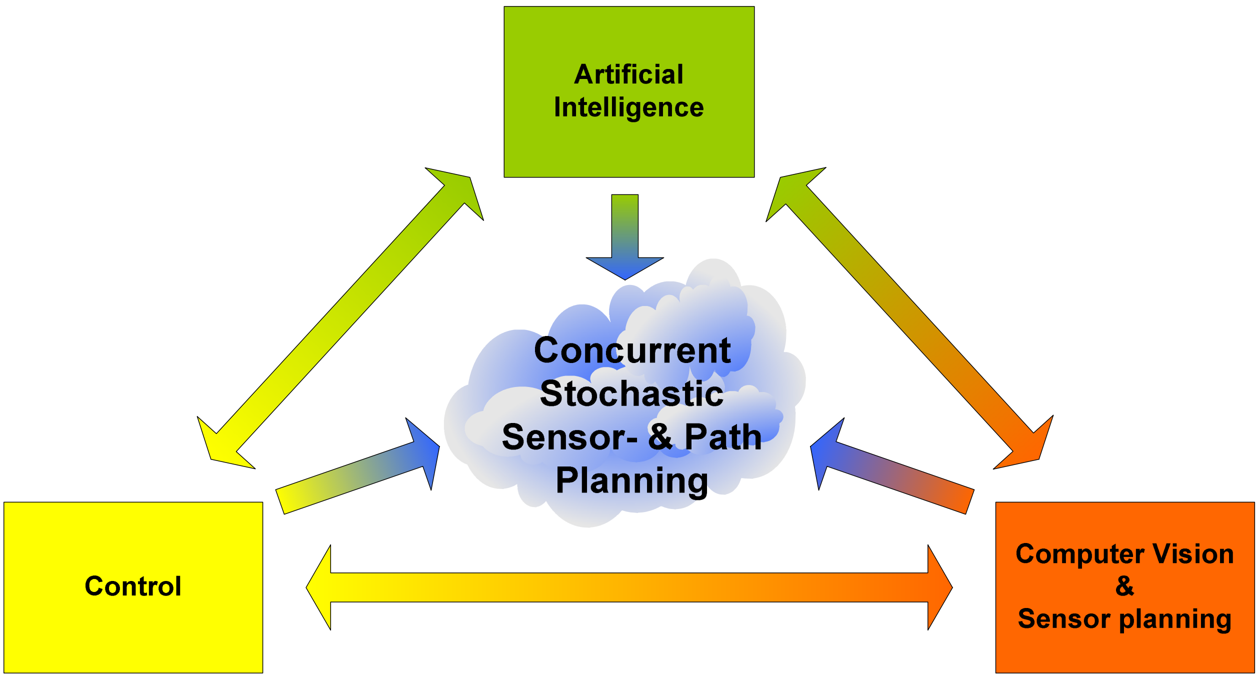
Towards Concurrent Sensor and Path Planning

A Survey of Planning Methods Applicable to UAV Surveillance

Jonas Nygårds
Per Skoglar
Jörgen Karlholm
Rickard Björström
Morgan Ulvklo



FOI-R-- 1711 --SE
ISSN 1650-1942

Sensor Technology
Scientific report

May 2005

FOI is an assignment-based authority under the Ministry of Defence. The core activities are research, method and technology development, as well as studies for the use of defence and security. The organization employs around 1350 people of whom around 950 are researchers. This makes FOI the largest research institute in Sweden. FOI provides its customers with leading expertise in a large number of fields such as security-policy studies and analyses in defence and security, assessment of different types of threats, systems for control and management of crises, protection against and management of hazardous substances, IT-security an the potential of new sensors.

FOI
Defence Research Agency
Sensor Technology
P.O. Box 1165
SE-581 11 Linköping

Tel: +46 (0)8 555 030 00
Fax: +46 (0)8 555 031 00

www.foi.se

Towards Concurrent Sensor and Path Planning

A Survey of Planning Methods Applicable to UAV Surveillance

FOI-R--1711--SE
ISSN 1650-1942

Sensor Technology
Scientific report

May 2005

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency	FOI-R--1711--SE	Scientific report
Sensor Technology P.O. Box 1165 SE-581 11 Linköping	**Research area code** 7. Mobility and space technology, incl materials	
	Month year May 2005	**Project no.** E3969
	Sub area code 71 Unmanned Vehicles	
	Sub area code 2	
Author/s (editor/s) Jonas Nygårds Per Skoglar Jörgen Karlholm Rickard Björström Morgan Ulvklo	**Project manager** Morgan Ulvklo **Approved by** Lena Klasén **Sponsoring agency** FMV/Swedish Armed Forces **Scientifically and technically responsible** Jonas Nygårds	

Report title

Towards Concurrent Sensor and Path Planning

A Survey of Planning Methods Applicable to UAV Surveillance

Abstract (not more than 200 words)

Autonomous path and sensor planning of a UAV with EO/IR sensors for surveillance and exploration is a very challenging problem. Realistic models of environment, sensors and platforms are very complex, due to the non-linear and stochastic properties of the world. Hence, algorithms and methods solving realistic planning problems are computationally very demanding. To develop a monolithic planner is probably an impossible task; instead a hierarchical decomposition is necessary. The issue then, is how to decompose the problem into sub-problems that guarantee that the overall objective is achieved.

The purpose of this survey is to present suitable methods and approaches that can be applied to surveillance and exploration with autonomous UAVs; in particular, concurrent sensor and path planning methods. First the UAV surveillance and reconnaissance problem is introduced. An overview of research fields and communities related to path planning and/or sensor planning is then given. The main parts of this report are presentations of methods, techniques and approaches to path planning and sensor planning. Furthermore, a brief survey of related research in Sweden is given.

The concurrent path and sensor planning problem is very complex and no method has been found that solves the problem in a satisfying way. However, some promising directions are identified and since much research related to this area is in progress, interesting results will hopefully appear during the next few years.

Keywords

path planning, sensor planning, autonomy, motion planning, UAV, EO/IR, surveillance, exploration, control, artificial intelligence, computer vision, robotics, sensor management, information theory, optimal control, graph search

Further bibliographic information	Language English
ISSN 1650-1942	**Pages** 90 p.
	Price acc. to pricelist

FOI 2005 Utgåva 12

Utgivare	Rapportnummer, ISRN	Klassificering
FOI - Totalförsvarets Forskningsinstitut - Sensorteknik Box 1165 581 11 Linköping	FOI-R--1711--SE	Vetenskaplig rapport
	Forskningsområde 7. Farkost- och rymdteknik, inkl material	
	Månad, år Maj 2005	**Projektnummer** E3969
	Delområde 71 Obemannade farkoster	
	Delområde 2	
Författare/redaktör Jonas Nygårds Per Skoglar Jörgen Karlholm Rickard Björström Morgan Ulvklo	**Projektledare** Morgan Ulvklo **Godkänd av** Lena Klasén **Uppdragsgivare/kundbeteckning** FMV / Försvarsmakten **Tekniskt och/eller vetenskapligt ansvarig** Jonas Nygårds	

Rapportens titel (i översättning)

Samtidig sensor- och ruttplanering – En kartläggning av metoder användbara för UAV-övervakning

Sammanfattning (högst 200 ord)

Autonom rutt- och sensorplanering av en UAV med EO/IR-sensorer för spaning och övervakning är ett mycket utmanande problem. Realistiska modeller av omvärld, sensorer och plattform är komplexa p.g.a. icke-linjära och stokastiska beskaffenheter hos vår värld. Följaktligen blir algoritmer och metoder som löser realistiska planeringsproblem väldigt beräkningskrävande. Att utveckla en enda monolitisk planerare är förmodligen en omöjlig uppgift, istället är en hierarkisk uppdelning nödvändig. En viktig fråga är dock hur denna uppdelning i mindre delproblem görs så att det övergripande målet ändå uppnås.

Syftet med denna litteraturstudie är att presentera metoder och angreppssätt användbara för spaning och övervakning med UAV:er; i synnerhet metoder för samtidig rutt- och sensorplanering. Först introduceras problemet med spaning och övervakning med autonoma UAV:er. Sedan ges en översikt över olika forskningsområden relaterade till rutt- och/eller sensorplanering. Rapportens huvuddelar presenterar metoder och angreppssätt för ruttplanering och sensorplanering. Dessutom ges en kortfattad genomgång av relaterad forskning i Sverige.

Problemet med samtidig rutt- och sensorplanering är mycket komplext och ingen metod har hittats som på ett tillfredställande sätt löser det. Det finns dock några angreppssätt och inriktningar som verkar lovande. Eftersom det pågår forskning relaterat till det aktuella problemet på flertalet platser i världen, kan man förvänta sig att intressanta resultat kommer fram inom de närmaste åren.

Nyckelord

ruttplanering, sensorplanering, autonomi, UAV, EO/IR, övervakning, spaning, reglerteknik, artificiell intelligens, datorseende, robotik, sensor management, informationsteori, optimal styrning, grafsökning

Övriga bibliografiska uppgifter	Språk Engelska
ISSN 1650-1942	**Antal sidor:** 90
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

1 Introduction **9**
1.1 Background and Objective . . . 9
1.2 Outline and Focus . . . 9

2 Path and Sensor Planning for a UAV with Vision Sensor **11**
2.1 Signal processing and surveillance tasks . . . 11
2.2 Enhanced levels of autonomous planning . . . 11
2.3 Surveillance services . . . 13
2.3.1 Surveillance Requests . . . 13
2.3.2 Surveillance Responses . . . 14
2.4 Autonomy Domain Constraints . . . 14
2.5 Planning Objectives and Operating Modes . . . 15
2.6 Planning Constraints . . . 15
2.7 Path and Sensor Planning Levels . . . 16

3 Related Research Fields and Communities **19**
3.1 The Wide Web of Planning Fields and Communities . . . 19
3.2 Control . . . 19
3.2.1 Adaptive and Restructurable Control . . . 20
3.2.2 Intelligent Control . . . 21
3.2.3 Expert Control . . . 21
3.2.4 Hybrid Control . . . 22
3.2.5 Dual Control . . . 22
3.3 Robotics . . . 22
3.3.1 Robot Motion Planning . . . 22
3.3.2 Visual Servoing . . . 23
3.3.3 Active Sensing . . . 23
3.4 Computer Vision . . . 23
3.4.1 Sensor Planning . . . 23
3.5 Aerospace . . . 24
3.6 Data Fusion . . . 24
3.6.1 Sensor Management and Resource Management . . . 24
3.7 Operations Research (OR) . . . 25
3.7.1 Search Theory . . . 25
3.8 Artificial Intelligence (AI) . . . 26
3.8.1 AI planning . . . 26
3.8.2 A Roadmap for Research in Robot Planning . . . 27
3.9 References . . . 28

4 Some Useful Results from Control, Optimization, and Estimation Theory 29
4.1 Optimal Stochastic Control Theory . . . 29
4.1.1 A typical finite horizon problem formulation . . . 29
4.1.2 The principle of optimality and dynamic programming . . . 30
4.1.3 Linear Quadratic Gaussian Control (LQG) . . . 30
4.1.4 Non-linear problems and sub-optimal control . . . 32
4.1.5 Infinite horizon - Reinforcement learning . . . 32
4.2 Forward Graph Search Algorithms . . . 33
4.2.1 General Forward Search Algorithm . . . 33
4.2.2 Breadth First Algorithm . . . 34
4.2.3 Depth First Algorithm . . . 34
4.2.4 Dijkstra's Algorithm . . . 34
4.2.5 A^* Algorithm . . . 35
4.3 Information Theory . . . 35
4.3.1 Entropic Information . . . 35
4.3.2 Fisher Information . . . 36
4.3.3 The Information Filter . . . 36
4.4 Exploration Driven by Uncertainty . . . 37
4.4.1 Linear Model Estimation Basics . . . 37
4.4.2 Reducing Uncertainty . . . 38
4.5 Classification of planning algorithms . . . 38
4.5.1 Stochastic and Deterministic Planning . . . 39
4.5.2 Optimal, Suboptimal and Heuristic Planning . . . 39
4.5.3 Implicit and Explicit Planning . . . 39
4.5.4 Continuous, Discrete and Hybrid Planning . . . 39

5 Path Planning 41
5.1 Robot Motion Planning . . . 41
5.2 Optimal Control Examples . . . 41
5.2.1 UAV Path Planning . . . 42
5.2.2 Vision Motion Planning with Uncertainty . . . 43
5.2.3 Dual Control in Robotics . . . 43
5.2.4 Reliable Control of Intelligent Machines . . . 43
5.2.5 An Optimal Observer Maneuver Example . . . 44
5.3 Applications of Search Theory . . . 48
5.4 Path Planning based on Visibility . . . 50
5.5 Graph-Based Path Planning . . . 50
5.5.1 Voronoi Diagram . . . 51
5.5.2 Probabilistic Roadmaps . . . 51
5.5.3 Occupancy Grid . . . 52
5.6 Path Planning based on Potential Field and Virtual Forces . . . 52
5.6.1 Potential Field . . . 52
5.6.2 Virtual Forces Approaches . . . 52
5.7 Trajectory Generation and Smoothing . . . 53
5.8 UAV Path Planning and Cooperation . . . 54
5.8.1 UAV Path Planning Examples . . . 54
5.8.2 Trajectory Planning with Mixed-Integer Linear Programming . . . 54
5.8.3 Cooperative Time Problems . . . 55
5.8.4 Cooperative Search Problems . . . 55
5.9 References . . . 56

6 Sensor Planning and Control **57**
6.1 Sensor Planning in Computer Vision 57
6.2 Next-Best-View (NBV) 58
6.2.1 NBV according to Reed et al. 58
6.2.2 NBV according to Pito 59
6.3 Sensor Placement with Genetic Algorithms 62
6.4 Exploration Driven by Uncertainty - A Gradient Search Approach 62
6.5 Decision-Theoretic Sensor Planning 63
6.6 Sensor Management, Allocation and Tracking 64
6.7 Visual Servoing and Reactive Control 64
6.8 References 66

7 Concurrent Path & Sensor Planning - Discussion and Conclusions **67**

A Overview of Related Research in Sweden **71**
A.1 FOI 71
A.1.1 Department of IR Systems 71
A.1.2 Department of Data and Information Fusion 71
A.1.3 Department of Autonomous Systems 72
A.2 Linköping University 72
A.2.1 WITAS 72
A.2.2 Control and Communication 73
A.2.3 Robotics and Autonomous Mechanical Systems (R/AMeS) 73
A.3 The Royal Institute of Technology (KTH) 73
A.3.1 Aeronautical and Vehicle Engineering 73
A.3.2 Centre for Autonomous Systems (CAS) 74
A.4 Chalmers University of Technology 74
A.5 Lund University 75
A.6 Örebro University 75
A.7 Luleå University of Technology 75
A.8 Halmstad University 75
A.9 Umeå University 76

B Reference List **77**

Chapter 1

Introduction

This report is concerned with methods and approaches for path and sensor planning for a UAV with gimballed EO/IR sensors. The report is the result of a study performed during the spring 2004.

1.1 Background and Objective

The need for autonomous capabilities, such as on-board sensor data processing, sensor management, and path planning, will increase in both manned and unmanned platforms designed for future network centric military operations. This arises from the constantly growing quantity of sensors and associated raw data, as well as limitations in communication bandwidth and processing capacity of human sensor operators.

Several basic functionalities of autonomous surveillance systems, e.g., target geolocation, robust navigation, collision avoidance, route and viewpoint planning all require advanced visual capabilities like target and landmark recognition, scene topography estimation, and image-motion computation. In order to raise the level of autonomy in these systems, it is necessary to take into account the uncertainty associated with the percepts of a cluttered and rapidly changing environment. These uncertainties arise from sensor noise, navigation errors, matching errors, prior knowledge model errors, and target prediction errors.

Concurrent sensor and path planning, taking into account both platform and sensor constraints, as well as threats and environmental conditions, is a very demanding task. Even more demanding, but still necessary, is the capability to dynamically adapt and replan the sensor utilization and the platform trajectory in response to changes in the environment as well as internal state, given *feedback* from new sensor data. Our working hypothesis is that integration of the detection-recognition chain with spatial awareness makes possible intelligent autonomous data acquisition by means of active sensor control and path planning.

This report is a survey of control and planning approaches from different scientific areas applicable to autonomous UAV surveillance.

1.2 Outline and Focus

Planning is a very large research field with several subfields and several communities as participants. This report focuses on path and sensor planning methods, but nevertheless the area is still very large. It is impossible to cover this research area completely, due to limited time. To give a wide overview, while allowing deeper study into methods important to the specific problem, it is inevitable that the outline is biased towards topics of particular relevance to us.

- Chapter 2 gives an introduction to the UAV surveillance and reconnaissance problem. In particular, the path and sensor planning challenges in UAV surveillance is discussed. This chapter is a summary of [153].

- Chapter 3 is an overview of research fields and communities related to path planning and/or sensor planning. The purposes are to give an introduction to different research areas related to planning and to show the complex web of connections between these research fields and communities.

- Chapter 4 presents some mathematical results and tools useful for planning problems. In particular, optimal control is considered.

- Chapter 5 and Chapter 6 present methods, techniques and approaches to the path and sensor planning problem, respectively.

- Chapter 7 discusses the stochastic concurrent path and sensor planning problem. Conclusions and promising research directions are given.

- Appendix A gives a brief presentation of related research in Sweden.

- Appendix B contains a list with interesting references.

Chapter 2

Path and Sensor Planning for a UAV with Vision Sensor

This chapter gives an introduction to the UAV surveillance and reconnaissance problem. In particular, the path and sensor planning challenges in UAV surveillance is discussed. In the current context, we define path planning as planning of the UAV platform trajectory, i.e., path, velocity, etc., whereas sensor planning is defined as planning of a gimballed EO/IR sensor, including gaze direction, focus, zoom, contrast, etc.

2.1 Signal processing and surveillance tasks

The system-oriented research at FOI puts special emphasis on EO/IR image processing and control mechanisms to enhance the level of autonomy in UAV surveillance. Research topics under consideration include:

- Reduction of the amount of data distributed from a sensor node, such as a UAV with EO/IR sensors, so that a network centric system is not overloaded.
- Development of sensor related network services that use advanced sensor data processing to concurrently solve problems such as area coverage, detection, association, tracking, geolocation, change detection, and classification.
- Improvement of data acquisition and sensor data analysis, using network distributed prior knowledge and complementary sensor data in the low level processing of a sensor node with controllable EO/IR sensors.
- Incorporation of real-time sensor data analysis in path planning and sensor management in order to improve the data acquisition process.
- Optimal utilization of surveillance imagery in precision targeting.

The research strives to develop a system architecture for UAV signal processing that incorporates these aspects. Sensor and platform planning are key components in such architecture. Basic image processing capabilities required to implement such performance enhancements are described in [153].

2.2 Enhanced levels of autonomous planning

The long term goal of the research on UAV surveillance at FOI is a framework for autonomous sensor management, designed to enhance the capability to handle multiple concurrent surveillance requests and raise the level of autonomy. A number of arguments are given here to describe limiting factors in UAV surveillance of today and point out enhanced capabilities with the introduction of higher level of autonomy in sensor and platform planning.

- The ability to manually control the sensor system and laser designator of some UAV systems is limited due to long round-trip latencies in communication link systems (mostly SATCOM related). Automatic tracking and high precision sensor pointing would significantly improve the performance of such systems.

- A human operator is limited to executing only a few tasks in parallel. Fast scheduling between parallel requests requires autonomous sensor control without the human transition time related to context changes. A typical example is a human operator detecting a number of ground targets, zooming in on one and at the same time losing the others out of sight. A fast man-machine interaction using a multi-target tracking system combined with autonomous sensor control would significantly enhance the performance in such a scenario. This is a critical capability when implementing many weapon engagement concepts based on Network Centric Warfare with online target position updates.

- Cueing and sifting mechanisms capable of detecting, tracking, and geolocating multiple ground targets would be of great importance in time critical situations to help the image analyst focus on relevant parts of the surveillance data. Integration of this detection-recognition chain with spatial awareness makes intelligent data acquisition possible by means of active sensor control and path planning. This is of considerable importance for achieving useful autonomous surveillance systems.

- Geolocation of stationary and moving ground targets can improve significantly by including prior knowledge in the estimation process, such as georeferenced imagery and other landmarks. Further improvements can be achieved by integrating the association process with active sensor management. The system will then autonomously schedule sensor resources between different surveillance and geolocation tasks.

- In some advanced future applications, the synchronization between the platform trajectory and the sensor control system will be critical. A typical UAV application related to this problem is low flight altitude surveillance in urban warfare. Very narrow time slots for the sensor control, due to the high degree of occlusion from buildings, require autonomous sensor planning and control to establish guaranteed ground coverage of the streets.

- Successful interaction between multiple UAV surveillance platforms and/or weapon platforms is strongly dependent on a proper representation of the surveillance scene. Decreased time to initiate engagement quality tracks of moving targets would be a direct benefit of sensor data exchange and autonomous sensor and platform planning.

- The ability to dynamically re-plan surveillance missions to accommodate new or updated requests for information will be a basic requirement of autonomous surveillance systems in the future. Being able to re-plan also permits servicing self-initiated surveillance requests based on target indications or other unexpected events in the imagery. This is a required basic component to enable weapon engagement of high-value time critical targets.

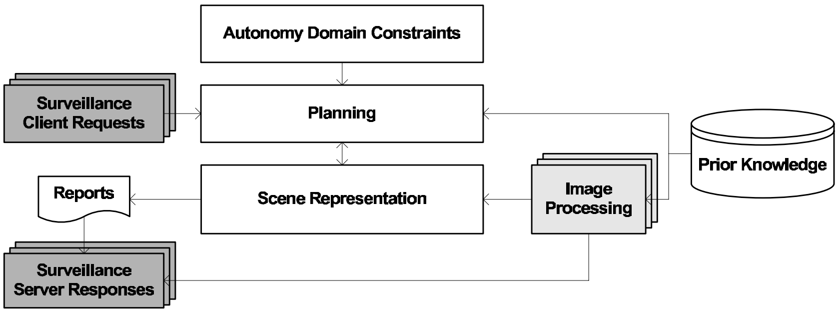


Figure 2.1. An overview of necessary signal processing components in autonomous UAV surveillance.

2.3 Surveillance services

Depending on priority, threat level, and flight-time, a new in-flight requested surveillance information service would require allocation of accessible surveillance resources to meet the request for new sensor data. The provider of the surveillance services is the signal processing and control system onboard the UAV itself, or related ground stations, depending on the level of autonomy of the system. However, planning, synchronization and management of surveillance resources over a larger area-of-responsibility is a very demanding procedure.

A client/server approach, designed for managing adaptable surveillance missions, is introduced in [153]. The framework is based on a relationship between information consumers, who dispatch *surveillance client requests*, and a service provider, responsible for *surveillance server responses* (see figure 2.1).

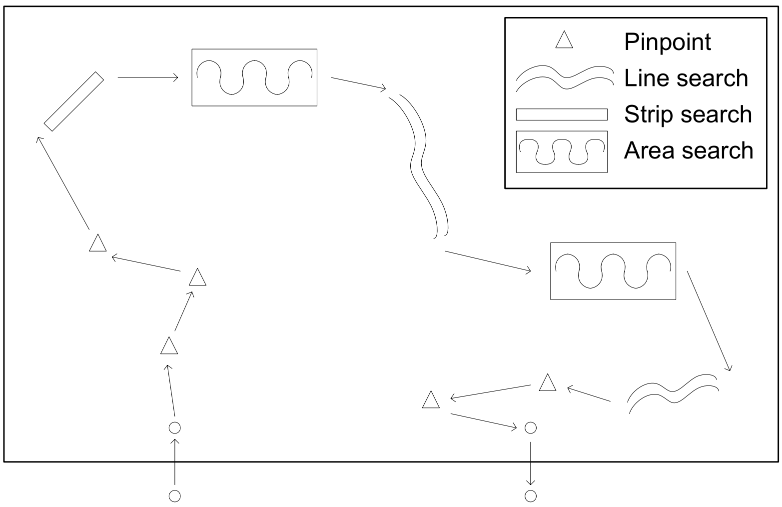


Figure 2.2. Different types of search patterns for surveillance requests in the area of responsibility (AOR).

2.3.1 Surveillance Requests

Surveillance client requests can be categorized using three different criteria: *search pattern*, *task*, and *origin*. The definition of different search patterns we use follows traditional air reconnaissance standards. Given an area of responsibility (AOR), four different surveillance search patterns are defined:

Pinpoint	A limited area around a reference point, known within 100 meters.
Strip search	A task along a straight line between two reference points.
Line search	A task along a communication route, e.g., a railway or road.
Area search	A task over a larger terrain, sea, or urban area.

Figure 2.2 illustrates a surveillance mission with multiple target areas. Given a specific search pattern, four different task categories with associated report requirements are defined:

A: New target	All target features should be reported.
B: Change detection	Subset of all features.
C: Attack planning	Requested features.
D: Damage assessment	Depending on target type.

Only categories A and B are today under consideration in the architecture due to the complexity of C and D. Therefore, there are totally eight combinations of task and search pattern possible in a surveillance request. From a planning point of view, client requests can be generated in three different ways:

Pre-planned	Requests given before take-off of the UAV platform.
In-flight external	New requests given during the implementation of the mission.
In-flight self-initiated	New requests initiated by the system itself. An example is a closer scrutinization of ROIs generated from detection of ground targets.

2.3.2 Surveillance Responses

Depending on available bandwidth and the customer's ability to interpret the imagery, the surveillance client responses consist of surveillance imagery in combination with target meta-data, such as geocoordinates, velocity and target type. Examples are:

- Overviews, such as video mosaics.
- High-resolution multi-view imagery for pinpoint requests.
- High-resolution multi-view imagery of detected ROIs.
- Change detection versus previous flights.
- Recurrent updates of surveillance information.

2.4 Autonomy Domain Constraints

Up to now, autonomous systems have mainly been categorized in terms of capabilities or "intelligence". The Air Vehicle Directorate at AFRL has introduced the notion of *autonomous control level*, ACL, and describe ten such levels, ranging from remotely piloted vehicles to fully autonomous swarms of UCAVs [1, 38]. In [38], Clough argues that the degree of autonomy of a UAV should be expressed in terms of to what extent it can replace humans in the OODA (observe, orient, decide, and act) loop.

In traditional airborne surveillance, air traffic in an AOR is very restricted and controlled. To be able to introduce an autonomous system into the airspace, one must clearly define that system's freedom of action and movement. To that end, we introduce the *Autonomy Domain Constraint*, ADC, which defines the domain of autonomy in terms of freedom of action and movement.

The most restrictive, *zero-level ADC*, permits only basic automatic functions, such as pre-programmed platform control using way-points, and pre-programmed or manual sensor control. The *first level ADC* applies to autonomous control of a sensor system. An example is vision-based feedback for solving multiple concurrent sensing tasks, e.g., multi-target tracking where not all targets are simultaneously in the field of view. Some sort of scheduling mechanism is then required for prioritizing and sequencing concurrent tasks. This level of ADC does not affect the actual platform trajectory, only sensor and signal processing control. In the case of active sensors, e.g. radar or laser, planning may have to take into account the risk of being detected. The *second level ADC* incorporates short-term platform trajectory planning in combination with sensor planning, for single or multiple platforms. The platform adheres to a predefined long term flight plan but modifies it locally in space and time to fit the current task. A flight corridor limits the platform autonomy in space and time. A *third level ADC* also includes long term planning of missions with multiple surveillance client requests. At this level of autonomy, an in-flight generated new request might completely change the ordering between the surveillance tasks of the mission. To summarize:

- ADC 0 Pre-programmed or manual control of platform and sensor.
- ADC 1 Sensor planning and servoing, and pre-programmed platform control.
- ADC 2 Short term platform planning (within free flight corridor) and ADC 1.
- ADC 3 Long term platform planning (within free airspace in AOR) and ADC 2.

2.5 Planning Objectives and Operating Modes

Section 2.3 discusses different objectives, tasks, requests and search patterns. They can be divided into two classes of request modes, *Surveillance & search* and *Tracking & data acquisition*. Surveillance & search in turn can be divided into area search, strip search, line search, and pinpoint. Tracking & data acquisition involves multi-target tracking, precise target coordinate generation, and detailed ROI data acquisition.

To successfully plan and perform these tasks autonomously, some operating modes that facilitate the planning and navigation are also required. Thus, we introduce a *Planning & navigation support* mode that builds and maintains a world model that the planning optimization and navigation estimation are based on. This mode is, for instance, performing probing actions, occlusion estimation, obstacle detection and map building.

Hence, we have three classes of operating modes:

- Surveillance & search
- Tracking & data acquisition
- Planning & navigation support

At the sensor level only one mode is executed at a time. However, several tasks requiring different modes may simultaneously be requesting the sensor resource, and the planning must therefore incorporate some kind of sensor scheduling to allow the system to quickly switch between different modes. The planner does not necessarily require an explicit scheduler; approaches may exist where the scheduling behaviour is a natural part of the planner framework.

2.6 Planning Constraints

The planning optimization process is affected by *planning constraints*. We have identified six classes of constraints:

- *Platform constraints* are associated with the UAV platform, such as dynamic, kinematic, nonholonomic, and fuel constraints.
- *Environmental constraints* define areas where the platform cannot or should not be, and thus include geometric constraints, accessibility constraints, obstacle avoidance, and threat avoidance. Also the autonomy domain constraints, discussed in Section 2.4, belong to environmental constraints.
- *Viewpoint constraints* define areas where visibility is reduced relative some task, for instance due to occlusion, distance, and viewing angle.
- *Sensor constraints* are associated with the gimbal, e.g., dynamic and kinematic constraints, and to the sensor itself, such as field-of-view, resolution, and contrast.
- *Target constraints* are associated with properties that affect the detectability of the target, e.g. target motion and pixels over target.
- *Timing constraints* affect all aspects of planning from platform to sensor.

2.7 Path and Sensor Planning Levels

In Section 2.3 four different search patterns were mentioned. Consider a *line search* example, road surveillance. This involves searching for targets along a road and gathering detailed information (high-resolution images, geo-referenced position, etc.) of detected targets. Problems in this surveillance task could be threats and occlusion due to trees, buildings, or terrain masking. The controller must be able to handle uncertainties, such as partially unknown occlusion and road position.

Prior information, e.g. GIS data and prior imagery, Figure 2.3*(a)*, is useful in the initial planning *(b)*, but as the surveillance process progresses it is necessary to look ahead *(c)* and adjust the plan *(d)* due to uncertainties and errors in the prior information. Performance measurements are needed to verify mission success. For instance, a high detection probability can be achieved without necessarily covering every square meter of the road or the ground.

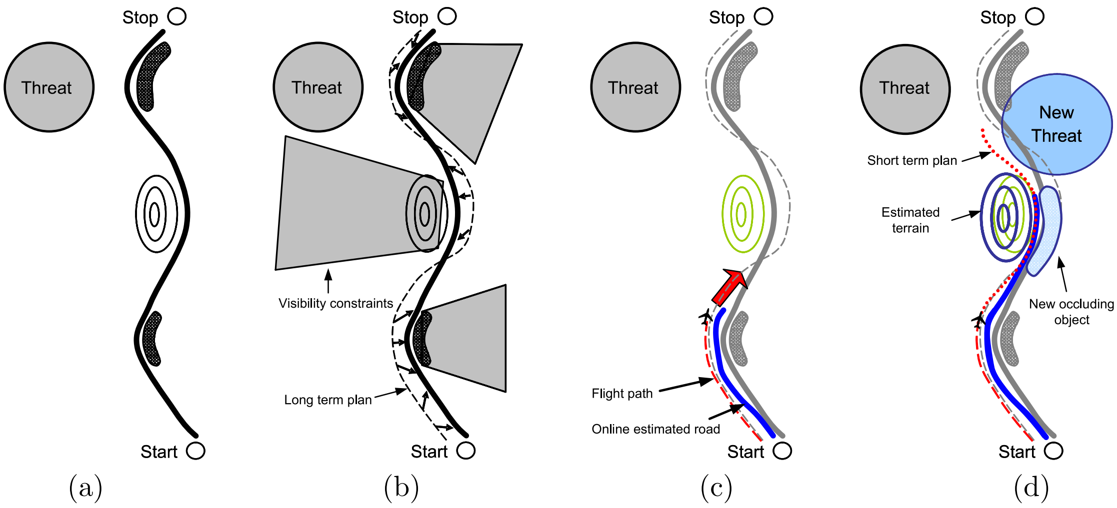


Figure 2.3. Road surveillance scenario. (a) Prior information. (b) Initial plan based on prior information. (c) "Probing", i.e. look ahead and update the world model, is necessary. (d) Replanning is required by new detected visibility and environmental constraints.

A successful solution to the road surveillance scenarios above should display properties such as probing, caution and reactive behaviour. Probing represents actions to enhance estimation precision in order to improve overall performance in the future. Caution is acting so as to minimize the consequences of erroneous assumptions about the state of the environment. Reactive behaviour means adapting to changes in a dynamic and uncertain environment, e.g. focusing attention on detected targets. Probing and caution are properties of dual controllers as described by, for instance, Maybeck [99].

This discussion motivates a decomposition of planning into the following functional and temporal hierarchy:

1. *Long-term platform path planning* considering prior knowledge, threats, pre-planned surveillance requests and time constraints.

2. *Short-term platform path planning* and *long-term sensor planning*, considering the long-term path plan, trajectory smoothing, detected threats, visibility, occlusion, probing, collision avoidance, and dynamic surveillance requests.

3. *Reactive platform path planning* and *short-term sensor planning*, considering the short-term path plan, trajectory smoothing, occlusion, collision avoidance, and gaze planning.

4. *Reactive sensor planning*, considering focus, zoom, contrast, and gaze in addition to the superior path and gaze plan.

The long-term path planning (level 1) is primarily deterministic and can be computed off-line. This plan might be manually prepared. Also the reactive sensor planning (level 4) may be considered separate from the other levels. However, the levels 2-3 represent a very challenging problem due to their stochastic nature, on-line computational demands, and reliance on sensor data analysis. Also, there is a strong coupling between the sensor and path planning, as well as between the planning levels. Consequently, the planning for levels 2-3 must be considered as one single problem.

In this section we have only considered a line search example. We believe that the discussion here can also be applied to the other search patterns; area, strip, and point.

Chapter 3

Related Research Fields and Communities

Planning is a very large and challenging research field with several subfields and several communities as participants. This chapter gives an overview of research fields and communities related, in some way, to planning. The purpose is to show that the planning problem is very complex and that the relationships between different research fields and communities are very intricate.

Research fields and communities can be divided in several ways. We have chosen the following areas: Control, Robotics, Computer Vision, Aerospace, Operations Research, Data Fusion, and Artificial Intelligence.

3.1 The Wide Web of Planning Fields and Communities

In doing this study we have chosen to cast a wide net. Different research communities are researching in similar research areas with different focus. Furthermore, different communities are developing the same tools and techniques independent of others to solve specific problems in each community. The challenge is to achieve an overview of all fields and communities to make it possible to use and combine state-of-the-art research results and knowledge from different fields.

Depending on the researcher's background the focus and impression of the relationships between the fields will vary. In Figure 3.1 *one* view of the intricate relationships between the fields is given. Probably every reader would like to redraw one or more links. The boundary between fields and communities is diffuse and the subdivision in the figure is somewhat arbitrarily. Furthermore, it is sometimes difficult to clearly separate a field, a community and an approach.

Considering this blurred wide web of fields and communities, it is difficult to achieve a balanced view within a reasonable effort. The following sections in this chapter are introducing the identified fields and communities.

3.2 Control

In *automatic control* the goal is to control the behaviour of a dynamic system. In general, the development of a control system starts with a modelling and system identification part, where a model of the system is constructed from physical insight and/or input-output data. The model is then used in the design of a regulator that uses feedback to control the system in a desired way. Control design for stochastic (Gaussian) and linear systems is a well understood research area, but the problem is harder for nonlinear

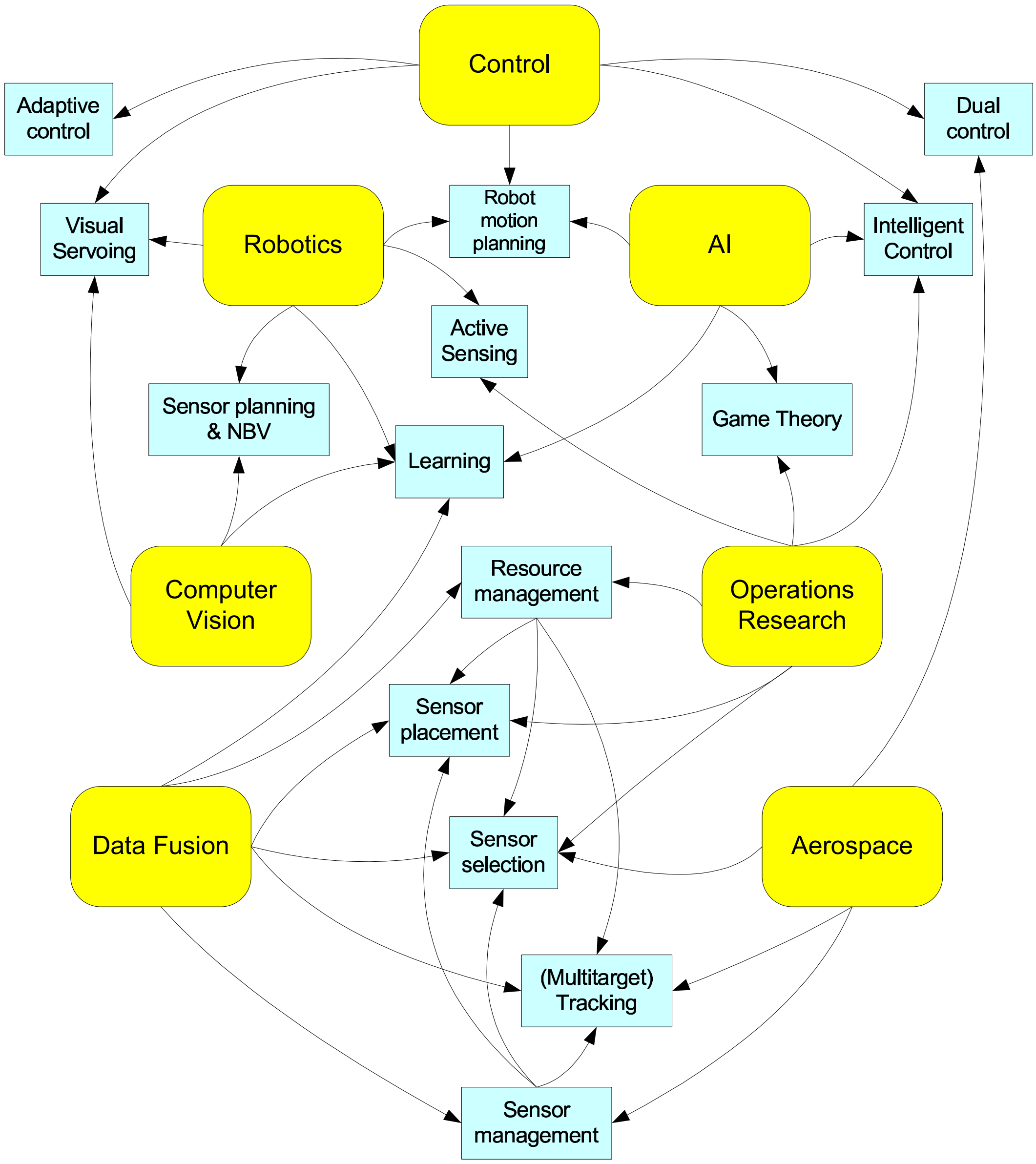


Figure 3.1. The wide web of planning fields and communities.

non-Gaussian systems and systems that change their properties. Some examples of approaches for non-linear and non-constant dynamic systems are

- *Robust control.* The controller is designed for a worst case.
- *Autotuning.* Manual or automatic reconfiguration of the regulator parameter. Often used with PID-controllers.
- *Gain scheduling.* A number of regulators are designed for different work points.
- *Adaptive control.* The system is controlled and identified concurrently. The system identification uses input-output data to estimate a model and this model is used to design the regulator on-line.

3.2.1 Adaptive and Restructurable Control

Adaptive control is interesting since the controller can adapt to systems with unpredictable and time-dependent changes. However, there are, of course, limitations in the adaption possibilities. Only system variations within a predefined class of models (parametric uncertainty models) can be handled. Furthermore, the controller has a limited ability to update the control law. In other words, even small differences between the presumed model and the actual system can result in severe problems and instability. A number of selections must be made: structure of the model, order (number of parameters), time delay, sampling time, structure of the control law, forgetting factor, etc.

A related approach is *restructurable control* which is used for controlling systems that change their dynamic structure in an unpredictable way. The system is monitored and the control law is selected dependent on the current structure. A simple example of restructurable control is a two-legged robot. If both legs can be used then the robot should walk normally, but if one leg is damaged, then the robot should use the other leg and jump to move forward. Thus, two significantly different control strategies are necessary. However, restructurable control suffers from the same problems as in adaptive control.

3.2.2 Intelligent Control

In "conventional" control theory the differential/difference equations framework is the mathematical foundation. However, there are control problem that are hard to describe in a differential/difference equations framework. For instance, problems where the controller must be able to adapt, learn and plan under large uncertainties and large amounts of data. In the field of *Intelligent Control* controllers for these types of problems are developed. Intelligent Control attempts to combine methods from conventional control, operations research and AI to solve more general and complex problems.

Antsaklis [125] mentions some characterizations that are essential of an intelligent control system. "An intelligent system must be highly adaptable to significant unanticipated changes, and so learning is essential. It must exhibit high degree of autonomy in dealing with changes. It must be able to deal with significant complexity, and this leads to certain sparse types of functional architectures such as hierarchies."

Examples of techniques in intelligent control are fuzzy control, neural networks, expert control, and genetic algorithms [128]. Research areas relevant to intelligent control are automatic control, fault diagnosis and reconfiguration, planning, learning, and optimization.

3.2.3 Expert Control

The definition of *expert control* is vague. Properties of expert control are often found in conventional control. Åström [11] provides some visionary goals of expert control, e.g.,

- satisfactory control of a large class of systems, e.g. time-varying, non-linear, disturbance exposed
- minimal prior knowledge required
- intelligent use of prior knowledge
- successive performance increase
- fault detection and diagnosis capability
- the control knowledge and heuristics can be easily examined and modified
- user specification of performance can be in qualitative terms, e.g. "as fast as possible"

Expert Control is related to Fuzzy Control but Expert Control is more general. An expert controller could be rule-based, as a fuzzy controller, but it could also use other knowledge-representation structures, e.g. frames, semantics nets, causal diagrams. Furthermore, the interference process can use more sophisticated methods to determine the decision.

3.2.4 Hybrid Control

Hybrid control is a fairly young branch within control science, concerned with the modelling and control of systems combining continuous and discrete states. In recent years, technological advances, such as faster computers and new sensor technology, have raised demands for more complicated systems. Many systems show such a complexity that combinations of discrete and continuous control are necessary. Consider, for instance, sliding mode controllers where the continuous feedback law is shifting for different regions, thus introducing a discrete state representation within a continuous problem. As in robotics, the resulting systems typically consist of a hierarchical control structure where discrete controllers supervise the operation of a set of continuous servos and controllers. Also, some physical phenomena can only be described by hybrid systems such as contact problems (bounce, backlash) etc. Typical analysis methods are bond graphs and Petri nets.

The area is naturally related to expert control and intelligent control. To some extent Hybrid Control is control science reclaiming territory from computer science, and to some extent it is the fruitful collaboration between researchers from control and computer science. A brief survey of the area is given in [10].

3.2.5 Dual Control

In stochastic systems with large uncertainties, an *optimal* feedback control law will not only steer the system in accordance with the reference signal. In addition, the control law will show *probing* and *caution* behaviour. Probing represents actions to enhance estimation precision in order to improve overall performance in the future. Caution is acting so as to minimize the consequences of erroneous assumptions about the state of the environment. Both these components are often in conflict with the error reducing part of the control law and control laws including this compromise is denoted *dual control*. See Maybeck [99] for details and Nilsson [117] for an application example in dual control. Also see Section 5.2.3.

3.3 Robotics

Robotics is the science and technology of robots, general purpose and programmable machine systems. Robots can be used for a variety of purposes, such as exploration, mining, and manufacturing. The word *robot* was coined by the Czech Karel Capek from the Czech word for forced labor or serf. Today the word robot has several different meanings. In this report we use the word for robot manipulators or unmanned mobile vehicles (mobile robots). Robotics are tightly coupled to large research fields such as automatic control, mechanics, computer vision and AI.

3.3.1 Robot Motion Planning

In almost all robot tasks, the robot must move from an initial configuration to a final configuration. The movement is constrained by dynamic and kinematic constraints of the robot itself and by the environment, e.g. obstacles. *Robot motion planning* is the process of selecting a suitable motion that takes the robot from an initial configuration to a final configuration while ensuring that all constraints are satisfied.

Motion planning can be divided into two groups depending on assumptions about the information available for planning. In *motion planning with complete information* perfect information about the robot and the environment is assumed. In *motion planning with incomplete information*, or sensor-based motion planning, the obstacles can be of arbitrary shape and the input information is, in general, local information from a sensor. Methods are called *dynamic* if body dynamics is taken into account or *kinematic* if the body dynamics is ignored. Motion planning can also be divided into two classes, *holonomic* and *non-holonomic*, depending on if all degrees of freedom can be changed independently or not. Related to the area of robot motion planning is of course also the problem of *obstacle avoidance*. A classical robot motion planning reference is [83]. Also see Section 5.1.

3.3.2 Visual Servoing

Vision has been used with robot manipulators for a long time. In *visual servoing* machine vision is incorporated with the control architecture and the task is to use visual information to control the position and orientation of the robot's end-effector.

The term "visual servo" was introduced by Weiss in the late seventies. The meaning of visual servoing has slightly changed and is today widely used for any system that uses a machine vision system to close the control loop. Visual servoing and *active vision* have much in common. See Section 6.7 for a more detailed introduction to visual servoing.

3.3.3 Active Sensing

Active sensing in robotics is concerned with problems where to position sensors and how to make decisions for next sensing actions, in order to maximize information gain and minimizing cost. Input to the robot is determined by optimizing a criterion, function of costs and utilities.

Mihaylova et al. [106] studies the active sensing problem from an optimal control formulation. They present a couple of performance criteria with respect to uncertainty. Suggested solutions are to use parametrized trajectories and optimize the parameters, or to solve the Partially Observable Markov Decision Processes (POMDP) problem by value iteration (dynamic programming), policy iteration etc.

3.4 Computer Vision

The goal of *Computer Vision* (CV) is to process images or sequences of images, acquired with a camera, in order to produce a scene representation or extract properties of objects in the world.

Much work in CV assumes that the sensor is not fully controllable. The problems of determining viewpoints and sensor parameters that will be most suitable for the task have received less attention. However, the interest for sensor planning in CV has increased.

3.4.1 Sensor Planning

Tarabanis [151] defines *sensor planning* as follows: given information about the environment (e.g. object, sensor) as well as information about the task (e.g. feature detection, object recognition, scene reconstruction) that the sensor system is to accomplish, develop strategies to automatically determine sensor parameter values that achieve this task with a certain degree of satisfaction.

A survey of sensor planning in CV is given in [151]. Approaches in sensor planning are classified by the vision task or by the amount of prior information about the scene. Three areas are identified, *object feature detection*, *model-based object recognition and localization*, and *scene reconstruction*, see Section 6.1.

In the so called *Next-Best-View* (NBV) problem the goal is to build a geometric model of a 3D object. The task is to find optimal parameters of the next view, such as sensor 3D position, viewing direction, field of view, and resolution. Earlier acquired images of the object and its capture positions are given in every step. See Section 6.2 for an introduction to NBV.

3.5 Aerospace

The Aerospace community is the problem owner of many problems related to the sensor and path planning studied in this report. From the Aerospace and Automatic Control communities comes most of the core of the more recently founded Data and Information fusion field. Actually most of the multitarget tracking results are still reported in Aerospace Journals. An early application of "Dual Control" was control of homing missiles [152]. Sensor management for multifunction radars as well as planning and control of UAVs are more recent Aerospace research fields.

3.6 Data Fusion

There are many ways to describe *Data fusion* (or Information fusion or just Fusion). A simple definition could be that Data fusion is the seamless integration of data from disparate sources. The data, collected from different sensors on different platforms in different geographical locations, are blended thematically, so that the differences in resolution and coverage, treatment of a theme, character and artifacts of data collection methods are eliminated. This desirable goal is not yet attained.

3.6.1 Sensor Management and Resource Management

Sensor management and *Resource management* are two subfields of Data fusion that bring a feedback loop into the fusion process to improve the data fusion performance. Sensor management aims at "managing, coordinating and integrating the sensor usage to accomplish specific and dynamic mission objectives" [115].

In [166] issues and approaches to multi-sensor management for information fusion are considered. A top-down problem solving structure is presented and five levels of functionality are identified: Mission planning, Resource deployment, Resource planning, Sensor scheduling, and Sensor control, Figure 3.2.

A survey of sensor management systems is given in [102]. The paper contains two tables with references in the Sensor Management research area, 1) General Sensor Management References, and 2) Summary of Sensor Management Techniques and Applications. The latter lists the following techniques: Heuristic, Expert System, Automatic Control Theory, Utility Theory, Fuzzy Logic, Cognition, Decision Theory, Probability Theory, Stochastic Dynamic Programming, Linear Programming, Neural Networks, Genetic Algorithms, and Information Theory.

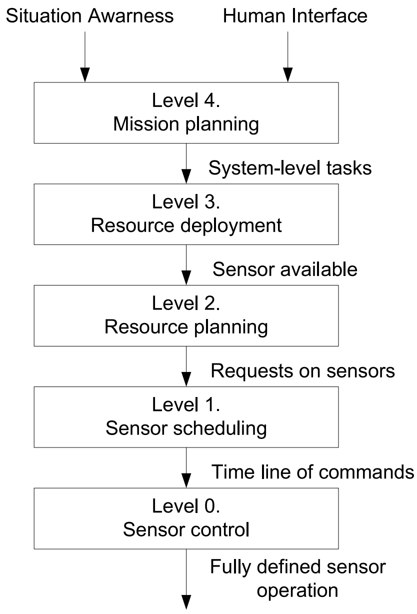


Figure 3.2. Top down problem solving by a sensor manager according to [166].

3.7 Operations Research (OR)

Operations Research (OR) is the science of rational decision making and the study, design and integration of complex situations and systems with the goal of predicting system behaviour and improving or optimizing system performance. *Optimization* is an alternative name of this field.

OR draws upon ideas from engineering, management, mathematics and psychology to contribute to a wide variety of application domains. The field is closely associated with several other fields, such as applied mathematics, computer science, economics, statistics, industrial engineering, and financial engineering. OR started as a field of applied Statistics. During World War II it was used for analyzing the targets of enemy's air attacks, and to determine where to place the radars to have the necessary territory covered.

Examples of techniques used in OR are Linear Programming, Nonlinear programming, Integer Programming, Scheduling, Markov Chains, Queueing Theory, Replacement, Simulation, Stock Control, Dynamic Programming, Decision Theory, and Game Theory.

3.7.1 Search Theory

Theoretical work on how to optimally conduct searches for objects of unknown location was initiated by the U.S. Navy Antisubmarine Warfare Operations Research Group (ASWORG) during the Second World War. Bernard Koopman of ASWORG is generally acknowledged as the founder of classical military search theory. Search theory has since been widely applied in military (in particular naval) operations, typical applications being submarine search and search and rescue operations. Search problems can be broadly categorised into the following types and subcategories:

- One-sided search problems. The searcher can choose a strategy, but the target can not, and does not react to the search. Targets can be stationary or moving.

- Two-sided search problems. Both searcher and target can choose strategies. The search can be cooperative or non-cooperative. In non-cooperative search the target acts to avoid being found (or caught). These problems are often referred to as *pursuit-evasion games*, and are generally the most difficult to analyse.

The most common criteria used in search strategy optimisation are the probability of finding the target in a given time interval and the expected time to find the target. See Section 5.3 for more details on Search theory.

3.8 Artificial Intelligence (AI)

Artificial Intelligence (AI) is the science and engineering of making intelligent machines, especially intelligent computer programs. Intelligence can be considered as the computational part of the ability to achieve goals in the world.

The research in AI started after World War II. A number of people independently begun work on intelligent machines. The English mathematician Alan Turing may have been the first. Turing is known for the *Turing test*; he argued that if a machine could successfully pretend to be human to a observer then the machine should be considered intelligent.

Examples of topics in AI are:

- *Learning from experience.* AI approaches based on, e.g., genetic algorithms and neural nets. Programs can only learn what facts or behaviours their formalism can represent, and unfortunately learning systems are almost all based on very limited abilities to represent information.

- *Planning.* Generating a strategy for achieving a goal, given facts about the world, the effects of actions, the particular situation, and a statement of a goal. In the most common cases, the strategy is just a sequence of actions. In general the decision-making models used admit no uncertainty which is a major disadvantage. Attempts have been made to incorporate uncertainty into the models by making use of techniques from Operations Research, such as Partially Observable Markov Decision Processes (POMDP), Dynamic Programming and Reinforcement Learning. Littman et al. gives a survey of such methods in [92].

Other topics in AI are logical AI, search AI, pattern recognition, representation, inference, common sense knowledge and reasoning, epistemology, ontology, and heuristics.

3.8.1 AI planning

Planning is one of the main areas within the AI community. However, the focus of most of the work does not fulfil our present needs. To quote the Robot Planning Roadmap [16], written by a number of leading AI researchers;
"In contrast to control theory and robotics, robot planning often uses fairly abstract models of the controlled system, at least by standards of control theory. Instead robot planning focusses on issues of task complexity."

The roadmap claims that the complementary strengths of planning and control can be exploited in hybrid systems, and identifies better modelling of the physical system as an important area for future work.

On the higher levels of long term planning where discrete representations are reasonable, the results on Markov Decision Processes (MDP) are of interest. A Markov Decision Process (MDP) is a decision process on a stochastic discrete state model where the state transitions are Markovian, that is, given a control decision, the probabilistic transition model for the new state is only dependent on the decision and the immediately proceeding state. At each moment, given the present state, the optimal decision is thus independent of the history leading up to the present state, thus reducing the complexity of the problem. In [68] a good overview of the work on Partially Observable MDP:s (POMDP) is given. The paper also introduces a graph representation of the possible policies that can be constructed off-line from which a finite memory controller can be extracted. The results and methods used for MDP problems are essentially dynamic programming algorithms and many researchers in the field are also considering reinforcement learning [69] which is a generalization of dynamic programming to problems with unknown state spaces.

The use of hierarchies to battle complexity is presented in [81], where the problem is divided into regions constituting sub goals. Each region constitutes a full MDP problem for solving the sub goal. The regions are connected by peripheral exit/entry states and the whole problem is treated as a semi Markov process.

The concept of any-time algorithms and incremental refinement planning is also of interest in the UAV scenario; the methods in [109] have roots in the AI/planing community [70].

3.8.2 A Roadmap for Research in Robot Planning

A roadmap for research in robot planning [16] has been written by Beetz et al. in the European Network of Excellence in AI Planning (PLANET). The roadmap describes and analyzes the potential for impact of robot planning on autonomous robot and agent control.

The domain of robot planning is defined and the relationships to related fields (control, autonomous agents, and AI) are presented.

- *Autonomous robot control:* "computational task of specifying how the robot is to respond to sensory data in order to accomplish a specified set of jobs".

- *Control program:* "formally representable object with a specified semantics that produces the robot's behavior". Program requirements: real-time concurrent behaviours, manipulate objects in the environment, failure detection, analysis, and recovery.

- *Robot plan:* "that part of the robot's program whose future execution the robot reasons about explicitly". In plan-based control, the plan has two roles: executable description, used to accomplish a job, and syntactic objects, used in the reasoning and (re)planning process.

- *Robot planning:* "automatic generation, refinement, revision, and optimization of robot plans".

- *Plan-based control:* robots "generate control actions by maintaining and executing a plan that is effective and has a high expected utility with respect to the robot's current goal and beliefs".

A *framework* of necessary computational mechanisms is proposed. The blocks are representations of plans, the execution of plans, automatic learning, and reasoning about plans, see Figure 3.3. *Representations* for plan-based control are discussed, in particular planning domain description language (PDDL), plan language, models of dynamical systems, and hybrid representations. *Learning* is emphasised as a critical part of plan-based control since the robot is operating in complex and dynamic environments. Learning is used in two different applications, learning of action models, e.g. duration and probability about action effects, and learning better execution routines, e.g. execution of low-level plans. Autonomous robots in dynamic and uncertain environments must not

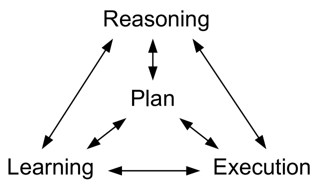


Figure 3.3. A framework for plan-based control. From [16].

only be able to form plans, but also be able to effectively manage and reason about their plans.

Some developments are briefly described, e.g. WITAS, MINERVA, Remote Agent, XAVIER, and CHIP. Challenging application scenarios are presented, e.g. autonomous robots with sophisticated manipulation skills, robot companion, autonomous spacecraft control, and autonomous household robot.

A research agenda is defined and technological topics where progress and breakthroughs can be expected in near-tem, medium-term, and long-term future, are presented. Some examples are:

- *Current and near-term future*: plan execution language with reasoning, richer interaction between planning and execution, time management, planning for human robot interaction, heterogenous representation and reasoning mechanisms for plan-based control, object recognition and manipulation tasks.
- *Medium-term future*: everyday-activity, model-based robot planning systems, computational models of plan debugging, i.e. autonomous service robots acting in realistic environments.
- *Long-term future*: Perform multiple jobs, in human environments, over long time-periods, and get better as they operate.

The roadmap is ended with recommendations to robot vendors, funding agencies and researchers.

3.9 References

Subject/Field/Topic	**References**
Adaptive Control	[14]
Dual Control	[99] [117] [165]
Intelligent and Autonomous Control	[9] [10] [125] [11] [128] [11] [33] [126]
Expert Control	[13] [11] [128]
Robot Motion Planning	[80] [84] [83]
Visual Servoing	[41] [98] [62]
Active Sensing	[106]
Sensor Planning	[151] [162] [36] [147]
NBV	[132] [134]
Sensor Management	[166] [102] [115]
Robot Planning	[16]
AI planning with uncertainty	[92] [81] [69]

Chapter 4

Some Useful Results from Control, Optimization, and Estimation Theory

The concurrent path and sensor planning problem is naturally expressed as an *optimization problem* and the full problem also needs to consider stochastic effects. This chapter first introduces *Optimal Stochastic Control Theory.* When the state space is discrete and deterministic the optimization problem usually reduces to a *graph-search* problem. Finally some results from information theory is introduced to provide a reasonable criterion to optimize. The case of optimization with continuous deterministic state, usually solved as a *calculus of variation problem*, is not considered in this theory section but an application in UAV path planning is given in Section 5.2.1.

4.1 Optimal Stochastic Control Theory

Optimal control is a recurring theme in many solutions to the sensor-planning problem with applications ranging from Operations Research, Control theory through Reinforcement learning. For this reason a short presentation of the theory is in place to provide background to many of the results presented below. There exist many good textbooks on the subject including the books upon which this section is based [99], [19] and [22]. To keep the presentation short we do not present the theory with the full mathematical rigor, instead the interested reader could consult the references above or the more recent work by Bertsekas et al, [23], [24].

Optimal Stochastic Control Theory can be applied to problems both of finite and infinite horizon, but in this work we will be mostly concerned with finite horizon problems.

4.1.1 A typical finite horizon problem formulation

The most important problems in UAV sensor planning can usually be formulated as sequential decision problems. In the sequential decision problem we have a functional that should be optimized, depending on the circumstances it is called the utility, value, cost or loss function. In the following we will refer to the functional as the loss function and assume that the length of the horizon is known to be N. The function is dependent on the chosen decision, control or action sequence $u^{N-1} = \{u(t_1), u(t_2), ..., u(t_{N-1})\}$ and since the problem is stochastic in nature we can only optimize for the expected loss. So we consider the expected loss

$$J(u^{N-1}) = E[L(u^{N-1})] \tag{4.1}$$

To further model the problem, consider a system where the state evolves as the discrete-time stochastic system

$$x(k+1) = f_k\left(x(k), u(k), w(k)\right) \quad k = 0, 1, ..., N-1, \tag{4.2}$$

where w represents the random disturbances. From the system only imperfect information of the state is available through the observations

$$z(k) = h_k\left(x(k), u(k), v(k)\right) \quad k = 0, 1, ..., N-1, \tag{4.3}$$

where v represents the random errors in the observations. Now the loss function is naturally described as a function of the final state. In some cases additional loss on the way is necessary to consider, especially an explicit cost for control is usually present, and in some cases also the state trajectory incurs additional cost.

To fully express the basic problem it is necessary to describe the constraints on the control strategy. One unavoidable constraint for a closed-loop control law is causality i.e. the control at a given time cannot be based on future measurements.

Let the information available to the controller at time k be defined by $I^{k-1} = \{x_0, z(1), u(1), ...z(k-1), u(k-1)\}$ i.e. the history of all previous control and measurements. Here we choose not to consider the measurements at time k available, since there usually is some additional processing time involved before the measurements become available. Then an admissible control law can be defined as a functional of available information i.e. $u^{N-1} = \{u_1(I^0), u_2(I^1), ..., u_{N-1}(I^{N-2})\}$.

The control problem is now represented by 4.2, 4.3 and the expected loss

$$J(x_0, u^N) = E[L_N(x(N)) + \sum_{k=1}^{N-1} L_k(x(k), u(k))|I^0]. \tag{4.4}$$

4.1.2 The principle of optimality and dynamic programming

At the core of optimal stochastic control and dynamic programming is the "optimality principle". As stated By Bellman: "Whatever any initial states and decision [or control law] are, all remaining decision must constitute an optimal policy with regard to the state which results from the first decision" [19]. In essence, if we know the optimal solution J^*_{k+1} of an $N-1$ step problem the optimal solution J^*_k of the corresponding N step problem can be written:

$$J^*_k(x_k, I^k) = \min_{u_k} E\left[L_k\left(x_k, u_k\right) + J^*_{k+1}\left(f\left(x_k, u_k, w_k\right), I^{k+1}\right)|I^k\right] \tag{4.5}$$

In general it is often impossible to find closed form solutions to 4.5. A standard solution is to search for approximate numerical solutions by discretisizing the problem and tabulating the optimal solution $J^*_k(I^k)$ for the discretized problem together with the corresponding optimal control $u^*_k(I^k)$.

As Bellman observed this method is susceptible to the "curse of dimensionality", where larger problems are prohibitive both computationally and in required storage. However, by the evolution of computers there are many problems that can be solved today that were unthinkable when Bellman first observed the "curse of dimensionality".

4.1.3 Linear Quadratic Gaussian Control (LQG)

This section introduces Linear Quadratic Gaussian Control (LQG) as an example of optimal control where explicit solutions are possible. The LQG problem is the resulting problem for a system with linear process- and measurement-models in conjunction with a quadratic criteria and Gaussian uncertainties.

The LQG problem gives insight into the solution steps of the optimal control problem but does not show "dual control" properties. The LQG problem is however of added interest since it can also be used for sub-optimal control in non-linear problems as a solution for a perturbation controller. The perturbations from a given trajectory can be approximated by a linearized system and then LQG theory gives a controller driving the state towards the given trajectory.

Consider the following time-discrete linear system:

$$x(k+1) = Fx(k) + Gu(k) + Bw(k) \tag{4.6}$$

$$z(k+1) = Hx(k) + v(k) \tag{4.7}$$

where x is a state vector, u a vector of control variables, z a vector of observations and w, v normal random variables with zero mean and the covariances

$$\begin{aligned} E\left[w(k)\, w(k)^T\right] &= Q \\ E\left[w(k)\, v(k)^T\right] &= 0 \\ E\left[v(k)\, v(k)^T\right] &= R \end{aligned}$$

The performance criterion to be optimized is the expected loss

$$J = E\left[x(N)\, X_f x(N)^T + \sum_{k=1}^{N-1} x(k)\, X_k x(k)^T + u(k)\, U_k u(k)^T\right] \tag{4.8}$$

where the matrices X_f and X_k are symmetric, nonnegative and U_k is assumed to be positive definite to avoid infinite control signals. The assumption on U_k can be relaxed.

As admissible control we choose $u(k)$ as a function of the accumulated measurement history $Z^k = \{z(1), z(2) ... z(k-1), z(k)\}$. For linear Gaussian models a sufficient statistic for Z^k is the expectation value $\hat{x}(k) = E[x(k)|Z^k]$ calculated by a Kalman filter using gain $K(k)$ [3]. Introduce $I^k = \hat{x}(k)$ as a representation for the sufficient statistic.

From the Kalman filter we know that the state estimate covariance $\Sigma(k)$ is independent of the measurement and state and hence cannot be changed by the control u, i.e. certainty equivalence applies to the LQG problem (see below).

Using Bellman's principle of optimality the solution to the optimization problem can now be found using a DP algorithm starting from the final stage

$$J_N\left(x(N), I^N\right) = E\left[x(N) X_f x(N)^T | I^N\right] = \hat{x}(N) X_f \hat{x}(N)^T + tr X_f \Sigma(N) = \hat{J}_N\left(I^N\right) \tag{4.9}$$

and at each step solving the Bellman equation i.e. the cost to go

$$\begin{aligned} \hat{J}_k\left(I^k\right) &= \min_{u(k)} E\left[x(k) X_k x(k)^T + u(k) U_k u(k) + J_{k+1}\left(x(k+1)\right) | I^k\right] \\ &= \min_{u(k)} \hat{x}(k) X_k \hat{x}(k)^T + tr X_k \Sigma(k) + u(k) U_k u(k) \\ &\quad + E\left[J_{k+1}\left(Fx(k) + Gu(k) + v(k)\right) | I^k\right] \end{aligned} \tag{4.10}$$

It is evident that the final stage 4.9 is a quadratic function. Furthermore, it can be shown that it remains quadratic for all steps so that

$$\begin{aligned} \hat{J}_k\left(I^k\right) &= \hat{x}(k) S(k) \hat{x}(k)^T + s(k) \\ &= \min_{u(k)} \hat{x}(k) X_k \hat{x}(k)^T + tr X_k \Sigma(k) + u(k) U_k u(k) \\ &\quad + \left(F\hat{x}(k) + Gu(k)\right) S(k+1) \left(F\hat{x}(k) + Gu(k)\right)^T \\ &\quad + tr S(k+1) K(t) \left(H\Sigma(k) H^T + R\right) K^T + s(k+1) \end{aligned} \tag{4.11}$$

with the backward recursions

$$\begin{aligned} S(N) &= X_f &(4.12)\\ L(k) &= \left(U_k + G^T S(k+1)G\right)^{-1} G^T S(t+1)F &(4.13)\\ S(k) &= (F - GL(k))^T S(k+1)\,(F - GL(k)) + L(k)U_k L(k)^T + X_k &(4.14)\\ s(N) &= trX_f\Sigma(N) &(4.15)\\ s(k) &= s(k+1) + trS(k+1)K(t)\left(H\Sigma(k)H^T + R\right)K^T + trX_k\Sigma(k) &(4.16)\end{aligned}$$

The minimum is attained for

$$u(k) = -L(k)\hat{x}(k) \qquad (4.17)$$

using the gain $L(k)$ from 4.13.

In a problem with perfect state information the only difference for the control is that the estimate in 4.17 is replaced by the true state. Controllers like this where the optimal control law is independent of the estimator are said to fulfil the *Certainty Equivalence* principle and the separation theorem. The gain in 4.12 does not change when we consider stochastic measurements and process noise, since $s(k)$ is independent of the control and all the effects of uncertainty are collected in $s(k)$. The added cost of not having perfect/deterministic state knowledge is reflected in $s(k)$ where the first term is future costs, the second term is the cost of measurement uncertainty for the current measurement and the last term represents cost of previous uncertainty i.e. the current a priori uncertainty.

4.1.4 Non-linear problems and sub-optimal control

In the linear (LQG) problem above it was possible to find a closed form solution. However, in most cases nonlinearities in the state transition equation, observation equation, loss function or the stochastic models do not allow such a solution and approximations are necessary to achieve “sub-optimal” solutions, capturing at least some of the benefits from dynamic programming. A popular method is to use Assumed Certainty Equivalence (ACE) i.e. to assume that estimation and control can be performed independently. Usually an extended Kalman Filter is used as estimator and if the state transition equation is non-linear an LQG solution can still be used for a perturbation around a given trajectory. Since the perturbation control is driving the state towards the chosen trajectory the linearizations for both the extended Kalman filter and the LQG control law can be calculated off-line. Since it is based on an assumption of “certainty equivalence” this approximation does not invest in control that gives a better future information and for this reason it is not well suited for the purpose of sensor planning by itself.

Another simple assumption is to do feedback control from the current measurement but assuming open loop control over the remaining steps, which result in a greedy algorithm that does not invest in possible future measurements, but it at least takes into account the impact of the current information on the final loss. This method is often referred to as open loop optimal feedback (OLOF) control. A natural extension is then to solve the problem for a fixed limited horizon.

4.1.5 Infinite horizon - Reinforcement learning

When the horizon over which to optimize the criteria is infinite it is often necessary to modify the criteria so that a bounded solution is possible. A typical infinite time criterion is discounted loss:

$$J^*_{k+1}(x_{k+1}, I^{k+1}) = \min_{u_{k+1}} E\left[L_{k+1}\left(x_{k+1}, u_{k+1}\right) + \lambda J^*_k\left(f\left(x_k, u_k, w_k\right), I^k\right) | I^{k+1}\right] \qquad (4.18)$$

where λ is a discount factor. Another loss function is average loss.

To achieve or approach optimal solutions two iterative algorithms have been used at least since the time of Bellman [20]. They are policy iteration and value iteration. It can be shown that both methods asymptotically converge to the optimum. Both methods have seen a revival within the area of reinforcement learning, where it is assumed that the system cannot be well modelled and hence the optimal policy must be learned from experience leading to the popular Q-learning method. The Q-learning is related to policy iteration and if every state-action pair is visited an infinite number of times an optimal policy can be learned. A good survey on reinforcement learning can be found in [69]. The recent popularity of reinforcement learning has led to many efficient algorithms that might be interesting even for a more model-based dynamic programming approach.

4.2 Forward Graph Search Algorithms

In this section we consider a discrete planning problem solved by a forward search in a graph. This section is mainly based on the presentation of discrete planning in [86]. Only *forward* search algorithms are considered below. However, *backward* and *bidirectional* search algorithms exist, but in deterministic graph search problems the solutions are the same.

A graph is a set of vertices (or states), discrete points of the state space, and edges, connections between vertices. Let V and E denote the sets of vertices and edges, respectively, and $|V|$ and $|E|$ are the number of vertices and edges. A weight may be assigned to each edge, representing the cost for moving along the edge. A search algorithm is gradually constructing the minimum cost (or length) path between any two vertices.

4.2.1 General Forward Search Algorithm

A search algorithm is called *systematic* if the algorithm is visiting every state in the graph, assuming the graph is finite. Furthermore, the algorithm should also keep track of all visited states. Thus, a systematic algorithm will be able to tell if a solution exists or not. If the graph is infinite, but has a countable number of states, the systematic requirements must be weakened; if a solution exist the algorithm must find it in finite time.

We define three kinds of state; *unvisited, dead, and alive.* Unvisited states have not been visited yet. Dead states have been visited and will never be visited again since every "neighbour" state has been visited. Alive states have been visited, but have at least one unvisited "neighbour" state.

Now consider the forward searching algorithm (from [86])

```
1   Q.insert(x_I)
2   while Q not empty do
3      x := Q.getFirst()
4      if x ∈ X_G
5         return SUCCESS
6      forall u ∈ U(x)
7         x' := f(x, u)
8         Store pointer x → x'
9         if x' not visited
10           Mark x' as visited
11           Q.insert(x')
12        else
13           Update cost − to − come for x'
14  return FAILURE
```

(4.19)

First an initial state x_I is added to the priority queue Q, containing all "alive" states, line 1. A while loop, lines 2-13, is executed until Q is empty, i.e. the entire graph has been explored. At line 3 the highest priority state is considered and removed from Q. If the state is within the goal states X_G the algorithm is successfully finished, lines 4-5. Otherwise, every possible action $U(x)$ is applied, lines 6-13. For each action u the resulting state $x' = f(x, u)$ is computed, line 7. At lines 8-9 the parent x to x' is stored and x' is determined whether being visited or not. If not visited, mark x' as visited and add to the queue, lines 10-11. If visited the state is dead or alive and already in Q, line 13. If the queue Q becomes empty, no solution exists, line 14.

If the algorithm terminates successfully, the resulting plan can be recovered by tracing children and parents from the final state to the initial state. Some algorithms have a cost $c(e) = c(x, u)$ associated with the edge $e \in E$ between two states. Then, also every state $x \in V$ has an associated cost $C(x)$ to return to the initial state x_I, and this cost may have to be updated if the same state is visited multiple times (line 13).

The major difference between different search algorithms is the priority function that sorts the priority queue Q. In the subsections below some different algorithms are presented. In the first two algorithms the edge costs are not considered, i.e. line 13 can be ignored.

4.2.2 Breadth First Algorithm

The *Breadth first* algorithm specifies the priority queue Q as a first-in-first-out (FIFO) queue. This algorithm is systematic and the search frontier grows uniformly. Thus, the first solution found is a smallest step plan. The computational cost is $O(|V| + |E|)$, assuming that all operations are performed in constant time (which is not the case in practice).

4.2.3 Depth First Algorithm

The *Depth first* algorithm specifies the priority queue Q as a stack, i.e. a last-in-first-out queue. This algorithm is also systematic, but the algorithm is investigating longer plans very early. The computational cost is also $O(|V| + |E|)$.

4.2.4 Dijkstra's Algorithm

Dijkstra's algorithm is an optimal minimum-cost search method. It is a special form of dynamic programming.

As stated above, $C(x)$ denotes the *cost-to-come*, i.e. the cost to reach x from the initial state x_I. Furthermore, let $C^*(x)$ be the optimal (minimum) cost, considering all possible path from x_I to x. The priority queue Q is sorted according to (optimal) cost-to-come. The cost-to-come is computed incrementally during the search according to $C'(x') := C^*(x)+c(e)$. If x' is alive and the new cost-to-come $C'(x')$ is better than the current cost $C(x')$, the cost-to-come for x' must be updated (line 12). The cost-to-come $C(x)$ becomes the optimal cost-to-come $C^*(x)$ when the state x is removed from the queue, i.e. when x is dead. This can be showed by induction.

The computational cost of Dijkstra's algorithm is $O(|V|log|V| + |E|)$ if the priority queue is implemented as a Fibonacci heap and all other operations are performed in constant time.

4.2.5 A^* Algorithm

The A^* algorithm is a variant of Dijkstra's algorithm, which tries to reduce the total number of explored states. Above the cost-to-come $C(x)$ is considered, now let $G(x)$ denote the *cost-to-go* from a state x to the final state x_G. In many problems it is possible to compute a reasonable estimate $\hat{G}^*(x)$ of the optimal cost-to-go $G^*(x)$.

Instead of sorting the priority queue Q according to $C^*(x)$, the A^* algorithm sorts Q according the sum $C^*(x) + \hat{G}^*(x)$, i.e. the estimated optimal cost from the initial state x_I to the final state x_G. This means that the search may not have to be performed on the whole graph, and hence the computational cost is lower. In general, this *heuristic* method is suboptimal, but if $\hat{G}^*(x)$ is guaranteed to be an *under*estimate the solution is optimal. In particular, if the trivial underestimate $\hat{G}^*(x) \equiv 0$ is used, then the A^* algorithm becomes equivalent to Dijkstra's algorithm.

4.3 Information Theory

Technically, *information* is a measure of the accuracy to which the value of a stochastic variable is known. There are two commonly used formal definitions of information, the *Entropic information* and *Fisher information*.

4.3.1 Entropic Information

Entropic information is defined from *entropy*. The entropy (or Shannon information) $h(\mathbf{X})$ associated with a probability distribution $p(\mathbf{x}) = p_{\mathbf{X}}(\mathbf{x})$, where $\mathbf{X}$ is a random variable, is defined as

$$h(\mathbf{X}) \equiv -E\{\log p(\mathbf{x})\} = -\int_{\mathcal{R}^N} p(\mathbf{x}) \log p(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{4.20}$$

(the continuous case). Entropic information $i(\mathbf{x})$ is simply the negative of the entropy

$$i(\mathbf{X}) \equiv -h(\mathbf{X}), \tag{4.21}$$

i.e., information is maximized when entropy is minimized. When the probability distribution of an n-dimensional state $\mathbf{x}$ is Gaussian, with mean $\bar{\mathbf{x}}$ and covariance P, the entropic information becomes

$$i(\mathbf{X}) = -\frac{1}{2} \log[(2\pi e)^n \ \det(\mathbf{P})]. \tag{4.22}$$

Of particular interest in estimation theory is the entropy of the posterior distribution $p(\mathbf{x}|\mathbf{Z}^k)$ where $\mathbf{x}$ is the state vector and $\mathbf{Z}^k = \{\mathbf{z}(k), \mathbf{z}(k-1), ..., \mathbf{z}(1)\}$ is the set of all observations up to time k. Using Bayes' theorem

$$p(\mathbf{x}|\mathbf{Z}^k) = \frac{p(\mathbf{z}(k)|\mathbf{x})p(\mathbf{x}|\mathbf{Z}^{k-1})}{p(\mathbf{z}(k)|\mathbf{Z}^{k-1})} \tag{4.23}$$

(assuming $\mathbf{z}(k)$ is independent of $\mathbf{Z}^{k-1}$, conditioned on $\mathbf{x}$) an interesting recursion is obtained

$$i(\mathbf{X}|\mathbf{Z}^k) = i(\mathbf{X}|\mathbf{Z}^{k-1}) + E\{\log \frac{p(\mathbf{z}(k)|\mathbf{x})}{p(\mathbf{z}(k)|\mathbf{Z}^{k-1})}\} \tag{4.24}$$

The second term on the right hand side is defined as the information about $\mathbf{X}$ contained in the observation $\mathbf{z}(k)$, or the *mutual information* $I(\mathbf{X}, \mathbf{z}(k))$ of $\mathbf{x}$ and $\mathbf{z}(k)$. Thus, the entropic information following an observation is increased by an amount equal to the information inherent in the observation.

4.3.2 Fisher Information

Fisher information gives a measure of the amount of information about $\mathbf{X}$ given observations $\mathbf{Z}^k$ up to time k and the probability distribution $p(\mathbf{Z}^k, \mathbf{x})$. The definition of the Fisher information $\mathcal{I}(k)$ is for random state variable $\mathbf{x}$

$$\mathcal{I}(k) = -E\big\{\nabla_x \nabla_x^T \log p(\mathbf{Z}^k, \mathbf{x})\big\} = -E\big\{\nabla_x \nabla_x^T \log p(\mathbf{x}|\mathbf{Z}^k)\big\} \tag{4.25}$$

and for non-random parameters $\mathbf{x}$ the definition is

$$\mathcal{I}(k) = -E\big\{\nabla_x \nabla_x^T \log p(\mathbf{Z}^k|\mathbf{x})\big\} \tag{4.26}$$

The inverse of the Fisher information is the Cramer-Rao lower bound and it is useful in estimation; it bounds the mean square error of any unbiased estimator of $\mathbf{X}$. In the Gaussian distribution case, the Fisher information becomes simply the inverse of the covariance, i.e. $\mathcal{I}(k) = P^{-1}(k|k)$. The relationship between entropic information and Fisher information for a Gaussian distribution is then

$$i(k) = -\frac{1}{2}\log\Big((2\pi e)^n \ \det P(k|k)\Big) = \frac{1}{2}\log\Big((2\pi e)^{-n} \ \det \mathcal{I}(k)\Big). \tag{4.27}$$

4.3.3 The Information Filter

Consider a linear state transition of the form

$$x(k) = F(k)x(k-1) + w(k) \tag{4.28}$$

where $F(k)$ is the state transition matrix and $w(k)$ is the state transition noise, which is zero-mean, Gaussian, uncorrelated in time, and with covariance matrix $Q(k)$. Let the linear observation model be expressed as

$$z(k) = H(k)x(k) + v(k) \tag{4.29}$$

where $H(k)$ is an observation matrix and $v(k)$ is the observation noise, which is zero-mean, Gaussian, uncorrelated in time, and with covariance matrix $R(k)$. Also assume that $w(k)$ and $v(k)$ are uncorrelated.

The *Information filter* is equivalent to the well known Kalman filter, but instead of maintaining a state vector $x(k|k)$ and its covariance matrix $P(k|k)$, the Information filter is maintaining an information state vector $y(k|k)$ and an information matrix $Y(k|k)$, defined as

$$\begin{aligned} Y(i|j) &\equiv P^{-1}(i|j) \\ y(i|j) &\equiv P^{-1}(i|j)x(i|j) \end{aligned} \tag{4.30}$$

respectively. The Information filter is derived by evaluating the Fisher information matrix (4.25) with the update relationship in (4.23) and assuming Gaussian probability distribution functions. The information matrix and the information state vector are updated according to

$$\begin{aligned} Y(k|k) &= Y(k|k-1) + H(k)^T R^{-1} H(k) \\ y(k|k) &= y(k|k-1) + H(k)^T R^{-1} z(k) \end{aligned} \tag{4.31}$$

(compare with (4.24)) and predicted as

$$\begin{aligned} Y(k|k-1) &= [F(k)Y^{-1}(k-1|k-1)F(k)^T + Q(k)]^{-1} \\ y(k|k-1) &= Y(k|k-1)F(k)Y^{-1}(k-1|k-1)y(k-1|k-1). \end{aligned} \tag{4.32}$$

The Information filter is very suitable for decentralized and/or multi-sensor applications. Assuming that sensor observations are conditionally independent, the fusion step from N sensors becomes remarkably simple

$$\begin{aligned} Y(k|k) &= Y(k|k-1) + \sum_{i=1}^{N} H_i^T(k) R_i^{-1} H_i(k) \\ y(k|k) &= y(k|k-1) + \sum_{i=1}^{N} H_i^T(k) R_i^{-1} z_i(k). \end{aligned} \tag{4.33}$$

See Manyika and Durrant-Whyte [97] and Cover and Thomas [42] for further readings about the Information filter and Information theory.

4.4 Exploration Driven by Uncertainty

In this section an exploration strategy is presented. The strategy considers the problem of "where to measure?" and is driven by the uncertainty of an internal representation of the world. The approach is related to information theory, see Section 4.3. This section is based on [162] and presents the theoretical background of the strategy. A summary of an implementation of the strategy in [162] is presented in Section 6.4.

4.4.1 Linear Model Estimation Basics

The interaction between the sensor and its environment is modelled by linear combination of an arbitrary set of (non-linear) basis functions. Thus

$$d_i = g_i^T(x_i)m \tag{4.34}$$

where x_i is a vector of control parameters, $g_i(x_i)$ is the basis function evaluated at x_i, m is a vector of model parameters, and d_i is a measurement. Assuming we have n measurements, (4.34) can be written as a linear system of equations

$$d = Gm \tag{4.35}$$

where g_i are rows in G. The maximum likelihood estimate $\hat{m}$ of the true parameter vector m_T is given by the pseudo inverse

$$\hat{m} = (G^T G)^{-1} G^T d \tag{4.36}$$

if $d_i = \mathcal{N}(0, \sigma^2)$. The error $\hat{e} = m_T - \hat{m}$ is then $\mathcal{N}(0, \mathrm{C})$ where

$$C = \sigma^2 (G^T G)^{-1} \equiv \sigma^2 H^{-1} \tag{4.37}$$

is the covariance matrix. Furthermore, it can be shown that the ellipsoid

$$\hat{e}^T H \hat{e} = \sigma^2 \chi_\alpha^2 \tag{4.38}$$

will enclose the true model with a probability of α. This ellipsoid is called the *ellipsoid of confidence*.

4.4.2 Reducing Uncertainty

It is obvious that the uncertainty is decreased if a better sensor is used. If σ^2 decreases, the ellipsoid of confidence also decreases. The uncertainty is also affected by the measurement location x_i, since $H = G^T G$ is dependent on x_i. The question is now how to select x_i, or how to select an appropriate uncertainty criterion. The selection of uncertainty criterion is dependent on the operational context. However, a generally useful criterion is based on the Shannon entropy, which measures the amount of information contained in the probability distribution representing the parameter errors. Maximizing the information is equivalent to minimizing the determinant of the parameter covariance, $|C|$.

Thus, the problem is to find a sensor location that will minimize $|C|$. Here the incremental problem is considered, i.e. given the covariance C_n computed from n measurements, which sensor measurement location x_{x+1} will minimize $|C_{n+1}|$? This is equivalent to maximizing $|H_{n+1}|$. Consider the incremental update of H

$$H_{n+1} = H_n + g_{n+1} g_{n+1}^T \tag{4.39}$$

The determinant of (4.39), written in terms of covariance C, is

$$|C_{n+1}| = \frac{|C_n|}{1 + \frac{g_{n+1}^T C_n g_{n+1}}{\sigma^2}} \tag{4.40}$$

Well known results from statistics give us predictions of the measurement, $\hat{d}(x_{n+1})$, that will be obtained at location x_{n+1} and the variance σ_D^2 of $\hat{d}$ as

$$\hat{d}(x_{n+1}) = g_{n+1}^T \hat{m}_n \tag{4.41}$$

and

$$\sigma_D^2(x_{n+1}) = g_{n+1}^T C_n g_{n+1} \tag{4.42}$$

respectively. Thus, (4.40) can be written as

$$\frac{|C_{n+1}|}{|C_n|} = \frac{1}{1 + \sigma_D^2(x_{n+1})/\sigma^2} \tag{4.43}$$

and the problem of minimizing $|C_{n+1}|$ is equivalent to the problem

$$\max_{x_{n+1}} \sigma_D^2(x_{n+1}) \tag{4.44}$$

or expressed in words, *the best location to make a measurement is the location where the ability to predict the measurement is worst.*

Convergence of $|C|$ can be shown in the linear case, if the gaze planning strategy drives the sensor to locations where σ_D is high. In the linear case the optimal locations are independent of the model and can then be computed off-line; this is not the case in the non-linear case.

4.5 Classification of planning algorithms

Methods and approaches, presented in this report, are a heterogeneous mix and it is not a straightforward task to compare different approaches. However, an attempt to classify the methods is made. Since there are numerous variants of each method, it is impossible to make the classification complete. Thus, the classification is only considering the very essential and basic part of each method.

Four different categories, with two or three classes each, have been identified:

- Stochastic vs. Deterministic
- Optimal vs. Suboptimal vs. Heuristics
- Implicit vs. Explicit
- Continuous vs. Discrete vs. Hybrid

4.5.1 Stochastic and Deterministic Planning

In the optimal *stochastic* control problem formulated in Chapter 4.1, the uncertainty in sensor information as well as uncertainty in motion is explicitly modelled and handled by the algorithm. However, Most high level path planners compute paths for a *deterministic* world model, growing obstacles with the expected uncertainty to get safe paths. Such deterministic planners can still handle some uncertainty by replanning when unexpected obstacles are sensed.

4.5.2 Optimal, Suboptimal and Heuristic Planning

In general the planning problem is expressed as an optimization problem. *Optimal* planners are solving the optimization problem in an optimal way. However, an optimal solution of the optimization problem does not guarantee that the solution is solving the original problem in a satisfying way, since the problem description in the optimization process might be a simplification of the original problem.

A *suboptimal* planner does not guarantee that the optimization solution is optimal. An advantage of suboptimal planners, with respect to optimal planners, is that they, in general, can handle more complex problems and are less computationally complex. A class of sub-optimal planners are approximations of the optimal control where the cost-to-go is approximated by a *heuristic* measure.

4.5.3 Implicit and Explicit Planning

In *explicit* planning methods the path between the start and goal configuration is computed explicitly, and in some cases, the associated inputs in the appropriate space. In *implicit* planning methods the motion is not explicitly computed before the motion occurs. Instead, the motion plan specifies how the robot interacts with the environment and how it responds to sensor data, i.e. the planning algorithm tells the robot how to move given its current state and its current knowledge.

4.5.4 Continuous, Discrete and Hybrid Planning

Continuous planners are solving continuous planning problems, i.e. the problem description and the solution are continuous-valued. *Discrete* planners are solving discrete planning problems, e.g. problems where the configuration space or the operational space is discrete-valued. *Hybrid* planners consist of both continuous and discrete parts. A typical hybrid system consists of a continuous process under control and supervision of a discrete process.

Chapter 5

Path Planning

5.1 Robot Motion Planning

Within the robotics community there are many references dealing with path planning or motion planning in a deterministic setting. For manipulators the planning is often performed in configuration space i.e. a space spanned by the joint angles of the robot. A central reference in this area is the book [83]. Many of the problems studied can be characterized as *Labyrinth problems* i.e. given a start location and a goal location find a connecting path through the labyrinth, as an example see [94].

Motion planning can be divided into two groups depending on assumptions about the information available for planning. In *motion planning with complete information* perfect information about the robot and the environment is assumed, shapes of obstacles are formulated algebraically, and the motion planning is an off-line algorithm. In *motion planning with incomplete information* or sensor-based motion planning, the obstacles can be of arbitrary shape and the input information is, in general, local information from a range finder or a vision sensor. This problem naturally fits into the methodology of control theory. Most of the methods in the second category still do not take the uncertainty in the measurements into consideration; the incompleteness of the information is modelled only as availability.

Methods are termed *dynamic* if body dynamics is taken into account, and *kinematic* if the body dynamics is ignored. Motion planning can also be divided into two groups, *holonomic* and *non-holonomic* motion planning. In holonomic motion planning all degrees of freedom can be changed independently, but in nonholonomic motion planning the degrees of freedom are not independent due to kinematic constraints. For example, a car can not rotate without changing its position. See [84] for a presentation of the progress in motion planning research within the 90's, with emphasis on nonholonomic motion planning.

A good example of the state of the art in motion planning is [85]. Here uncertainty in the robot motion is modelled and principles from dynamic programming are used. Unfortunately, from our point of view, optimisation with respect to the sensor information is not included.

Related to the area of robot motion planning is of course also the problem of obstacle avoidance.

5.2 Optimal Control Examples

Optimal Control is presented in Section 4.1. The Optimal Control formulation is very appealing. It is very flexible, in that the framework can deal with state spaces and trajectories that are stochastic or deterministic, implicit or explicit, continuous or discrete.

The problem definition is strictly mathematical and stochastic effects are naturally included. However, the drawback is that problems are, in general, computationally very complex. In practise, the optimal global solution is often impossible to achieve and approximation giving suboptimal solutions are necessary.

There are many examples of optimal control applied to path planning problems. A few examples are listed below and some of them are presented further in the following subsections.

- Robot Motion Planning [80] and UAV Path Planning [27]. See Section 5.2.1
- Vision motion planning with uncertainty [108]. See Section 5.2.2.
- Dual Control in Robotics [117]. See Section 5.2.3.
- Dual control based on approximate posterior density functions [2]. Introduces sum-of-Gaussians (SoG) as a compact representation of posterior density functions.
- Reliable Control of Intelligent Machines [141]. See Section 5.2.4.
- Optimal observer maneuver for bearing-only tracking and area coverage [56], [127], [43]. See Section 5.2.5.
- Distributed Sensor Platform Control [56].

5.2.1 UAV Path Planning

A simple example of UAV path planning with optimal control techniques is presented in [27].

An optimal control problem can be expressed as the optimization problem on the form (compare with (4.9) in Chapter 4.1)

$$\begin{aligned} \min_{u(t)} J &= \int_{t_0}^{t_f} L(x(t),u(t))dt + h(x(t_f)) \\ \dot{x} &= f(x(t),u(t)) \\ g(x(t),u(t)) &= 0 \\ u(t) &\in U, \quad 0 \le t \le t_f \\ x(t_0) &= x_0, \quad x(t_f) = x_f \end{aligned} \tag{5.1}$$

In the path planning context, $u(t)$ represents the control input, x_0 and x_f specify the initial and final state respectively. f describes the dynamics of the system, and g the kinematic constraints on the system. $h(x) \le 0$ and $L(x,u)$ are penalties on the final state and on the trajectory/control, respectively. The goal is to define $f(x,u)$, $g(x,u)$, $h(x)$ and $L(x,u)$ and then minimize J with respect to u.

Consider a 3 DOF example where position (x,y) and heading (ϕ) define the state. Assume that the vehicle is travelling at constant speed and the penalty of the turning rate is represented by $T(u)$. The problem can then be expressed as

$$\begin{aligned} \min_{u(t)} J &= \int_{t_0}^{t_f} (1 + T(u(t)))dt \\ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} &= \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ c_1 \arctan(u) \end{bmatrix} \\ x(t_0) &= x_0, \quad x(t_f) = x_f \end{aligned} \tag{5.2}$$

Kinematic/dynamic/stealth constraints can be incorporated. However, the problem is numerically complex and convergence can not be guaranteed. In practise, a local optimal solution is obtained since the problem is a nonlinear non-convex problem.

5.2.2 Vision Motion Planning with Uncertainty

Miura et al. [108] present an early example of using Dynamic programming to solve a motion planning problem including uncertainty models for vision sensors. Stereo sensor is used to measure the posts of a door-opening, but an association problem complicates the measurement of the width of the opening. By formulating an optimal control problem a compromise between choosing a longer detour or positioning the robot for a better measurement of the door-posts is found by dynamic programming. An interesting continuation of the problem is presented in [109], where *acting-while-planning* and *planning-while-acting* parallelism is used to break down the problem into simplifications where the robot embarks on a path even before a final path has been decided using an iterative refinement anytime-planner. Whenever the iterative refinement planner has committed itself to a single subgoal a planning-while-acting is started to pro-actively plan for the states that can be expected as the result of the information gained at the next sub-goal, for instance the door is too narrow, the door is ok, or the door is still uncertain.

5.2.3 Dual Control in Robotics

In [117] a one-dimensional positioning problem with feedback from a structured light range-camera is studied to give insight into the typical problems of dual control. A dynamic programming algorithm is used under the approximation that sufficient information can be expressed using just the expectation of the position and the corresponding covariance, which gives a two dimensional information state that can be well approximated by a discretisation on a grid without experiencing "the curse of dimensionality". In the formulation both the state dependent measurement uncertainty and limited field of view are considered giving both probing and cautious behaviour of the algorithm. The measurement uncertainty is the same as for a stereo camera, i.e., the standard deviation is proportional to the distance squared. The lessons learned are then applied to control algorithms for control in the plane of mobile robot docking [118]. In the mobile docking case dual programming is not used, instead in one case a certainty equivalence (LQG) controller is designed using the experience from the one dimensional problem. By the use of sum-of-Gaussians (SoG) the probability of fulfilling the task tolerances is approximated. If the probability of success is too low, replanning with a longer horizon is used. One interesting observation is that very few terms in the SoG can cause the controller to be cautious.

5.2.4 Reliable Control of Intelligent Machines

Saridis [141] has developed a method for the design of intelligent control systems based on the minimization of entropy on all levels of a hierarchical control structure. Entropy is used as a measure of the system uncertainties, and by reducing the entropy the stochastic error in the states, which describe the system, are reduced.

Given an explicit task to be executed, an intelligent machine must be able to select a planning strategy such that a desired set of specifications are reliably satisfied. Saridis addresses this issue through a technique termed *reliable control and sensor fusion*, which is a reliability analysis technique based on entropy and therefore invariant with respect to homogeneous coordinate frame transformations. A set of entropy constraints is computed that must be met to ensure reliable operation.

The application studied in [101] is positioning of a robot using a vision system. Here, entropy is not used directly as criteria to minimize but instead as a way to evaluate if a given plan is feasible and reliable. A further discussion on the relation between reliability and entropy can be found in [113].

5.2.5 An Optimal Observer Maneuver Example

To illustrate the use of information in control problems an example is given, based on [56]. A single vehicle with a bearing-only sensor is considered. The task is to localize a stationary feature in the xy-plane. This is done by seeking control actions, i.e. a trajectory, that maximize an information utility. The example illustrates information as a performance metric and the effect of different planning horizons.

Sensor Platform Model

The sensor is attached to a sensor platform moving in the xy-plane with constant velocity V. The location $[x \quad y]^T$ and the direction of the vehicle are described by the state $\mathbf{x}_s(t)$, Figure 5.1. The vehicle's heading is denoted as ψ and the rate of change of the platform

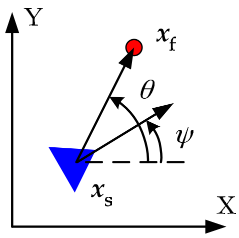


Figure 5.1. 2D observation model

heading $\dot{\psi}$ is the control variable. The equations describing the sensor platform are summarized as

$$\mathbf{x}_s(t) = \begin{bmatrix} x(t) \\ y(t) \\ \psi(t) \end{bmatrix}, \quad \mathbf{x}_s(0) = \mathbf{x}_s^0$$

$$u(t) = \dot{\psi} \tag{5.3}$$

$$\dot{\mathbf{x}}_s(t) = \begin{bmatrix} V\cos(\psi(t)) \\ V\sin(\psi(t)) \\ u(t) \end{bmatrix}.$$

Feature Model

The feature is represented by a stationary point $\mathbf{x}_f = [x_f \quad y_f]^T$ in the xy-plane. The uncertainty of the location is captured in the covariance of a two dimensional Gaussian distribution $P_f(t)$. In the information filter, this is represented by a information matrix $Y(t)$ as the inverse of the covariance as

$$Y(t) = P_f^{-1}(t). \tag{5.4}$$

Since a Gaussian distribution is assumed, the entropic information and the Fisher information are proportional according to (4.27). The feature process model is

$$\dot{\mathbf{x}}_f(t) = \omega(t), \tag{5.5}$$

where the process noise $\omega(t)$ is a zero mean Gaussian process with uncorrelated covariance $Q(t)$.

Sensor Model

The vehicle carries a sensor that makes observations of the feature. The observation is the bearing of the feature, i.e., the relative angle to the feature, Figure 5.1. The observation model is

$$z(t) = h(\mathbf{x}_f, \mathbf{x}_s) \tag{5.6}$$

$$h(t) = \theta(t) - \psi(t) + \nu(t) = \arctan_2(\frac{y_f - y_s}{x_f - x_s}) - \psi(t) + \nu(t) \tag{5.7}$$

where $\nu(t)$ is a zero mean uncorrelated Gaussian process with variance $R = \sigma^2$. Taking the Jacobian with respect to the feature state gives the linearized relationship between the sensed output and the states

$$\begin{aligned} H(t) &= \nabla_{\hat{\mathbf{x}}_\mathbf{f}} \mathbf{h}(\mathbf{x}_f, \mathbf{x}_s) \\ &= [\frac{-(\hat{y}_f - y_s(t))}{(\hat{x}_f - x_s(t))^2 + (\hat{y}_f - y_s(t))^2}, \frac{\hat{x}_f - x_s(t)}{(\hat{x}_f - x_s(t))^2 + (\hat{y}_f - y_s(t))^2}] \\ &= \frac{1}{\hat{r}(t)} \left[-\sin\hat{\theta}(t), \cos\hat{\theta}(t) \right], \end{aligned} \tag{5.8}$$

with $[\hat{x}_f \;\; \hat{y}_f]$ as the estimated feature location, and $[\hat{r} \;\; \hat{\theta}]$ estimation of $[r \;\; \theta]$ respectively. The resulting observed information is derived as (compare with (4.31))

$$I(t) = H^T(t) R^{-1} H(t). \tag{5.9}$$

System Equations

The state of the system consists of the platform model and the information matrix representing the uncertainty of the location of the feature. The update of the vehicle states $\mathbf{x}_s$ is given in (5.3). The update of information can be expressed as (compare with (4.32))

$$\dot{Y}(t) = -Y(t) Q Y(t) + I(t). \tag{5.10}$$

The rate of change in information is some loss due to process noise and the gain of information by observation. The matrices $Y(t)$ and $I(t)$ are symmetric and Q is a diagonal matrix as

$$Y(t) = \begin{bmatrix} \mathrm{Y}_x & \mathrm{Y}_{xy} \\ \mathrm{Y}_{xy} & \mathrm{Y}_y \end{bmatrix}, \; I(t) = \begin{bmatrix} \mathrm{I}_x & \mathrm{I}_{xy} \\ \mathrm{I}_{xy} & \mathrm{I}_y \end{bmatrix} \text{ and } Q = \begin{bmatrix} \mathrm{Q}_x & 0 \\ 0 & \mathrm{Q}_y \end{bmatrix}. \tag{5.11}$$

The feature information matrix is symmetric and it is therefore sufficient to calculate three of four values, which means that the states representing the information would be

$$\mathbf{x}_{info} = \begin{bmatrix} \mathrm{Y}_x \\ \mathrm{Y}_{xy} \\ \mathrm{Y}_y \end{bmatrix}. \tag{5.12}$$

The equations derived for the evolution of the feature information combined with the equations of the vehicle dynamics describe fully the system state and the augmented system equations become

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{x}}_s \\ \dot{\mathbf{x}}_{info} \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\psi}(t) \\ \dot{\mathrm{Y}}_x(t) \\ \dot{\mathrm{Y}}_{xy}(t) \\ \dot{\mathrm{Y}}_y(t) \end{bmatrix} = \begin{bmatrix} V\cos(\psi(t)) \\ V\sin(\psi(t)) \\ u(t) \\ -\mathrm{Y}_x^2(t)\mathrm{Q}_x - \mathrm{Y}_{xy}^2(t)\mathrm{Q}_y + \mathrm{I}_x(t) \\ -\mathrm{Y}_x(t)\mathrm{Q}_x\mathrm{Y}_{xy}(t) - \mathrm{Y}_{xy}(t)\mathrm{Q}_y\mathrm{Y}_y(t) + \mathrm{I}_{xy}(t) \\ -\mathrm{Y}_{xy}^2(t)\mathrm{Q}_x - \mathrm{Y}_y^2(t)\mathrm{Q}_y + \mathrm{I}_y(t) \end{bmatrix}. \tag{5.13}$$

The task is to reduce the uncertainty of the feature by maximizing information. Introduce, for instance, the determinant as utility function

$$J(t_f) = \det(Y(t_f)) = \mathrm{Y}_x(t_f)\mathrm{Y}_y(t_f) - \mathrm{Y}_{xy}^2(t_f). \tag{5.14}$$

and maximize it at the planning horizon t_f.

The problem can be considered as an optimal control problem, since the task is to find a sequence of control signals that maximizes the utility function at the end time t_f. A numerical solution to the optimal control problem can be found by parameterization of the control vector $u(t)$ over the optimization horizon, and the problem is then transformed into a nonlinear programming problem.

Simulation Result

In all simulations the vehicle has constant velocity of 1 [m/s] and initial location in the origin. The position of the feature is [10 10]. A solution is presented in the following figures; Figure 5.2 shows the trajectory of the sensor platform. The observed information and the parameterized control signal are shown in Figure 5.3.

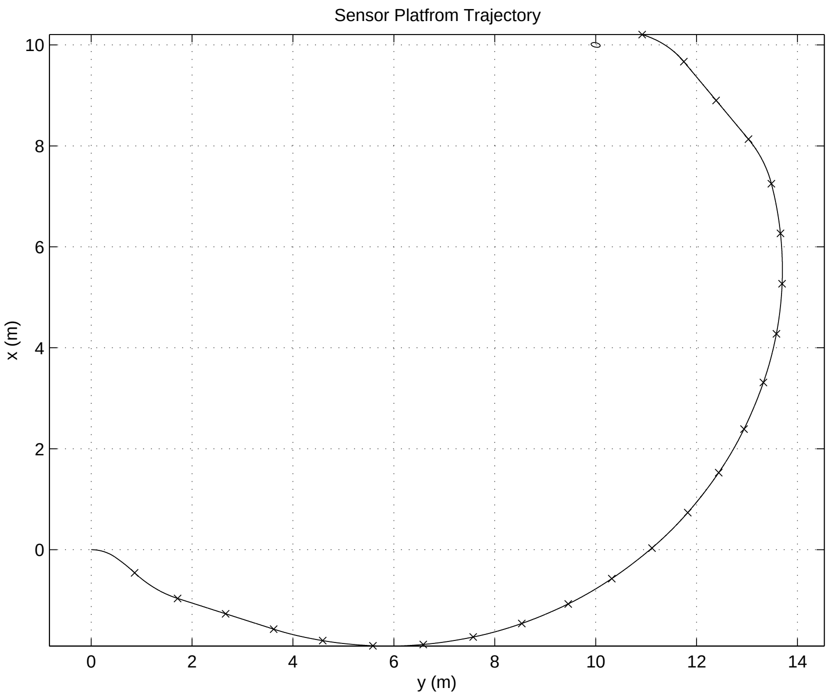


Figure 5.2. Trajectory of sensor platform. Each 'x' mark a new optimization. The feature location's uncertainty is an ellipse about its location.

The sensor platform's trajectory is affected by the planning horizon. Figure 5.4 shows a comparison between short (1 s), intermediate (4 s) and long (8 s) horizon time, Table 5.1. In all cases the total time is 16 s.

Case	**1. (Short)**	**2. (Intermediate)**	**3. (Long)**
Horizon time	1	4	8
Optimization stages	16	4	2
Control parameters	2	5	8

Table 5.1. Details of the three cases used to investigate the effect of different horizon times.

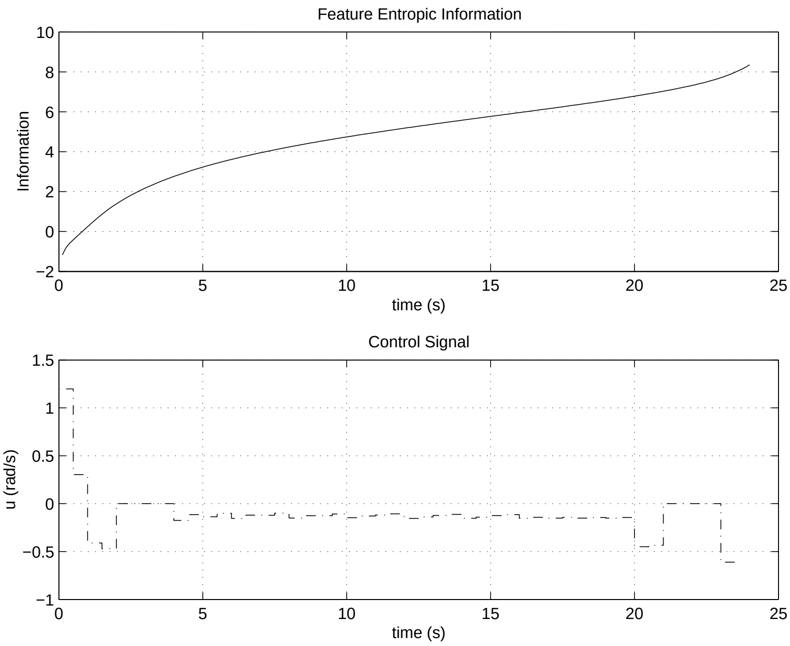


Figure 5.3. Feature entropic information and the parameterized control signal.

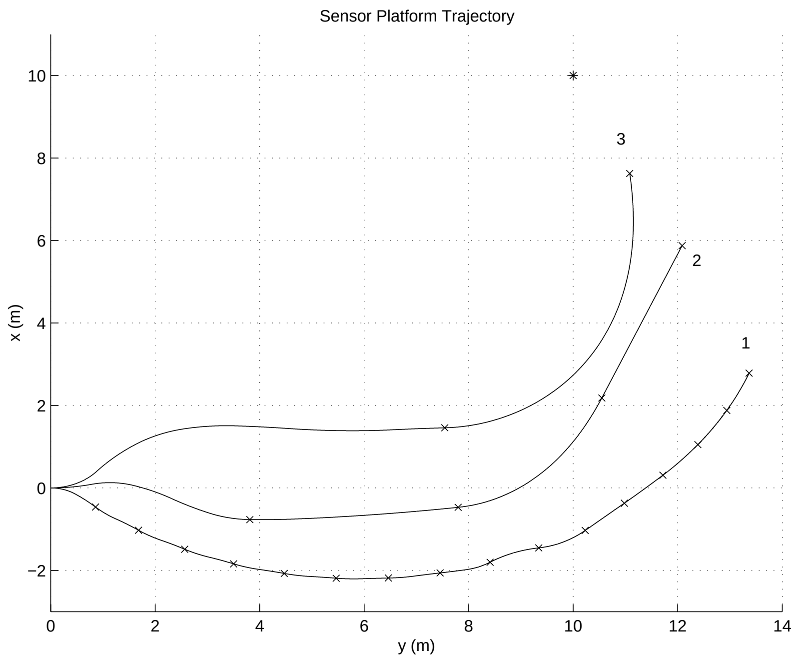


Figure 5.4. The trajectories for platforms with different optimization times.

As can been seen, the platform with long horizon travels more directly towards the feature. In the beginning, it will not gain as much information as the case with short horizon, but since it plans further into the future, the information value at the end of the horizon time will be larger than the information value of the short horizon planner, Figure 5.5. The use of information as a performance index is intuitive, the more information is available of a feature, the more certain is its location. A problem is, however, to choose the time horizon. As the example shows, a longer time increase the information faster, but a drawback is the computational complexity, especially with a large number of control parameters.

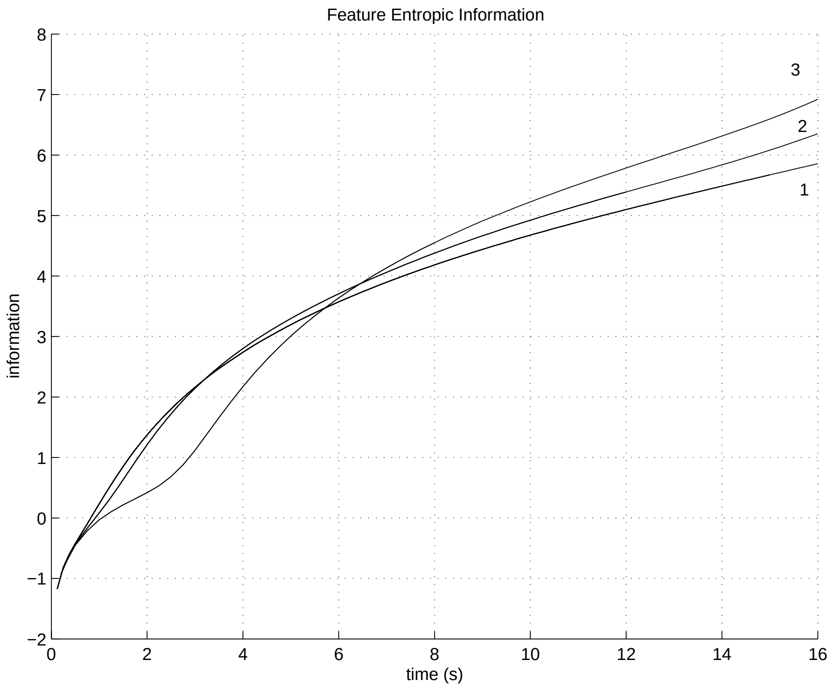


Figure 5.5. The information for platforms with different optimization times.

5.3 Applications of Search Theory

Search theory, a subfield of Operations Research, is concerned with the problem of how to optimally conduct searches for objects of unknown location. This section is largely based on [55].

Search problems can be broadly categorised into the following types and subcategories:

- One-sided search problems. The searcher can choose a strategy, but the target can not, and does not react to the search.
  - Stationary targets.
  - Moving target. In general considerably more difficult than searching for stationary targets, since the target can move into a region that has already been searched.
- Two-sided search problems. Both searcher and target can choose strategies.
  - Cooperative search. Both sides act to increase the chances of detection. Often characterises search and rescue operations.
  - Non-cooperative search. The target acts to avoid being found (or caught). These problems are often referred to as *pursuit-evasion games*, and are generally the most difficult to analyse.

The most common criteria used in search strategy optimisation are

- The probability of finding the target in a given time interval T.
- The expected time to find the target.

The searcher should then either maximise the first or minimise the second criterion. For an evading target the goal is the opposite: to determine a strategy that minimises the probability of being detected, or maximises the expected time to detection.

Classical search theory, as developed by Koopman *et al.* [77], is mainly concerned with determining the optimal *search effort density* for one-sided problems, i.e., how large fraction of the available time T should be spent in each part of the search region, given a prior distribution for the target location. The scenario addressed by Koopman was a patrol aircraft searching for a ship in open sea, using the human eye as sensor. Koopman assumed that the instantaneous probability of sighting a target is proportional to the solid angle subtended by it, which leads to an approximately inverse cube relationship between instantaneous detection probability and range. If the search platform and the target both maintain constant speeds and courses, as in Figure 5.6(a), the geometry of the scenario can always be reduced to that of Figure 5.6(b), involving a stationary target at a lateral range r from the searcher's path. The total detection probability during a

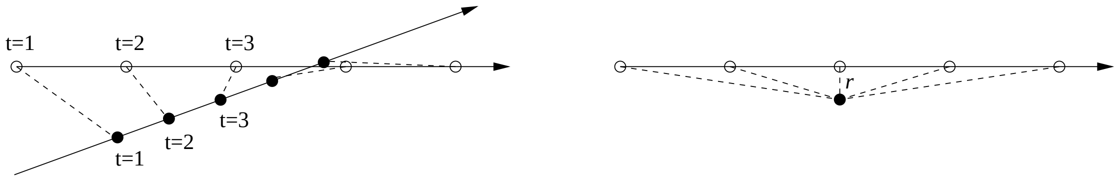


Figure 5.6. Lateral range. Searcher and target passing each other. (b) Equivalent scenario in 'relative motion space', characterised by the lateral range r.

passage (or sweep) can then be expressed as a function $P_D(r)$ of the lateral range to the target, and is obtained by integrating the instantaneous detection probability along the path of the searcher. $P_D(r)$, in turn, can be used to define an *effective search width*, $W = \int_{-\infty}^{\infty} P_D(r)dr$, which corresponds to the sweep width of an ideal sensor with $p_D(r) = 1$ uniformly for $-W/2 \leq r \leq W/2$. If a region of area A is searched randomly in a uniform manner, the cumulative probability of having detected the target through time t is given by $F_D(t) = 1 - \exp(-Wvt/A)$, where v is the speed of the searcher. Using this random search formula, Koopman determined the optimal search effort density for unimodal (e.g., Gaussian) prior target location distributions. The posterior target probability distribution resulting from an unsuccessful search was computed using Bayes' rule. Koopman also considered more realistic search patterns, e.g., parallel tracks, for which he determined the maximum acceptable offset between tracks to achieve a given detection probability. In the 1950s Charnes and Cooper [35] addressed more general prior distributions using discretisation, approximating the search area by a set C of non-overlapping cells, each characterised by area, effective search width, search speed, and target probability. This extension makes the theory much more useful for domains more complex than open sea, e.g., coastal regions and land. The approach is applicable to both optimisation criteria mentioned above.

It was not until the 1980s that results for more complex target motion models appeared, when Brown [31] and Washburn [159] studied the problem of allocating resources between cells for a search divided into N periods, each brief enough for the target to be assumed confined to a single cell for the period. Between successive searches, the target can move between different cells, forming a trajectory $\mathbf{c}_1 \rightarrow \mathbf{c}_2 \rightarrow \cdots \rightarrow \mathbf{c}_N$, where $\mathbf{c}_i \in C$ denotes the cell containing the target during period i. The motion is assumed first order Markovian, which means that the probability of a particular trajectory is given by

$$p(\mathbf{c}_1 \rightarrow \mathbf{c}_2 \rightarrow \cdots \rightarrow \mathbf{c}_N) = p_1(c_1) \prod_{i=1}^{N-1} p_{i,i+1}(c_i, c_{i+1})$$

This distribution is assumed known.

Classical search theory does not take into account the travel time between successive search locations. The first results on path-constrained search appeared in the late 1970s [148]. A typical setting is as for the Markovian motion problem above, but with the added constraint that if the searcher is in cell c at period k, it can only proceed to a subset $S(c, k) \subset C$ at time $k + 1$. Solutions can be found using branch-and-bound procedures [160], which, using the fact that the detection probability can not decrease as the search time is extended, organise the optimisation problem as an efficient tree search.

The Charnes and Cooper approach, complemented with the Markovian motion model and Stewart's search path constraint formulation, represents the state-of-the-art in computer based search planning, as found in tools in operational use today. This method, however, is not without limitations. In particular, the detection model is very simple, and neglects the strong dependency of detection performance on target location, terrain topography, and sensor position that exists in certain applications. For instance, in aerial surveillance of urban or rough terrain, the visibility is highly constrained by occluding objects, which means that correct positioning and pointing of the sensor is critical. This, in turn, implies that the searcher must know its position with high accuracy, which is also necessary in order to accurately geolocate detected targets. Successful planning in rough terrain requires a detailed model of the terrain topography. The searcher must be prepared for deviations from an apriori model, continuously ascertain that the terrain matches the model, and be able to update the model and replan its search path online. The search process becomes integrated with localisation and *exploration* of the search domain. Recently, a few results on integrating path and sensor planning with localisation and terrain topography estimation have been published [110, 96].

5.4 Path Planning based on Visibility

Wang [158] considers the problem of path planning based on visibility for a single mobile observer equipped with a camera. The goal is to obtain maximum visual coverage of a 2D surface in Euclidian space R^3. Two problems, shortest path and maximum visibility, are considered. The existence of solutions to the problems are discussed and numerical algorithms are proposed for approximative problems. The surface is prior knowledge and no dynamic model of the mobile observer is considered, but gradient constraints on the path can be added. Furthermore, the sensor has an unlimited field-of-view.

5.5 Graph-Based Path Planning

A *graph* is a set of vertices, discrete points of the state space, and edges, connections between the vertices. A weight is assigned to each edge, representing the cost for the vehicle to move along the edge. The path planning algorithm is searching for the shortest path (or minimum cost path) between any two vertices.

Dijkstra's algorithm is one method for determining the minimum cost path between any two vertices. An alternative minimum cost algorithm is the Bellman-Ford algorithm. It is more computationally complex than Dijkstra, but Bellman-Ford has the advantage that the cost of the edges can be updated without restarting the algorithm. There also exist heuristic algorithms, e.g. the A^* algorithm that reduce the complexity of the graph search. See Section 4.2 for an introduction to graph search algorithms.

One strategy to decrease the computational complexity is to use a sequence of graphs and successively increase the density of vertices and edges. A coarse flight path is useful as an initial condition for other approaches to the path planning problem, such as optimal control and trajectory smoothing.

There exist several strategies for constructing the graph: visibility graph, Voronoi diagram, free way, silhouette, randomized roadmap ("probabilistic roadmap"), and cell decomposition. Some strategies are presented below.

Graph-based path planning is deterministic, explicit, discrete, and suboptimal. However, an optimal solution, on the set of discrete points, can be obtained in the shortest path problem. Graph-based path planning is suitable for coarse (long-term) path planning, as well as many-DOF robots. The amount of calculations is bounded (finite graph), but large for realistic problems. Further drawbacks are that a static and "deterministic" workspace is assumed and that the solution is quantized.

5.5.1 Voronoi Diagram

A common method for constructing simple graphs is to use Voronoi polygons. Assume that positions of a number of "threats" are given. For every triplet of threats, there exists one circle that passes through all three treats. If no other threat is enclosed by the circle the triplet defines a Delaunay triangle. The set of all Delaunay triangles is called a Delaunay triangulation and the centers of the circles are called Voronoi points. A Voronoi diagram can now be constructed by connecting the Voronoi points. The edges in a Voronoi diagram are equidistant from pairs of threat locations, Figure 5.7. An extension of the Voronoi diagram is the *generalized Voronoi diagram*, which apart from point "threats", can handle lines, obstacles, etc.

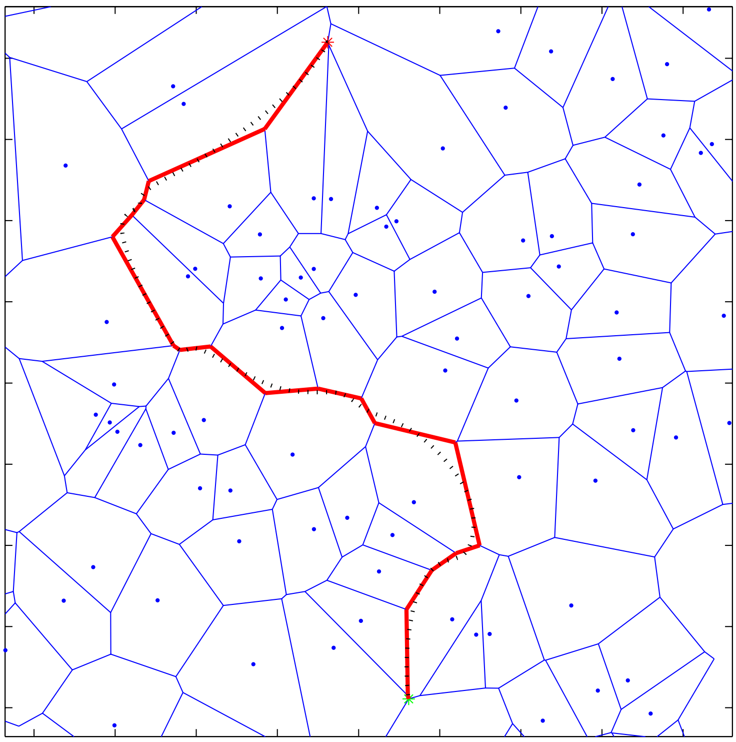


Figure 5.7. Graph planning example based on a Voronoi diagram (in blue) and Dijkstra's algorithm. The edge cost is depending on the edge length and the distance to nearest "threat". The minimum cost path is red and the smoothed path (cubic spline) is black dotted.

5.5.2 Probabilistic Roadmaps

Probabilistic Roadmap Planner (PRM) (or Probabilistic Path Planner) is a planner that can compute collision-free paths for robots of virtually any type moving in a static environment. PRM is particularly interesting for robots with many degrees of freedom. The method proceeds in two phases: a *learning phase* and a *query phase*.

In the learning phase a *roadmap* is built. The roadmap consists of nodes with collision-free configurations and edges corresponding to collision-free paths between adjacent nodes. The roadmap is constructed by repeating the two following steps. Firstly, pick a random collision-free configuration of the robot, and then connect the configurations by using a simple and fast planner, called *local planner*. Thus, the roadmap is built in the robot's configuration space (C-space) and is stored as a undirected graph R. The configurations and the paths are nodes and edges, respectively, of R.

In the *query phase*, a search for a path between an initial and a goal configuration is performed. Firstly, a path from the start and the final configurations to two nearby nodes in the roadmap are found. Then, a graph search in the roadmap is performed, resulting in a sequence of edges connecting these two nodes.

5.5.3 Occupancy Grid

Instead of constructing the graph with vertices and edges directly, the workspace can be modelled as a *probability map* or *occupancy grid* [67]. The probabilistic map consists of cells and an occupancy or risk value is associated with each cells. The map is updated from sensor measurement using Bayes' rule. The path planning problem is now to find the path from a initial cell to a final cell that minimizes the path risk. This problem can be converted to a minimum cost path problem with vertices and edges.

5.6 Path Planning based on Potential Field and Virtual Forces

5.6.1 Potential Field

In the first class of potential field methods a virtual potential field is created that has a sink at the goal and where obstacles are creating repulsing forces. The motion plan is then given by evaluating the effect of the potential field on the robot, usually by a point mass model. For a grid based 2D world model, a good example is the "Navigation function NF1" [83] or the related Distance Transform algorithm [64]. The navigation function is constructed to have only a global minimum at the goal. This global minimum is achieved by labelling each cell with the distance to goal taking the obstacles into account as exemplified in Figure 5.8. In the example a part of a floor-plan of a building at Linköping University is used as a map and a navigation function is calculated for the corresponding grid with a given goal point. Note how one starting point results in a path outside the building.

Problems formulated as potential fields are deterministic and implicit, but they can be either continuous or discrete depending on the potential description. Drawbacks are that kinematic/dynamic constraints must be artificially incorporated and care must be taken to avoid local minima when the potential field is designed. One solution is to consider harmonic potential functions as exemplified by [75], [39].

5.6.2 Virtual Forces Approaches

In the second class of methods, using virtual forces, the path is represented by a chain of point masses connected to one another by springs and dampers. The two ends are connected to the initial and final location respectively. The idea is to force the chain away from threats by using a virtual force field. Each threat is represented by a repulsive force field which pushes away each mass and the chain system will converge to its potential

Figure 5.8. Example of path planning using a navigation function for indoor navigation. Two paths are found by gradient descent on the navigation function from different starting points.

energy minimum. The differential equations are stable initial value problems, i.e. we can expect fast convergence from any initial condition. However, the convergence is not global and a coarse path from a graph approach is often used as initial conditions. Furthermore, in the basic approach it has its lowest resolution in the area where highest resolution is required. An advantage is that kinematic/dynamic constraints can be incorporated. Potential field problems with virtual forces are deterministic, implicit, and suboptimal.

5.7 Trajectory Generation and Smoothing

When using methods that do not consider platform and sensor dynamic constraints explicitly, the resulting path must be modified since the path might not be feasible. In particular discrete paths, e.g., from graph approaches, must be smoothed.

Often the terms *path* and *trajectory* are used as synonyms. However, strictly, a path is a pure geometric description and a trajectory is a path on which a time law is specified, i.e. a trajectory has velocity and acceleration specified for each point but a path does not. Thus, for methods that are only generating a path, the path must be extended to a trajectory.

Simple trajectory generation and smoothing approaches use sequences of splines or arcs/lines together with constant velocity and constrained maximum turn rate.

Advanced methods use dynamic models of the system to compute a feasible trajectory. For instance, a strategy called *on-line trajectory time-scaling* [142] used in robotics takes the overall closed-loop dynamics into account. The desired path is given as $q_d(s)$ where $s = s(t)$ is strictly increasing scalar. The path acceleration $\ddot{s}(t)$ is then calculated based on the input constraints.

A UAV trajectory smoothing algorithm based on nonlinear optimization techniques with *cubic splines* and an initial Voronoi graph can be found in [66]. In [103] a *chain* trajectory smoothing method is applied to a initial Voronoi graph. The method is used for coordinated rendezvous of UAVs and the shape refinement is carried out by treating the chain as a dynamic system with the endpoint positions constrained. Forces are applied to the chain causing it to change shape, first straightening forces to make the path feasible and then repulsive forces are applied to links near threats. A similar approach with virtual forces is presented in [27], see Section 5.6.2.

An approach based on *Pontryagin's Minimum Principle*, suitable for problems with path length, velocity, and curvature constraints, is presented in [4]. This trajectory can be generated in real-time, and is thus suitable for in-flight trajectory generation in a dynamic environment.

Differentially flat systems have useful properties which can be exploited in trajectory generation and tracking for nonlinear control systems. Real-time trajectory generation for mechanical systems are presented in [155] [107].

5.8 UAV Path Planning and Cooperation

This section is a summary of some references on path planning and cooperation for UAV applications.

5.8.1 UAV Path Planning Examples

Several papers use the Voronoi graph approach in the UAV path planning problem. Different smoothing techniques are then applied to refine the path, e.g. differential flatness [89], virtual forces [27], spline based optimization [66], and arcs [34]. See Section 5.7.

Jun et al. [67] consider a grid-based probability map with a minimum cost algorithm and finally path smoothing with arcs. Richards et al. [136] use a mixed-integer linear programming over a short planning horizon, see Section 5.8.2. In [44] a probabilistic approach is presented. A probabilistic map of the operation area is used to determine the conditional probability of a UAV getting disabled by a threat given a certain path. The planning strategy uses local information and has dynamical constraints. Tsourveloudis et al. [116] use a genetic algorithm-based strategy to develop an on-line and an off-line path planner in a 3D rough terrain environment.

A survey of UAV path planning approaches is presented in [27]. Graph (Voronoi), optimal control and virtual forces approaches are described. This survey also considers the stealth properties of simple non-uniform radar detection ranges of the UAV. Norsell [119] presents a more detailed study of aircraft trajectory optimization with tactical constraints.

In [87] a path planning strategy for a UAV tracking a ground vehicle is presented. In general, the UAV trajectory is a sinusoidal trajectory with the amplitude depending on the speed of the vehicle. If the vehicle is moving slower than a certain threshold, or not moving at all, the UAV starts to loiter by following a circular or "rose" curve trajectory.

5.8.2 Trajectory Planning with Mixed-Integer Linear Programming

In [17] [136] a UAV trajectory planning method with obstacle avoidance based on mixed-integer linear programming is presented. An aircraft is modelled as a point mass with constant speed and limited turn rate. A mixed-integer linear program solving a minimum time problem can be formulated with force input and collision avoidance constraints.

To reduce the computational cost, a receding horizon framework is introduced. The optimizing problem is solved over a period of time called *planning horizon*, Figure 5.9. The resulting aircraft input signal from the optimization is applied over a shorter time period, the *execution horizon*. Then a new optimization is performed from the state that is reached. This replanning strategy can handle disturbances and modelling errors.

In general, the planning horizon trajectory is not reaching the goal position. To ensure that aircraft will reach the goal, a cost map is provided to the trajectory design phase. The cost map is an approximate minimum time to reach the goal from each

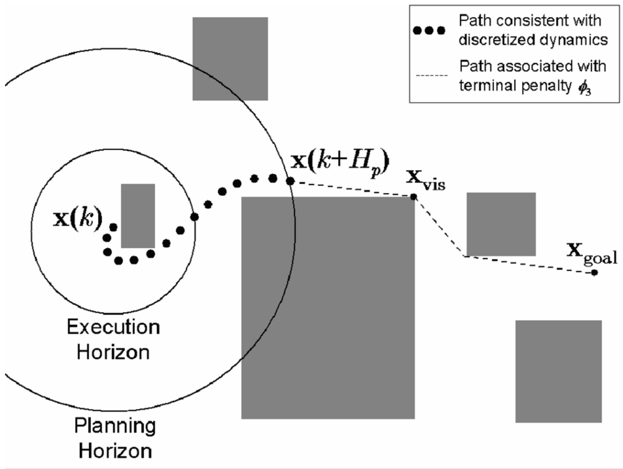


Figure 5.9. The planning and execution horizons. From [17].

graph node and is computed with a modified Dijkstra's algorithm that compensates for reduction in speed when turning.

The approach is deterministic, explicit, and suboptimal. An advantage is the planning horizon which is desirable for complex problems.

5.8.3 Cooperative Time Problems

En example of a cooperative timing problem is when a number of UAVs must maneuver through a dynamically changing threat field to arrive at their destinations simultaneously [66], [103], [104].

In [135] and [18] the problem of task allocation and trajectory planning for a fleet of UAVs are adressed. Simple scenarios with timing constraints, no-fly-zones, and different vehicle dynamics are considered. Two methods are proposed, one based on mixed-integer linear programing and one heuristic approximate approach. The mixed-integer linear program is globally optimal, but computationally complex.

In [34] a hierarchical decentralized decision system is presented that decomposes the timing problem into a rendezvous agent and a trajectory planning agent.

5.8.4 Cooperative Search Problems

Polycarpou et al. [50] address the problem of generating trajectories to follow in order for multiple UAVs to cooperatively search for a target in a given area. Prior knowledge about the distribution of the target is available. A discrete time stochastic decision model is formulated and implemented using a dynamic programming algorithm.

The problem of cooperative search by a team of UAVs with collision-avoidance and communication range constraints is considered in [15]. Two suboptimal methods are developed: the best leader and optimal best path cooperative search algorithms.

5.9 References

Subject/Field/Topic	**References**
Robot Motion Planning	[83] [80] [85] [84] [86]
Labyrinth problem	[94]
Maximum turn strategy	[95]
Optimal control theory	[99] [19] [22] [23] [24]
Information theory	[42], [97]
Path planning	[27] [80]
Optimal observer maneuver	[56] [127]
Distributed sensor platform control	[56]
Vision motion planning with uncertainty	[109] [108]
Reliable control of intelligent machines	[141] [101] [113]
Dual control	[2] [117] [99] [165]
Applications of Search theory	[55] [77] [35] [31] [159] [148] [160]
Integrating planning, localisation, and map-building	[110] [96]
Path Planning based on Visibility	[158]
Voronoi graph	[27]
Hierarchical generalized Voronoi graph	[37]
Probabilistic Roadmap	[74] [73] [84] [144]
Occupancy grid (probability map)	[67]
Potential field and Virtual forces	[83] [64] [75] [27] [133]
Streaming functions	[161]
Mixed-integer linear programming and receding horizon	[136] [17]
UAV trajectory smoothing	[66], [103], [27], [4]
Trajectory time-scaling	[142]
Differentially flatness	[112] [155]
UAV path planning	[27] [89] [66] [34] [67] [136] [44] [116] [119] [87] [4] [103]
UAV cooperative timing problems	[66] [103] [104] [135] [18] [34]
UAV cooperative searching	[50] [15]

Chapter 6

Sensor Planning and Control

A description of the *sensor planning* problem could be: given information about the environment (e.g. object, sensor) as well as information about the task (e.g. feature detection, object recognition, scene reconstruction) that the sensor system is to accomplish, develop strategies to automatically determine sensor parameter values that achieve this task with a certain degree of satisfaction [151]. Sensor planning is related to active vision and active sensing.

6.1 Sensor Planning in Computer Vision

A survey of sensor planning in Computer Vision is given in [151]. The article is limited to sensor planning for vision sensors using high-level model-based approaches. Approaches in sensor planning are classified by the vision task or by the amount of prior information about the scene. Three areas are identified:

- *Object feature detection.* Seeks to automatically determine vision sensor parameter values for which particular features of a known object in a known pose satisfy particular constraints when imaged.

- *Model-based object recognition and localization.* Choose sensing-operations that will prove most useful when trying to identify an object and determine its pose.

- *Scene reconstruction.* A model of the world is incrementally built by sensing the unknown world from effective sensor configurations determined by using the information acquired. Various approaches differ in the criterion with which new sensor configurations are chosen and in the way the multiple views are integrated into the model.

The main focus in [151] is on object feature detection and some approaches are presented:

- *Generate-and-test approach.* The sensor configuration domain is discretized. A number of sensor configurations are generated and evaluated with respect to the task constraints. The sensor parameter determination is formulated as a search over this discretized domain.

- *Synthetic approach.* The task requirements are expressed analytically and the sensor parameter values that satisfy the task constraints are directly determined.

- *Sensor simulation approach.* A scene is visualized given object, sensor, and light source descriptions. Satisfactory sensor configurations can, for instance, be found by a generate-and-test approach.

- *Expert system approach.* Viewing and illumination knowledge is incorporated into an expert system rule base. This approach addresses a high-level aspect of the problem in which a particular technique is chosen.

A analytical approach is elegant and extensible to multiple features. However, the function space is high-dimensional and the approach leads to a constrained nonlinear optimization problem. A discretized approach is more simple and straightforward, but it has a number of drawbacks, such as computational cost, inaccuracy, assumptions of certain viewing parameters, and problems with complex features.

In general, sensor planning methods assume a static and known environment as well as static sensor placement. Important research problems are dynamic sensor planning and feature uncertainty and accuracy.

6.2 Next-Best-View (NBV)

In the so called Next-Best-View (NBV) problem, the goal is to model an object or a scene, with initially unknown geometry. An automatic modelling process contains two parts:

- an incremental modeler that builds solid models,
- a sensor planner that analyzes the current model and computes the next sensor parameters.

The task of the sensor planner is to find optimal parameters of the next view, such as sensor 3d position, viewing direction, field of view, and resolution. Earlier acquired images of the object and its capture positions are given in every step. Limitations in the operating range and the environment add constraints to the problem. For instance, collision avoidance has to be considered.

There are several competing goals. Operational goals could be minimizing the number of capture points, minimizing the total amount of data, or minimizing the length of the path between the capture points. Models synthesis goals could be maximizing the model quality and minimizing the model surface area with insufficient density. The problem is solved by optimizing an objective function in the search space.

A comprehensive review of methods (up to 1996) can be found in [58]. [47] is a recent survey of active 3d object recognition and scene analysis and interpretation; it contains reviews of next view planning approaches. Two NBV references, [134] and [132], are presented below. Also [36] and [162] presented in Section 6.3 and 6.4 respectively, can be considered as next view planning approaches.

A disadvantage of NBV approaches is that the path planning of the sensor is simplified or ignored. Focus is primarily on minimizing the number of views to be able to build a complete model of the target. Furthermore, the methods are most suitable for range images and inherently an outside-looking-in approach. Computation of the bounded visibility volume for each object, or the entire model, is required. NBV approaches are deterministic, suboptimal (heuristic), and discrete, in general.

6.2.1 NBV according to Reed et al.

This section presents the NBV algorithm in [134]. The planner considers three constraints:

- *Sensor imaging constraints.* Limitations on the imaging of a surface in the scene, e.g., the sensor must be within a certain angle of inclination w.r.t. the surface.

- *Scene occlusion constraints.* Limitations due to the fact that some parts of the target surface are occluded from some locations.
- *Sensor placement constraints.* Limitations on the range of sensor placements.

Each of the constraints are represented by a volume, $V_{imaging}$, $V_{occlusion}$ and $V_{placement}$, respectively.

The method is shape-independent and uses a continuous-space representation. Figure 6.1 shows an overview of the sensor planning process. First one or more targets are

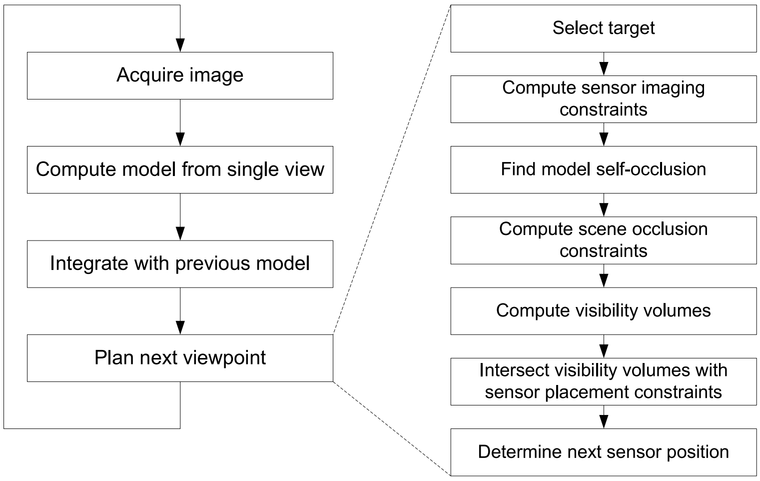


Figure 6.1. Overview of the NBV sensor planning process in [134].

selected. The imaging constraints are determined for each target. The volume $V_{imaging}$ represents locations from which the sensor can image one of the target surfaces. $V_{imaging}$ is dependent on breakdown angle, depth of field, standoff, range, resolution, etc. For each surface in the current composite model, an occlusion volume O_i is computed. The occlusion volume $V_{occlusion}$ is the union of all O_i except the one belonging to the target surface, i.e.

$$V_{occlusion} = \bigcup_{i \neq target} O_i \tag{6.1}$$

The visibility volume describes the set of all sensor positions that have unoccluded view of the target,

$$V_{target} = V_{imaging} - V_{occlusion} \tag{6.2}$$

Finally, the sensor placement constraints are included and the final volume V_{plan} is

$$V_{plan} = V_{target} \bigcap V_{placement} \tag{6.3}$$

which represents accessible, unoccluded positions from which the sensor can properly acquire the target surface.

The V_{plan} for all target surfaces are used to construct a planning histogram where the count represents the number (or area) of target surfaces visible from that sensor location. The next sensor position is then selected as the position of the peak in the planning histogram.

6.2.2 NBV according to Pito

In [132] Pito summarizes some desirable constraints of the next best view and some desirable properties of an NBV algorithm. The next best view should consider

- *Fundamental constraints*, i.e. the need to sample into some unseen portion of the viewing volume.
- *Scanning constraints*, which ensure that the areas to be scanned actually can be scanned.
- *Overlap constraints*, which reduce the search space by using the fact that most range data integrating algorithms perform best when range data overlaps.

Desirable properties of the NBV algorithm are

- *Overlap identification*, identification of rescanned surfaces so it is possible to determine if the registration will succeed.
- *Tolerance*, some way of ensuring that the object has been sampled with at least a minimum accuracy.
- *No assumptions* about the geometry or the topology.
- *Computationally feasible.*
- *Self termination* when the object is completely explored.
- *Generalizable* to any scanning setup.

The remaining of this section is an introduction to the *PS algorithm* developed by Pito [132]. Each time a range image is taken, the viewing volume is partitioned into a *seen volume* and a *void volume* (unseen volume). The scanned surface of the object is called *seen surface* and the surface of the void volume is called *void surface*, Figure 6.2.

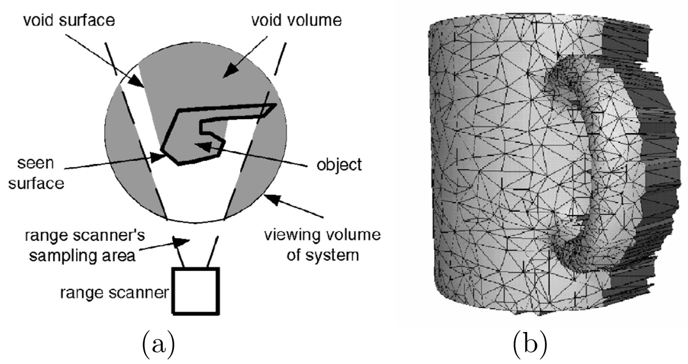


Figure 6.2. (a) Surfaces and volumes in a range image. (b) Mesh model derived from range data. Seen surface in light gray and void surface in dark gray. From [132].

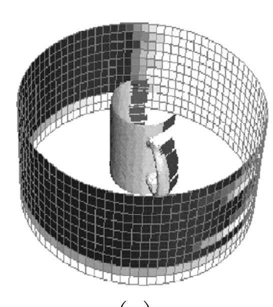
To satisfy the fundamental and overlap constraints the scanner should, from its next position, scan some of the surface already seen and the void surface, i.e. into the void volume. This is repeated, and the void volume is decreased monotonically until all accessible portions are explored and the scanning process is terminated. Thus, to succeed three pieces of information are needed: what must be scanned, what can be scanned from a certain point, and what has already been scanned. The strategy is to position the scanner so that it samples the void surface near the edge of the seen surface.

Ranging ray (RR) is the ray of each sample in a range scanner. The distance to the scanned surface along the ray is determined. Let r denote a RR and n the corresponding surface normal. Furthermore, θ_r is the angle between r and n. If θ_r is greater than a *breakdown angle* θ_B of the scanner, then the measurement is unacceptable. All rays that could sample a planar surface path are called the *observation rays* (OR) of that path.

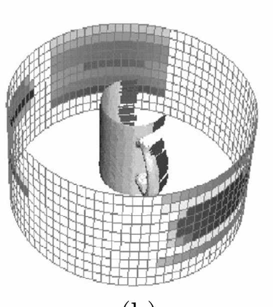
The seen surface is represented by a partial model of the object. A set of *void patches* is attached to the edge of the partial model and oriented to lie on the void surface.

To determine the next-best-view, the PS algorithm makes use of the RRs, the partial model and its void patches. If an RR from a scanner in a particular direction is collinear with an OR, the scanner will sample that surface if a range image was taken from that location. The algorithm searches for scanner positions where the scanner's RRs are collinear with ORs of the void patches and the partial model. A *positional space* facilitates the determination of how many RRs and ORs are collinear. The positional space is composed of two subspaces that record *ranging and observation rays* (RORs). The *position space surface* (PSS) records a point along each ROR. The PSS must therefore enclose the viewing volume, and the selection of the shape depends on the current object and workspace; cylinder and sphere are common shapes. The *positional space direction* PSD records the local direction of the ROR when its it intersects the PSS.

Position Space (PS) is represented by a 4d scalar field, $P(w, y, \theta, \phi)$. The PS representation of a ROR, r, is determined by the intersection (w, y) of r with the PSS and by the local direction (θ, ϕ) of r relative that PSS cell. The image in PS of a scanner at a position x_i is denoted by P_c^i, whereas the void and the seen surface images in PS are denoted by P_v and P_s respectively, Figure 6.3.



(a)



(b)

Figure 6.3. (a) The seen surface image in PS. (b) The void surface image in PS. From [132].

The next-best-view is chosen as the position that samples as many void patches as possible subject to at least a certain amount of the partial model being resampled. Thus

$$\max_{1 \le i \le n} N(i) = \mathrm{o}(\mathrm{o_v(i)}, \mathrm{o_s(i)}) \tag{6.4}$$

where

$$\mathrm{o_v(i)} = \sum_{\mathrm{w}} \sum_{\mathrm{y}} \sum_{\theta} \sum_{\phi} \mathrm{P_c^i P_v} \tag{6.5}$$

and

$$\mathrm{o_s(i)} = \sum_{\mathrm{w}} \sum_{\mathrm{y}} \sum_{\theta} \sum_{\phi} \mathrm{P_c^i P_s} \tag{6.6}$$

are the confidence-weighed area of void patch and partial model visible by the range scanner at position x_i, respectively. The objective function $\mathrm{o}(\mathrm{o_v(i)}, \mathrm{o_v(i)})$ ensures that as many void patches as possible are viewed while ascertaining that at least a minimum amount of the seen surface is resampled. A simple nonlinear function is

$$\mathrm{o(v, s)} = \begin{cases} v, & \text{if s} > \text{t} \\ 0, & \text{otherwise} \end{cases} \tag{6.7}$$

where t is a threshold.

6.3 Sensor Placement with Genetic Algorithms

Chen et al. [36] present a method for automatic sensor placement for model-based robot vision, e.g. industrial inspection. The sensor is moved from one pose to another around the object to observe all features of interest. This involves determination of the optimal sensor placements and a shortest path through these viewpoints. The optimal sensor placement graph is achieved by a *genetic algorithm* in which a min-max criterion is used for the evaluation and a shortest path is determined by Christofides algorithm. This is one of few papers to optimise viewpoint set and trajectory.

A 3D model of the object is assumed available. Viewpoint constraints are:

- Visibility
- Viewing angle
- Field of view
- Resolution
- In-focus or viewing distance
- Overlap
- Occlusion
- Image contrast
- Kinematic reachability

The basic approach can be summarized in the following steps:

1. Generate a number of viewpoints.
2. Reduce redundant viewpoints.
3. If the placement constraints are not satisfied, increase the number of viewpoints.
4. Construct a graph corresponding to the space distribution of the viewpoints.
5. Find a shortest path to optimize robot operations.

Dunn et al. [46] address the problem of how to distribute a given set of viewpoints between multiple sensor platforms (task distribution), and how to move each sensor between its viewpoints (tour planning). The basic approach is to generate a candidate subset of viewpoints for each sensor, and then solve the travelling salesman problem for each sensor. A genetic algorithm is used for searching the space of possible solutions.

6.4 Exploration Driven by Uncertainty - A Gradient Search Approach

Autonomous machines can only be successful if they can handle uncertainties and actively seek out places in the world that have useful information. In [162] an autonomous exploration strategy is developed based on a gradient search method in sensor planning. This section is a brief description of this strategy.

Recall the definitions in Section 4.4. The interaction between the sensor and its environment is modelled by linear combination of an arbitrary set of basis functions. Thus

$$d_i = g_i^T(x_i)m \tag{6.8}$$

where x_i is a vector of control parameters, $g_i(x_i)$ is the basis function evaluated at x_i, m is a vector of model parameters, and d_i is a measurement. The maximum likelihood estimate of m is $\hat{m}$ with covariance C. Given measurements obtained at time $i = 0, ..., n$, the prediction of a measurement $\hat{d}(x_{n+1})$ obtained at location x_{n+1} can be calculated. The variance of $\hat{d}$ is σ_D^2.

The result in Section 4.4 tells us the best place to make a single measurement. The problem is to minimize $|C_{n+1}|$ which is equivalent to

$$\max_{x_{n+1}} \sigma_D^2(x_{n+1}) \tag{6.9}$$

Thus, the best place to make a measurement is the location where the ability to predict the measurement is worst. However, since a single measurement rarely will meet our needs, the problem is to find a sequence of locations until the covariance of the estimated parameters is acceptable.

The linear framework can be used in the non-linear case, if the non-linear model is linearized around $\hat{m}$. However, the implicit assumption that the a posterior parameter distribution is Gaussian is, in general, not true. A consequence is that the maximum likelihood solution is biased. Furthermore, the covariance is an optimistic estimate of the expected parameter errors. These problems can be overcome since the purpose of exploration is to constrain the model parameters and the strategy presented below uses the "direction" of the covariance instead of the magnitude.

In [162] non-linear super-ellipsoids are used to model objects. The iterative strategy is as follows. At each step $\sigma_D^2 = \sigma_D^2(x, \hat{m}_n)$ is computed. The next location x_{n+1} is chosen as that which maximizes σ_D^2 and lies within a region where the linear approximation is valid. The sensor is moved to the new location, a new measurement is made and the model parameters are updated. The process repeats until the model uncertainty meets some acceptability criteria.

It is, however, difficult to determine the region where the linear approximation is valid. Instead, the maximization is performed in the local neighbourhood of the current location and the sensor is moved in the direction of the maximum. Thus, the sensor follows an approximation of the gradient $\nabla_x \sigma_D^2$. In practise, the maximum σ_D^2 is searched on a circle around the current sensor location and the next sensor location is the circle point with maximum σ_D^2. Thus, the distance that the sensor travels each iteration is dependent on the radius of the search circle. The step length has to be feasible with respect to the current model. If the step length is too small the convergence will be slow, and if too large there is a risk of missing important features.

The gradient strategy is compared to an avoidance strategy, where the sensor moves away from the places it has visited previously. In general the gradient strategy performs better than the avoidance strategy. One strength of the gradient strategy is its adaptability.

A monitoring system should verify operational feasibility, detect and correct problems, handle accessibility constraints, and stop collecting data when the specification is met.

The complete explorer consists of two major parts, an vision strategy and a feed-back gaze planning loop. The vision part is a classical strategy with 1) data acquisition and sensor control, 2) visual reconstruction, 3) data fusion, 4) shape analysis and parts decomposition, 5) volumetric modelling. The feed-back loop consists of 1) model validation, 2) gaze planning strategy 3) sensor trajectory planner.

6.5 Decision-Theoretic Sensor Planning

Cook et al. [40] present a decision-theoretic approach to cooperative sensor planning between multiple UGVs. The goal is to maximize the value of information gained by the sensors while maintaining vehicle stealth.

Two capabilities are developed. The first capability selects points along a path or in a bounded region that provide optimal observation locations with respect to a specific area. The second capability selects optimal pan, tilt and field-of-view for each UGV's sensor as it moves in formation.

The approach is based on utility and decision theories. The function to maximise is

$$\mathcal{U}(A,S,P) = k_1\mathcal{U}_{scan}(A,S,P) + k_2\mathcal{U}_{stealth}(S,P) + k_1k_2\mathcal{U}_{scan}(A,S,P)\mathcal{U}_{stealth}(S,P) \tag{6.10}$$

where A is the area to scan, S is the sensor type, P is the sensor position, $\mathcal{U}_{scan}$ is the expected value of information obtained during the sensor action, and $\mathcal{U}_{stealth}$ is the expected utility of maintaining stealth. The constants k_1 and k_2 determine the relative weights between the terms.

The work described in [40] ignores the path planning problem and the probabilities used for the utility measures are static.

6.6 Sensor Management, Allocation and Tracking

A good overview of sensor management for tracking systems is given in [25]. The main thrust in the area is towards multifunction radars as exemplified by [164]. In [164] resource allocation for an electronically scanned antenna (ESA) is considered. The problem can be naturally divided into subtasks of tracking and searching. Each subtask can be optimized locally and the coordination is handled by Lagrange relaxation over the constraints on the available resources (time and energy) in the radar system. An entry level of resource allocation is track maintenance where the sensor resource is conserved by only updating tracks that otherwise would exceed acceptable limits on their covariance. A typical example of algorithms limiting track uncertainty is [93] where, besides direct limits on the uncertainty, the requirements of the controller are also considered. The controller is separated into regions where constant control is possible; whenever there is a possibility for the state to be in multiple regions, a sensor request is issued. More complex models for track loss, where also the association problem is considered, used in[164].

6.7 Visual Servoing and Reactive Control

Vision has been used with robot manipulators for a long time. Traditionally, visual sensing is open-loop, i.e. looking and then moving (*look-then-move*). The accuracy is increased by a visual feedback loop. In *visual servoing*, machine vision is fully incorporated and provides a closed-loop pose (3d position and orientation) control for a robot end-effector.

The advantages of visual feedback are that the accuracy of the system is increased and the system becomes less sensitive to calibration and model errors. The disadvantage is that the introduced feedback can result in system instability.

The vision sensor(s) can be stationary or mounted on the robot's arm (eye-in-hand). The latter configuration is preferable in general, since the system can then provide endpoint-relative positioning information directly in Cartesian or task space.

Figure 6.4 illustrates the difference between look-then-move and visual servoing. In look-then-move the target pose is estimated. Given a model of the environment, the highest level is capable of reasoning about the task. A sequence of movements is then planned and executed. Visual servoing in contrast, is seen as a "low level" shortcut in the hierarchy, i.e., visual servoing should be considered as a *reactive* control instead of planning.

Visual servo structures can be divided into four classes:

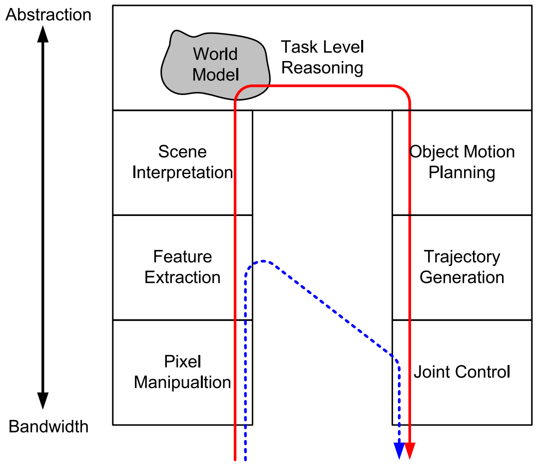


Figure 6.4. Structure of model based robot and vision system (solid line) and the "short-circuited" information flow in a visual servo system (dashed line) [41].

- Dynamic position-based look-and-move.
- Dynamic image-based look-and-move.
- Position-based visual servo (PBVS).
- Image-based visual servo (IBVS).

In *position-based servoing* (PBVS) features are extracted from the image and together with a geometric target model the pose of the target relative the camera is determined, Figure 6.5. A controller then minimizes the error between the estimated and desired poses. Thus, the pose estimation and the control computation are two separate blocks. In PBVS the pose of targets or features is estimated. Examples of methods are

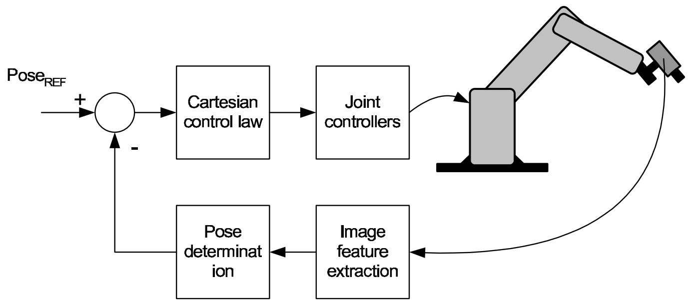


Figure 6.5. Position-based visual servoing.

- Photogrammetric techniques.
- Stereo vision.
- Structure from motion, SFM (or depth from motion).

In *image-based servoing* (IBVS) servoing is done on the basis of images features directly, Figure 6.6. The error is defined in the image parameter space. The *dynamic look-and-move* also make use of joint feedback, but PBVS and IBVS do not. In IBVS the servo controller uses the location of features in the image plane to compute the control input. For example, let the goal be to change the initial camera view, f_0 to a final view, f_1. f_i belongs to the *image feature parameter space*, where *image feature*

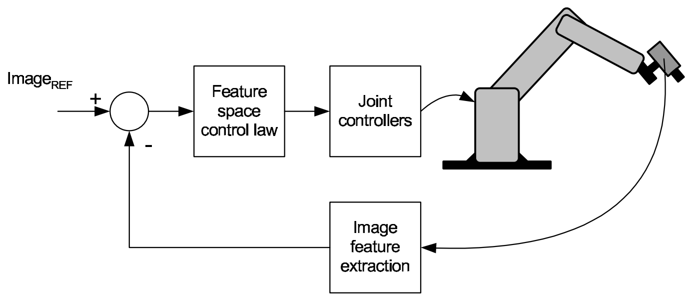


Figure 6.6. Image-based visual servoing.

is any structural feature that can be extracted from the image. x denotes the pose of the end-effector relative the target, it is seen that f_i is nonlinear function of x_i. The relationship may be linearized about an operating point, x_i, resulting in

$$\dot{f} = J_f(x)\dot{x} \tag{6.11}$$

where J is the *image Jacobian* (or feature Jacobian). If J is square and non-singular a simple proportional control law is given by

$$\dot{x} = KJ_f^{-1}(x)(f_f - f(t)) \tag{6.12}$$

where K is a diagonal gain matrix. Assume a constant Jacobian matrix J, defining the relationship between pose rate $\dot{x}$ and the end-effector pose rate $\dot{y}$. Furthermore, define the relationship between the end-effector rates and the manipulator joint rates, $\dot{\theta}$, as

$$\dot{\theta} = J_\theta(\theta)\dot{y} \tag{6.13}$$

A complete control law can then be written as

$$\dot{\theta} = KJ_\theta^{-1}(\theta)JJ_f^{-1}(x)(f_f - f(t)) \tag{6.14}$$

The key problem in IBVS is the estimation of the image Jacobian.

References

See [41] for a detailed presentation of visual servoing. A survey of visual servoing is given in [62]. In [98] a visual servoing approach to problems with collision avoidance, occlusion, and field-of-view in dynamic sensor planning is presented. [129] presents a visual servoing approach to UAV surveillance.

6.8 References

Subject/Field/Topic	**References**
Object feature detection	[151]
NBV	[132] [134] [36] [47]
Sensor placement and shortest path with genetic algorithms	[36] [46]
Exploration driven by uncertainty - gradient approach	[162]
Decision-theoretic sensor planning	[40]
Sensor Management	[166] [102] [115]
Tracking	[25] [93] [164]
Resource Allocation in Airborne Surveillance Radar	[164]
Visual servoing	[41] [62] [98]
Visual servoing approach to UAV surveillance	[129]

Chapter 7

Concurrent Path & Sensor Planning - Discussion and Conclusions

Planning is a very large research field with several subfields and several communities as participants. In this report we focus on path and sensor planning methods, but not even this area is completely covered.

Autonomous concurrent path and sensor planning of a UAV with EO/IR sensors for surveillance and exploration is a very challenging problem. Realistic models of environment, sensors and platforms are very complex, due to the non-linear and stochastic properties of the world. Hence, algorithms and methods solving realistic planning problems are computationally very demanding. Furthermore, the optimal solution is impossible to find, but this is not critical since there are, in principle, an infinite number of solutions that are *sufficiently* good. However, the problem of finding a sufficiently good local optimum is still very hard.

Since the planning problem is very complex, researchers are forced to make simplifications. The simplifications not only depend on the application, but also on the background of the researcher. Different communities are researching in similar areas with different techniques and foci. On the other hand, different communities are developing similar tools and techniques independently of each other to solve specific problems in each community. The challenge in writing this survey has been to achieve an overview of all fields and communities in order to make it possible to use and combine state-of-the-art methods, techniques and knowledge from different fields suitable for our particular planning problem.

In the study of path and sensor planning methods, three major challenges have been identified. Firstly, planning in an stochastic and dynamically changing world is very difficult. Successful management of uncertainties is necessary for successful planning in a realistic and stochastic environment. For instance, *dual control* is a desirable property of the system, see Sections 3.2.5 and 5.2.3. To reduce the complexity of a planning problem, the path and sensor planning are often solved separately, or the focus is either on the path or the sensor planning problem. This can be allowed in some applications, but in others this leads to poor results. Concurrent path and sensor planning are required for good solutions in autonomous low-altitude UAV surveillance and reconnaissance with gimballed EO/IR sensors. Thus, the challenge is to find suitable methods that solve the path and sensor planning simultaneously. The third problem is that all algorithms solving realistic problems become computationally very demanding. This problem, of course, is connected to the ones already mentioned. A *monolithic* planner is probably an unrealistic goal, thus a hierarchical decomposition is required. The issue then is how to decompose the problem into sub-problems that guarantee that the overall objective is achieved. See discussion in [33, 126].

Special approaches developed in separate communities or fields are often not suitable for an integrated system, for instance an autonomous UAV surveillance and reconnaissance system, where a very broad spectrum of tasks must be considered. Figure 7.1 is an attempt to illustrate that different planning and control methods can be placed on the edges of a triangle where the corners represent the research communities/fields Control, AI, and Sensor planning & Computer vision, respectively. However, very few approaches/methods are near the centre of the triangle representing the problem of *Stochastic and Concurrent Sensor and Path Planning.*

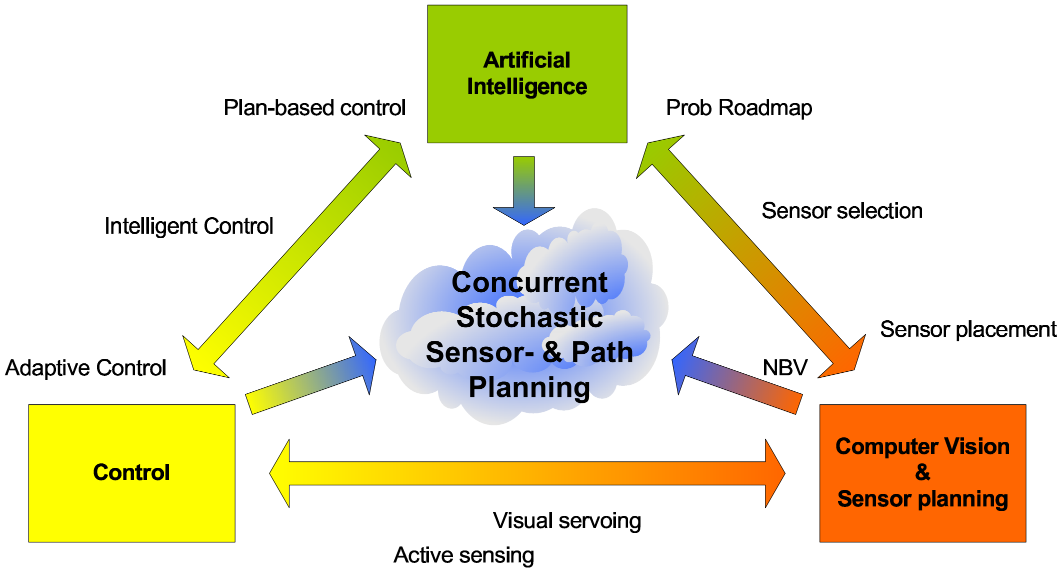


Figure 7.1. Coarse illustration of the relationships between important disciplines and the stochastic and concurrent path and sensor planning.

Despite the sparsity of research involving an integrated view of the stochastic concurrent sensor and path planning problem, we identify some promising results. The fact that many of the references dealing with the path and/or sensor planning problem are of recent origin (published after 2000) indicates that it is a growing field of research, well worthy of consideration in the future.

In Chapters 5 and 6, a number of solutions have been reported under the heading that they are most closely related to, path or sensor planning. Early results with main focus on path planning are presented in Sections 5.2.2 and 5.2.3. Also, the results presented in Section 6.4 consider concurrent path and sensor planning but mainly from a sensor planning point of view. More recently the Australian Center for Field Robotics (ACFR) presented the results reviewed in Section 5.2.5, where an *information theoretic* criterion is used for path planning. The group has also looked at combinations of the needs of navigation, information and exploration in [29] and [96]. A similar approach is taken in a collaboration between Carnegie Mellon University (CMU) and University of Bonn resulting in *coastal navigation* [139]. In coastal navigation the task is to navigate from point A to B and thus no exploratory criteria is needed. The ACFR group has also studied search using information theoretic criteria in [28]. Other groups studying search with relevance to the concurrent path/sensor planning problem are found at CMU [110] and Berkeley [156].

Optimal Control is a very flexible tool for planning and control problems (Section 4.1). In addition, the theory is rather well understood. Optimal Control can be applied to a very broad range of problems, as seen in Section 5.2, and therefore can be used at all planning levels. The problem with Optimal Control approaches is the "curse of dimensionality"; problems must be relatively simple to be solvable. The application of Optimal control to Information theoretic control in [57] has produced results directly applicable to the short term path planning level, see Sections 4.3 and 5.2.5. A unifying formulation of the problem as an Optimal Stochastic Control problem minimizing an information utility is an inviting idea. However, approximations and simplifications are necessary in order to obtain problems that are solvable in real-time. A divide-and-conquer strategy producing a cascaded hierarchy matching the sensor-planning levels is probably necessary.

Further simplification is possible by discretisation within each level, exploring differing resolutions both in time and space. In this task we believe that the *probabilistic roadmap* approach (Section 5.5.2) can be extended to also consider sensing and information concerns besides the strict motion planning. A sensor aspect can be incorporated rather simply by including virtual links representing the sensor target, or by including sensor parameters in the state vector. By this, we expect in a sense to exploit the same possibilities that are used by particle filters in estimation [45].

Similar reasoning can be used as a motivation for *genetic algorithms* [46] [36] (Section 6.3). Some interesting methods for reducing the search/optimisation space are the use of *sum-of-gaussians* [2] [118] to capture the nonlinear effects on the distributions, and also methods of adaptive discretisation[111]. Since sub-optimal solutions might be acceptable, another promising direction is iterative refinement producing *any-time-plans* as in [109]. This approach can be combined with the numerical optimization performed in [57] and [117]. Finally the aspects of exploration, model refinement and model support must be considered [110].

To summarize; the concurrent path and sensor planning problem is very complex and no method has been found that solves the problem in a satisfying way. However, some promising directions are identified and since much research related to this area is in progress, interesting results will hopefully appear during the next few years.

Appendix A

Overview of Related Research in Sweden

A.1 FOI

A.1.1 Department of IR Systems

The system-oriented research at FOI puts special emphasis on EO/IR image processing and control mechanisms to enhance the level of autonomy in UAV surveillance. Some examples of research topics under consideration are

- Development of sensor related network services such as area coverage, detection, association, tracking, geolocation, change detection, and classification.
- Improvement of data acquisition and sensor data analysis, using network distributed prior knowledge and complementary sensor data.
- Incorporation of real-time sensor data analysis in path planning and sensor management to improve the data acquisition process.

See Chapter 2 and [71] [120] [153] [154] for further information.

A.1.2 Department of Data and Information Fusion

The Department of Data and Information Fusion has many projects that consider the problem of sensor management with relevance to this study. Directly identified are two projects on *Decision support for platform operators* and *Decision Support and Information Acquisition*. The department is also hub of the R^2A^2*-lab*.

Decision support for platform operators

This project considers sensor management for operator support on platforms such as fighter jets, where the new multi-function radars give rise to new possibilities. The sensor management is mainly handled as a scheduling problem [150].

Decision Support and Information Acquisition

A direct consequence of the project *Strategisk forskningskärna Informationsfusion* is the support for a PhD student at CAS/KTH working on Decision Support and Information Acquisition [65]. The work has its focus on the higher abstraction level of information-fusion [166], but is moving towards more sensor related issues.

R^2A^2-lab

R^2A^2-lab is an informal cooperation between FOI and the University research groups; R/AMeS at Linköping University and R&A at Luleå University of Technology. The group has been working on sensor-based control of robot platforms since the early 1990s.

A.1.3 Department of Autonomous Systems

Path planning as well as sensing are important parts of most autonomous systems. The sensing research carried out at the Department of Autonomous Systems is focused on the localization and orientation of the vehicle itself. Examples are the work on improving the performance of GPS recievers reported in [21], and the Inertial Navigation System (INS) studies in [88]. Path planning is studied within the projects *Missile Guidance and Control* and *Collaborating missiles systems in a network centric defence*. In [8], optimal control problems associated with the so-called unicycle model, also applicable to fix velocity minimum turn radius UAVs, are studied. One current direction of the *Missile Guidance and Control* project is examining the applicability of methods similar to those used in [119] to problems of a gaming nature. A PhD student supported by the department is working within the area of path planning and autonomous systems at the Division of Optimization and Systems Theory, KTH.

Some work closely related to path planning can also be found in [124], where a theoretical framework is proposed for improving the convergence properties of the Dynamic Window approach, [53], to ground vehicle Obstacle Avoidance. Furthermore, in [163], a mission planning (combined path planning an resource allocation) algorithm for a group of UAVs conducting a Suppression of Enemy Air Defence (SEAD) mission is presented. Finally, a flocking and obstacle avoidance algorithm using methods from sensor area coverage was suggested in [90].

A.2 Linköping University

A.2.1 WITAS

WITAS (Wallenberg laboratory for Information Technology and Autonomous Systems) is engaged in goal-directed basic research in the area of intelligent autonomous vehicles. Its current project focuses on the development of an airborne computer system that is able to make rational decisions about the continued operation of the aircraft, based on various sources of knowledge including pre-stored geographical knowledge, knowledge obtained from vision sensors, and knowledge communicated to it by data link. In October 2003 a successfully demonstration was held at the National Emergency Services school at Revinge, Sweden.[1]

Probabilistic Roadmap Based Planning for an autonomous UAV

The path planner [130, 131] used in the WITAS project is an adaption of the probabilistic roadmap algorithm in [74], see Section 5.5.2. The main parts of the WITAS path planner are illustrated in Figure A.1. The planner is divided into off-line and on-line phases. In the off-line phase a roadmap is generated using a 3D polygon model of the area and helicopter kinematics as inputs. Helicopter configurations (3D position and orientation angle) are first randomly generated, and then collision free and kinematically and dynamically feasible connections between adjacent configurations are generated.

During the run-time phase the initial and final configuration are connected to the roadmap and an A* search is used to generate a trajectory. Finally, the trajectory

[1] The WITAS overview information is from http://www.ida.liu.se/ext/witas/

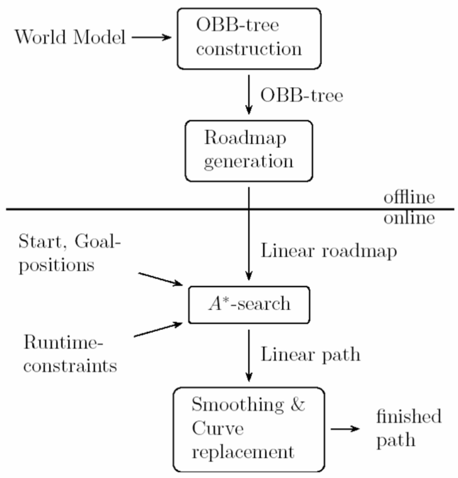


Figure A.1. The WITAS path planner [131].

is smoothed by applying different smoothing operators and checking that the path is collision-free.

A.2.2 Control and Communication

Norrlöf and Nyström are active in the industrial robot field. One activity in the ISIS project aims at developing new algorithms, methods and tools for the generation and optimization of trajectories for industrial robots [122].

Jansson, Karlsson and Gustafsson, in cooperation with Volvo Car Corporation, are studying collision avoidance for passenger cars [72].

A number of Masters theses in aircraft mission replanning have been carried out in cooperation with Saab Gripen. The A^*-algorithm [76] and Model Predictive Control (MPC) [145] have been evaluated.

A.2.3 Robotics and Autonomous Mechanical Systems (R/AMeS)

R/AMeS is a small group of researchers within the larger R^2A^2-lab, see Section A.1.2, that has been working since the early 1990s with robot control and sensor feedback. The work has resulted in two PhDs of direct relevance to the study [117],[121]. The Licentiate thesis [61] is also of some interest since it considers tele-commands using an A^*-method to find a path indoors where safe passages are defined by Voronoi-graphs. The more recent work within the group has been aimed at investigating the benefits of collaboration between a group of robots for navigation[5, 7, 6]. This work has so far not involved sensor feedback but only identified the possible gains by considering synergies within the moving team.

A.3 The Royal Institute of Technology (KTH)

Most of the work relevant to this study is performed within the Centre for Autonomous Systems (CAS); however one project in the Aeronautics department also merits attention.

A.3.1 Aeronautical and Vehicle Engineering

At the Aeronautical and Vehicle Engineering department a recent PhD thesis explores trajectory optimization. In the thesis *Aircraft Trajectory Optimization with Tactical*

Constraints[119], an approach to incorporate tactical constraints in aircraft trajectory optimization is presented.

A.3.2 Centre for Autonomous Systems (CAS)

The Centre for Autonomous Systems (CAS) was founded in 1996 with support from the Swedish Foundation for Strategic Researches (SSF). The center coordinates research within many of the departments of KTH, combining the resources within the groups Computational Vision and Active Perception (CVAP), Mechatronics/Machine elements (DAMEK), Signal Processing and Control (S3), and Optimization and Systems Theory (OPTSYST).

The integration of the groups was achieved through three thematic demonstrators

- The intelligent delivery and service agent
- Autonomous systems for difficult terrain
- Industrial Automation Systems

The Centre received a continued SSF contract in 2002 and now with focus on

- Mobile manipulation in an in-door environment
- Wheeled systems for operation in semi-structured out-door environments

Some references below may not be supported directly from CAS but still exist within the general environment and therefore no effort has been made to explore any such distinction.

Visual servoing for manipulation is one of the activities of CVAP [78]. Since the focus is on manipulation in in-door environment most of this work is not directly applicable to the UAV-surveillance task. Within the S3 group, various aspects of control and signal processing for mobile robots are studied. The main efforts are on mobile robots moving in a 2D world, i.e., were the state is $[x, y, \theta]$, e.g., multi robot tracking [100] where OPTSYST is also involved. In [146] pursuit-evasion games are studied from a communications point of view. Here the entropy is used as a measure to restrict communication for a scenario using simplified sensor-models in a 2D grid-world. The group is also working on path-planning using methods inspired from probabilistic roadmaps such as rapidly exploring random trees (RRT) [91] and probabilistic cell decomposition [149]. Finally, the OPTSYST group is also involved in mobile robot control, path-planning and tracking, using the virtual vehicle approach [49]. Formation control [48], in conjunction with obstacle avoidance [123] is also studied.

One important area is navigation. This area has been of high priority within CAS and though most of the results are not of direct interest for the present study the combination of navigation and exploration as in [143] has relevance. In the outdoor scenario an interesting new project is Urban rescue robotics. In addition to navigation. road-tracking is a planned activity that might give results to follow up on in the future.

A.4 Chalmers University of Technology

At the Department of Signals and Systems resource allocation for an electronically scanned radar antenna (ESA) is considered [164]. The problem can be naturally divided into subtasks of tracking and searching. Each subtask can be optimized locally and the coordination is handled by Lagrange relaxation over the constraints on the available resources (time and energy) in the radar system.

A.5 Lund University

The Division of Robotics has activities in both industrial and service robots. For instance, one topic of interest is increasing the intelligence and autonomy in industrial robot systems, in particular, for welding applications [26] [54]. The Robotics laboratory is also working with visual servoing.

The Department of Automatic Control has interesting research in hybrid [59], adaptive [14], stochastic, and dual control [165]. There is also research being done in integrated control and scheduling [32] and increased autonomy at the local control level [157].

A.6 Örebro University

The Center for Applied Autonomous Sensor Systems (AASS) consists of four research laboratories: The Biologically Inspired Systems Lab, The Mobile Robotics Lab, The Intelligent Control Lab, and The Learning Systems Lab.

One interesting project at the Mobile Robotics Lab is "Generating and executing plans under uncertainty". In [140] Saffiotti discusses the challenge of developing autonomous robots operating without human intervention in real-world environments. Two important issues are how to realize robust motion control and how to flexibly execute navigation plans. Solutions based on fuzzy logic are presented.

An interesting project at the Intelligent Control Lab is "Visual-servoing based simulated flight". Persson [129] has developed a UAV simulation environment to investigate the performance and possibilities of visual-servoing techniques applied to UAV surveillance and reconnaissance missions, e.g., data acquisition and tracking with a vision sensor.

A.7 Luleå University of Technology

At Luleå University of Technology the Robotics and Automation (RA) group has been studying mobile robots for more than a decade. The group is now part of the Embedded Internet System Laboratory (EISLAB) as well as the R^2A^2-lab. The group presented early results on feedback control using active bearings-only sensor and retro-reflective landmarks [63]. The technique has later been industrialized and is used by a company that has the largest market-share of laser guided vehicles, a sub group of automatic trucks or Automatic guided vehicles (AGV:s). Reflective beacons was later also used for reversing with a trailer [82]. Since then the group has been mainly working with navigation using range measuring lasers and the Hough transform, but also with automatic control for passing doorways and following corridors as in [51] [52]. The most recent work is using such feedback for teleoperation by telecommands [138].

A.8 Halmstad University

At the Intelligent Systems Lab Åstrand and Baerveldt are developing an agricultural mobile robot with vision-based perception for mechanical weed control [12]. One goal is to reduce the use of chemicals for weed control. Two vision systems are mounted on the robot. One vision system is pointing forwards and is able to guide the robot along the rows by recognising the row structure formed by the crops. The second vision system is pointing downwards and identifies the crops among the weed plants. A weeding-tool removes the weed within the row of crops.

Kruusmaa is studying path planning for mobile robots. A path selection algorithm for repeated traversal in dynamic environments is presented in [79].

A.9 Umeå University

At the centre for Intelligent Off-road Vehicles (IFOR) one project naturally draws attention, namely *Autonomous Navigation for Forest Machines*. The project is however only recently started so there are not many references to judge from [60],[137], but the project goals are high and it will be interesting to follow up on the project within a few years.

Appendix B

Reference List

Subject/Field/Topic	References	Section
Adaptive Control	[14]	3.2.1
Intelligent and Autonomous Control	[9] [125] [11] [128] [11] [33] [126]	3.2.2
Expert Control	[13] [11] [128]	3.2.3
Optimal Control	[99] [19] [22] [23] [24]	4.1
Path planning	[27] [80]	5.2.1
Optimal observer maneuver	[56] [127]	5.2.5
Distributed sensor platform control	[56]	
Vision motion planning with uncertainty	[109] [108]	5.2.2
Reliable control of intelligent machines	[141] [101] [113]	5.2.4
Dual control	[2] [117] [99] [165] [152]	5.2.3
Information theory	[42], [97]	4.3
Distributed sensor platform control	[97] [57] [56]	
Optimal observer maneuver	[56], [127]	5.2.5
Sensor parameter selection	[43]	
Reliable control of intelligent machines	[141] [101] [113]	5.2.4
Artificial Intelligence		3.8
Robot Planning	[16]	3.8.2
AI planning with uncertainty	[92] [81] [69]	
Any Time Planning	[70]	
Robot Motion Planning	[83] [80] [85] [86] [84]	5.1
Labyrinth problem	[94]	
Maximum turn strategy	[95]	
Elastic Strip	[30]	
Graph Approaches		5.5
Voronoi diagram	[27]	5.5.1
Hierarchical generalized Voronoi graph	[37]	
Probabilistic Roadmap	[74] [73] [84] [144]	5.5.2
Occupancy grid (probability map)	[67]	5.5.3
Potential field and Virtual forces	[83] [64] [75] [27] [133]	5.6
Streaming functions	[161]	

Table B.1. Reference list.

Subject/Field/Topic	References	Section
Exploration		
Exploration and Navigation	[96] [29] [139]	5.3
Search and Exploration	[28] [110] [156]	
Search theory	[55] [77] [35] [31]	5.3
	[159] [148] [160]	
Pursuit Evasion	[156]	
Game theory	[114]	
Target tracking with mobile sensors	[65]	
Trajectory Generation and Cooperation		5.7
UAV trajectory smoothing	[66], [103], [27], [4]	
Time-scaling	[142]	
Differentially flatness	[112] [155] [107]	
UAV Path Planning and Cooperation		
UAV path planning	[27] [89] [66] [34]	
	[67] [136] [44] [116]	
	[119] [87] [4] [103]	5.8.1
UAV cooperative timing problems	[66] [103] [104] [135]	
	[18] [34]	5.8.3
UAV cooperative searching	[50] [15]	5.8.4
Mixed-integer linear programming		
and receding horizon	[136] [17]	5.8.2
Sensor Planning	[151] [47]	6
Object feature detection	[151]	6.1
NBV	[132] [134] [147] [36]	6.2
Sensor placement and shortest path		
with genetic algorithms	[36] [46]	6.3
Exploration driven by uncertainty		
- gradient approach	[162]	6.4
Decision-theoretic sensor planning	[40]	6.5
Path Planning based on Visibility	[158]	
Active Sensing	[106] [105]	3.3.3
Sensor Management		3.6.1
Surveys	[166] [102] [115]	
Tracking	[25] [93] [164]	6.6
Resource Allocation in Airborne		
Surveillance Radar	[164]	
Visual servoing	[41] [62] [98]	6.7
UAV application	[129]	

Table B.2. Reference list continued.

Bibliography

[1] Unmanned Aerial Vehicles roadmap 2002-2027. Report, Office of the Secretary of Defense, Departement of Defense, USA, December 2002. http://www.acq.osd.mil/usd/uav_roadmap.pdf.

[2] D. Alspach. Dual control based on approximate a posteriori density functions. *IEEE Transactions on Automatic Control*, 17(5):689–693, Oct 1972.

[3] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Engelwood Cliffs, 1979.

[4] E.P. Anderson, R.W. Beard, and T.W. McLain. Dynamic trajectory smoothing for UAVs. Technical Report MAGICC-2003-01, February 2003.

[5] L.A.A. Andersson. Information sharing in mobile robot teams, deployment with uncertainty consideration. Licentiate thesis, Linköping University, June 2004.

[6] L.A.A. Andersson and J. Nygårds. Effects on uncertainty utilizing formation-planning in robot teams. In *The 5th Symposium on Intelligent Autonomous Vehicles, IAV 2004*.

[7] L.A.A. Andersson and J. Nygårds. On sensor fusion between a pair of heterogeneous robots. In *The 6th International Conference on Information Fusion, FUSION 2003*, 2003.

[8] D. Anisi. Optimal motion control of a ground vehicle. Scientific report FOI-R–0961–SE, FOI, 2003.

[9] P. J. Antsaklis. Intelligent control. Chapter in book Encyclopedia of Electrical and Electronics Engineering, John Wiley and Sons, Inc., 1997.

[10] P. J. Antsaklis and X. D. Koutsoukos. On Hybrid Control of Complex Systems: A survey. Technical report, University of Notre Dame, 1997. ISIS-97-017.

[11] P. J. Antsaklis and K.M. Passino, editors. *An introduction to intelligent and autonomous control*. Kluwer Academic Publishers, 1992.

[12] B. Åstrand and A.J. Baerveldt. Design of an agricultural mobile robot for mechanical weed control. In *4th European Workshop on Advanced Mobile Robots, Eurobot '01*, Sep 2001.

[13] K. J. Åström, J. J. Anton, and K.-E. Årzén. Expert control. *Automatica*, 22(3):277–286, 1986.

[14] K.J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 1995.

[15] R.W. Beard and T.W. McLain. Multiple UAV cooperative search under collision avoidance and limited range communication constraints. In *IEEE Conference on Decision and Control*, pages 25–30, Dec 2003.

[16] M. Beetz. A roadmap for research in robot planning. European Network of Excellence in AI Planning (PLANET), online at http://www9.in.tum.de/research/tcu/ (2004-05-06).

[17] J. Bellingham, A. Richards, and J.P. How. Receding horizon control of autonomous aerial vehicles. In *IEEE American Control Conference*, May 2002.

[18] J. Bellingham, M. Tillerson, A. Richards, and J. How. Multi-task allocation and path planning for cooperating UAVs. In *Conference on Coordination, Control and Optimization*, November 2001.

[19] R. Bellman. *Dynamic Programming.* Princeton Univ. Press, 1957.

[20] R. Bellman. *Adaptive Control Processes.* Princeton University Press, 1961.

[21] F Berefelt, B Boberg, F Eklöf, J Malmström, L Pääjärvi, P Strömbäck, and S-L Wirkander. INS/GPS integration with adaptive beamforming. Technical Report FOI-S–0984–SE, FOI, 2003.

[22] D. P. Bertsekas. *Dynamic Programming and Stochastic Control.* Academic Press, 1976.

[23] D. P. Bertsekas. *Dynamic Programming and Optimal Control: 2nd Edition.* Athena Scientific, 2001.

[24] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

[25] S.S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House,Inc., 1999.

[26] G. Bolmsjö, M. Olsson, and K. Brink. *Increased autonomy in industrial robotic systems: A framework*, chapter 2. 1999. ISBN 0-7923-5580-6.

[27] S.A. Bortoff. Path planning for unmanned air vehicles. In *American Control Conference*, pages 364–368, June 2000.

[28] F. Bourgault, T. Furukawa, and H.F. Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS03)*, volume 1, pages 48–53, Oct. 2003.

[29] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System IROS02*, volume 1, pages 540–545, 30 Sept.-5 Oct. 2002.

[30] O. Brock and O. Khatib. Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths. In *IEEE International Conference on Robotics and Automation*, 2000.

[31] S. S. Brown. Optimal search for a moving target in discrete time and space. *Operations Research*, 28:1275–1289, 1980.

[32] A. Cervin. *Integrated Control and Real-Time Scheduling.* PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, April 2003.

[33] P.R. Chandler and M. Pachter. Research issues in autonomous control of tactical UAVs. In *American Control Conference*, volume 1, pages 394 – 398, June 1998.

[34] P.R. Chandler, M. Pachter, and S. Rasmussen. UAV cooperative control. In *American Control Conference*, volume 1, pages 50 – 55, June 2001.

[35] A. Charnes and W. W. Cooper. The theory of search: Optimum distribution of search effort. *Management Science*, 5:44–50, 1953.

[36] S. Y. Chen and Y. F. Li. Automatic sensor placement for model-based robot vision. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 34(1):393–408, 2004.

[37] H. Choset and J. Burdick. Sensor-based exploration: the hierarchical generalized Voronoi graph. *International Journal of Robotics Research*, 19(2):96–125, Feb 2000.

[38] B. T. Clough. Metrics, schmetrics! How do You determine a UAV's autonomy anyway? In *Proceedings of the 2002 PerMIS Workshop*. NIST, August 2002. NIST Special Publication 990.

[39] C. I. Connolly and R. A. Grupen. On the applications of harmonic functions to robotics. *Journal of Robotics Systems*, 10(7):931–946, 1993.

[40] D.J. Cook, P. Gmytrasiewicz, and L.B. Holder. Decision-theoretic cooperative sensor planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1013–1023, 1996.

[41] P. I. Corke. *Visual control of robots*. Research Studied Press Ltd., 1996.

[42] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & sons, Inc., 1991.

[43] 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*, 24(2):145–157, 2002.

[44] A Dogan. Probabilistic path planning for UAVs. In *AIAA "Unmanned Unlimited" Systems, Technologies, and Operations - Aerospace, Land, and Sea Conference and Workshop & Exhibit*, Sep 2003.

[45] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[46] E. Dunn and G. Olague. Evolutionary computation for sensor planning: The task distribution plan. *EURASIP Journal on Applied Signal Processing*, 2003(8):748–756, 2003.

[47] S. Dutta Roy, S. Chaudhury, and S. Banerjee. Active recognition through next view planning: A survey. *Pattern Recognition*, 37(3):429–446, March 2004.

[48] M. Egerstedt and X. Hu. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951, December 2001.

[49] M. Egerstedt, X. Hu, and A. Stotsky. Control of mobile platforms using a virtual vehicle approach. *IEEE Transactions on Automatic Control*, 46(11):1777–1782, November 2001.

[50] M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand. Cooperative control for multiple autonomous UAVs searching for targets. In *IEEE Conference on Decision and Control*, volume 3, pages 2823 – 2828, Dec 2002.

[51] J. Forsberg, U. Larsson, P. Ahman, and Å. Wernersson. The Hough transform inside the feedback loop of a mobile robot. In *IEEE International Conference on Robotics and Automation*, pages 791–798, 1993.

[52] J. Forsberg, U. Larsson, and Å. Wernersson. Mobile robot navigation using the range-weighted Hough transform. *IEEE Robotics & Automation Magazine*, 2(1):18–26, March 1995.

[53] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, March 1997.

[54] M. Fridenfalk. *Development of intelligent robot systems based on sensor control.* PhD thesis, Lund University, Lund, Sweden, 2003. ISBN 91-628-5550-6.

[55] J. R. Frost and L. D. Stone. Review of search theory: Advances and applications to search and rescue decision support. Technical Report CG-D-15-01, U.S. Coast Guard Research and Development Center, Sep 2001.

[56] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms.* PhD thesis, The University of Sydney, ACFR, March 2002.

[57] B. Grocholsky, H. Durrant-Whyte, and P. Gibbens. An information-theoretic approach to decentralized control of multiple autonomous flight vehicles. In *Sensor Fusion and Decentralized Control in Robotic Systems III*, Nov 2000.

[58] O. Hall-Holt. Scrutinizing real-world geometry: the Next Best View. Online at http://www-graphics.stanford.edu/˜olaf/nbv.html (2004-06-01).

[59] S. Hedlund. *Computational Methods for Optimal Control of Hybrid Systems.* PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, May 2003.

[60] T. Hellström. Autonomous navigation for forest machines. Technical report, Umeå University, 2002.

[61] T. Högström. Telecommands for semi autonomous control in telerobotics. Licentiate thesis, Linköping University, May 1997.

[62] S. Hutchinson, G. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, Oct 1996.

[63] K. Hyyppä. Luleå turbo turtle (ltt). In *IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS89)*, 1989.

[64] R.A. Jarvis. Distance transform based collision-free path planning for robot. In *Advanced Mobile Robots*, pages 3–31. World Scientific Publishing, 1994.

[65] R. Johansson. Information acquision in data fusion systems. Licentiate thesis, KTH, TRITA-NA-0328, November 2003.

[66] K. B. Judd and T. W. McLain. Spline based path planning for unmanned air vehicles. In *AIAA Guidance, Navigation and Control Conference*, volume AIAA-2001-4238, August 2001.

[67] M. Jun and R. D´Andrea. *Path planning for unmanned aerial vehicles in uncertain and adversarial environments.* Kluwer, 2002. In S. Butenko, R. Murphey, and P. Pardalos, editors, Cooperative Control: Models, Applications and Algorithms, chapter 6, pages 95-111.

[68] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[69] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artifcial Intelligence Research*, (4):237–285, 1996.

[70] S. Kambhampati, C. Knoblock, and Q. Yang. Planning as refinement search: A unified framework for evaluating design tradeoffs in partial order planning. *Artificial Intelligence Special Issue on Planning and Scheduling*, 1995.

[71] J. Karlholm. Design and evaluation of a hierarchy of boosted classifiers for detection of ground targets in aerial surveillance imagery. In *Automatic Target Recognition XIV, Proc. SPIE*, volume 5426, April 2004. To be published.

[72] R. Karlsson, J. Jansson, and F. Gustafsson. Model-based statistical tracking and decision making for collision avoidance application. Technical Report LiTH-ISY-R-2599, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, Mar 2004.

[73] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1):166–171, 1998.

[74] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[75] J.-O. Kim and P.K. Khosla. Real-tim obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, pages 338–349, June 1992.

[76] L. Källgren. Real-time replanning of mission routes based upon threats. Master's thesis, Department of Electrical Engineering, Linköping University, 2001.

[77] B.O. Koopman. *Search and Screening: General Principles with Historical Applications*. Pergamon Press, 1980.

[78] D. Kragic and H.I. Christensen. A framework for visual servoing. In M. Vincze and J. Crowley, editors, *ICVS-03*, volume 2626 of *LNCS*. Springer Verlag, March 2003.

[79] M. Kruusmaa, J. Willemson, and K. Heero. Path selection for mobile robots in dynamic environments. In *1st European Conference on Mobile Robots*, Sep 2003.

[80] V. Kumar, M. Zefran, and J. Ostrowski. Motion planning and control of robots. *Handbook of Industrial Robotics*, Oct 1997.

[81] T. Lane and L. P. Kaelbling. Toward hierarchical decomposition for planning in uncertain environments. In *IJCAI Workshop on Planning under Uncertainty and Incomplete Information*, pages 1–7, 2001.

[82] U. Larsson, C. Zell, K. Hyyppä, and Å. Wernersson. Navigating an articulated vehicle and reversing with a trailer. In *IEEE International Conference on Robotics and Automation*, pages 2398–2404, 1994.

[83] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[84] J. P. Laumond, editor. *Robot motion planning and control, Lectures Notes in Control and Information Sciences 229.* Springer, 1998.

[85] S. LaValle. *A Game-Theoretic Framework for Robot Motion Planning.* PhD thesis, University of Illinois at Urbana-Champaign, 1995.

[86] S. M. LaValle. *Planning Algorithms.* [Online], 1999-2004. Available at http://msl.cs.uiuc.edu/planning/.

[87] J. Lee, R. Huang, A. Vaughn, X. Xiao, J.K. Hedrick, M. Zennaro, and R. Sengupta. Strategies of path-planning for a UAV to track a ground vehicle. In *AINS*, June 2003.

[88] A. Lennartsson and D. Skoogh. Sensor redundancy for inertial navigation. Technical report FOI-R–1085–SE, FOI, 2003.

[89] S.-M. Li, J.D. Boskovic, S. Seereeram, R. Prasanth, J. Amin, R.K. Mehra, and R. Beard. Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs). In *American Control Conference*, pages 274–279, May 2002.

[90] M. Lindhé, P. Ögren, and K-H. Johansson. Flocking with obstacle avoidance using methods from coverage control. Presented at Reglermöte 2004 at Chalmers University of Technology, Sweden, May 2004.

[91] F. Lingelbach. Path planning using probabilistic cell decomposition. In *IEEE International Conference on Robotics and Automation*, New Orelans, April 2004.

[92] M. L. Littman and S. M. Majercik. Large-scale planning under uncertainty: A survey. In *NASA Workshop on Planning and Scheduling in Space*, pages 27:1–8, 1997.

[93] S. Liu and L.E. Holloway. Active sensing policies for stochastic systems. *IEEE Transactions on Automatic Control*, 47(2):373–377, 2002.

[94] V.J. Lumelsky. A comparative study on the path length performance of maze-searching and robot motion planning algorithms. *IEEE Transactions on Robotics and Automation*, 7(1):57–66, 1991.

[95] V.J. Lumelsky and A.M. Shkel. Incorporating body dynamics into the sensor-based motion planning paradigm. The maximum turn strategy. In *IEEE International Conference on Robotics and Automation*, pages 1637–1643, 1995.

[96] A.A. Makarenko, S.B. Williams, F. Bourgault, and H.F. Durrant-Whyte. An experiment in integrated exploration. In *International Conference on Intelligent Robots and Systems*, 2002.

[97] J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach.* Prentice Hall, 1994.

[98] E. Marchand and G.D. Hager. Dynamic sensor planning in visual servoing. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1988–1993, May 1998.

[99] P. S. Maybeck. *Stochastic Models, Estimation and Control*, volume 3. Academic Press, 1979.

[100] M. Mazo, A. Speranzon, K.H. Johansson, and X. Hu. Multi-robot tracking of a moving object using directional sensors. In *IEEE International Conference on Robotics and Automation*, 2004.

[101] J. E. McInroy and G. N. Saridis. Reliable control and sensor fusion in intelligent machines. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 487–492, 1991.

[102] G. McIntyre and K. Hintz. A comprehensive approach to sensor management, Part I: A survey of modern sensor management systems. In *IEEE Transactions on Systems, Man and Cybernetics*, 1999.

[103] T. McLain and R. Beard. Cooperative rendezvous of multiple unmanned air vehicles. In *AIAA Guidance, Navigation and Control Conference*, volume AIAA-2000-4369:AAO-37126, August 2000.

[104] T. W. McLain. Coordinated control of unmanned air vehicles. Afrl report, 1999.

[105] L. Mihaylova, J. De Schutter, and H. Bruyninckx. A multisine approach for trajectory optimization based on information gain. In *IEEE/RSJ International Conference on Intelligent Robots and System*, Sept-Oct 2002.

[106] L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, and J. De Schutter. Active sensing for robotics - a survey. In I. Dimov, I. Lirkov, S. Margenov, and Z. Zahari, editors, *Proceedings of the 5th International Conference on Numerical Methods and Applications*, pages 316–324, Borovets, Bulgaria, August 2002. Springer.

[107] M.B. Milam, K. Mushambi, and R.M. Murray. A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Conference on Decision and Control*, 2000.

[108] J. Miura and Y. Shirai. Vision-motion planning with uncertainty. In *IEEE International Conference on Robotics and Automation*, pages 1772–1777, Nice, 1992.

[109] J. Miura and Y. Shirai. Parallelizing planning and action of a mobile robot based on planning-action consistency. In *IEEE International Conference on Robotics and Automation*, pages 1750–1756, Seoul, 2001.

[110] S. J. Moorehead. Autonomous surface exploration for mobile robots. Technical Report CMU-RI-TR-01-30, The Robotics Institute, Carnegie Mellon University, Aug 2001.

[111] R. Munos and A. Moore. Variable resolution discretization in optimal control. *Machine Learning*, 49(2/3):291–323, November/December 2002.

[112] R.M. Murray, M. Rathinam, and W. Sluis. Differential flatness of mechanical control systems: A catalog of prototype systems. In *ASME Int'l Mechanical Engineering Congress and Exposition*, 1995.

[113] J. C. Musto and G. N. Saridis. Entropy-based reliability analysis for intelligent machines. *IEEE Transactions on Systems, Man and Cybernetics*, 27(2):239–244, April 1997.

[114] R. B. Myerson. *Game theory: analysis of conflict*. Harvard University Press, 1991.

[115] G.W. Ng and K.H. Ng. Sensor management - what, why and how. *Information Fusion*, 1(2):67–75, Dec 2000.

[116] I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis, and A.N. Kostaras. Evolutionary algorithm based offline/online path planner for UAV navigation. *Systems, Man and Cybernetics, Part B*, 33(6):898– 912, Dec 2003.

[117] B. Nilsson. *On Robot Control using Range Sensors with Special Reference to Active Uncertainty Reduction*. PhD thesis, Linköping University, 1997. Dissertation No. 511.

[118] B. Nilsson, J. Nygårds, and Å. Wernersson. On-range sensor feedback for mobile robot docking within prescribed posture tolerances. *Journal of Robotic Systems*, 14(4):197–312, 1997.

[119] M. Norsell. *Aircraft Trajectory Optimization with Tactical Constraints*. PhD thesis, The Royal Institute of Technology, June 2004.

[120] J. Nygårds, P. Skoglar, M. Ulvklo, and T. Högström. Navigation aided image processing in UAV surveillance: Preliminary results and design of an airborne experimental system. *Journal of Robotic Systems*, 21(2), 2004.

[121] J. Nygårds. *On Robot Feedback from Range Sensors: Reliable Control by Active Reduction of Uncertainty and Ambiguities*. PhD thesis, Linköping University, 1998. Thesis No. 528.

[122] M. Nyström and M. Norrlöf. Path generation for industrial robots. Technical Report LiTH-ISY-R-2529, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, Aug 2003.

[123] P. Ögren. *Formations and Obstacle Avoidance in Mobile Robot Control*. PhD thesis, Royal Institute of Technology, KTH, Stockholm, June 2003.

[124] P. Ögren and N.E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 21(2):188–195, April 2005.

[125] P. J. Antsaklis (chair). Defining intelligent control. *IEEE Control Systems Magazine*, 14(3):4–5,58–66, June 1993.

[126] M. Pachter and P.R. Chandler. Challenges of autonomous control. In *IEEE Control Systems Magazine*, volume 18, pages 92–97, Aug 1998.

[127] J.M. Passerieux and D. van Cappel. Optimal observer maneuver for bearing-only tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):777–788, July 1998.

[128] K.M. Passino. *Perspectives in Control Engineering: New Concepts and Applications*, chapter Intelligent Control: An Overview of Techniques, pages 104–133. IEEE Press, 2001.

[129] M. Persson. A simulation environment for visual servoing, 2002. Lic. thesis, Örebro University.

[130] P. O. Pettersson. Helicopter path planning using probabilistic roadmaps. Master's thesis, Artificial Intelligence & Integrated Computer Systems Division, Linköping University, Sweden, 2003.

[131] P. O. Pettersson and P. Doherty. Probabilistic roadmap based planning for an autonomous unmanned aerial vehicle. In *ICAPS-04 Workshop on Connecting Planning Theory with Practice*, 2004.

[132] R. Pito. A sensor-based solution to the "Next Best View" problem. *Proc of 13th Int. Conf. on Pattern Recognition*, 1:941–945, August 1996.

[133] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *IEEE International Conference on Robotics and Automation*, pages 802–807, May 1993.

[134] M. Reed and P. Allen. Constraint-based sensor planning for scene modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Dec 2000.

[135] A. Richards, J. Bellingham, M. Tillerson, and J.P. How. Co-ordination and control of multiple UAVs. In *AIAA Guidance, Navigation, and Control Conference*, August 2002.

[136] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference*, volume ACC02-AIAA-1057, May 2002.

[137] O. Ringdahl. Path tracking and obstacle avoidance algorithms. Master's thesis, Umeå University, 2003.

[138] S. Rönnbäck. On a robotic matlab java testbed for mobile robots. Licentiate thesis, Luleå University Of Technology, 2004.

[139] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation: Robot navigation under uncertainty in dynamic environments. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 35–40, May 1999.

[140] A. Saffiotti. Handling uncertainty in control of autonomous robots. In M. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today*, number 1600 in LNAI, pages 381–408. Springer-Verlag, Berlin, DE, 1999. Online at http://www.aass.oru.se/~asaffio/.

[141] G.N. Saridis. On the theory of intelligent machines: a survey. In *Proceedings of the 27th IEEE Conference on Decision and Control*, volume 3, pages 1799–1804, 1988.

[142] L. Sciavicco and B. Siciliano. *Modeling and control of robot manipulators.* The McGraw-Hill Companies, Inc, 1996.

[143] M. Seiz, P. Jensfelt, and H. I. Christensen. Active exploration for feature based global localization. In *Proc. of the International Conference on Intelligent Robots and Systems*, 2000.

[144] T. Simeon, J.-P. Laumond, and C. Nissoux. Visibility based probabilistic roadmaps for motion planning. *Advanced Robotics Journal*, 14(6), 2000.

[145] Z. Sjanic. Online mission planning based on model predective contrl. Master's thesis, Department of Electrical Engineering, Linköping University, 2001.

[146] A. Speranzon and K.H. Johansson. On localization and communication issues in pursuit-evasion games. In *Proceeding of the Workshop on Cooperative Robotics, IROS02*, 2002.

[147] I. Stamos and P. Allen. Interactive sensor planning. In *Computer Vision and Pattern Recognition Conference (CVPR)*, pages 489–494, June 1998.

[148] T. J. Stewart. Search for a moving target when search motion is restricted. *Computers and Operations Research*, 6:129–140, 1979.

[149] M. Strandberg. Augmenting RRT-planners with local trees. In *IEEE Int. Conf. on Robotics and Automation*, April 2004.

[150] D. Strömberg and F. Lantz. Operator control of shared resources in sensor networks. In *The 6:th International Conference on Information Fusion*, pages 575–582, 2003.

[151] K.A. Tarabanis, P.K. Allen, R.Y. Tsai, and T.J. Watson. A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1):86–104, Feb 1995.

[152] E. Tse and Y. Bar-Shalom. Adaptive dual control for stochastic nonlinear systems with free end-time. *IEEE Transactions on Automatic Control*, 18(2):670–675, 1975.

[153] M. Ulvklo, J. Nygårds, J. Karlholm, and P. Skoglar. Image processing and sensor management for autonomous UAV surveillance. In *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XXVII, Proc. SPIE*, volume 5409, April 2004. To be published.

[154] M. Ulvklo, J. Nygårds, P. Skoglar, and J. Karlholm. Image analysis and sensor management in UAV surveillance. In *Proceedings of the SSBA 04 Symposium on Image Analysis*, pages 134–137, Uppsala, Sweden, March 2004. SSBA.

[155] M.J. van Nieuwstadt and R.M. Murray. Real time trajectory generation for differentially flat systems. In *Int'l. J. Robust & Nonlinear Control*, volume 8, pages 995–1020, 1998.

[156] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18(5):662–669, 2002.

[157] A. Wallén. *Tools for Autonomous Process Control.* PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, February 2000.

[158] P.K.C. Wang. Optimal path planning based on visibility. *Optimization Theory and Applications*, 117(1):157–181, April 2003.

[159] A. R. Washburn. Search for a moving target: The fab algorithm. *Operations Research*, 29:1227–1230, 1983.

[160] A. R. Washburn. Branch and bound methods for a search problem. *Naval Research Logistics*, 45(3):243–257, 1997.

[161] S. Waydo and R.M. Murray. Vehicle motion planning using stream functions. In *IEEE International Conference on Robotics and Automation*, Sep 2003.

[162] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, 1997.

[163] M. Winstrand. Mission planning and control of multiple UAVs. Scientific report FOI-R–1382–SE, FOI, 2004.

[164] J. Wintenby. *Resource Allocation in Airborne Surveillance Radar.* PhD thesis, Chalmers University of Technology, 2003.

[165] B. Wittenmark. Adaptive dual control. In H. Unbehauen, editor, *Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO*. Eolss Publishers, Oxford, UK, 2002. Paper 6.43.15.6.

[166] N. Xiong and P. Svensson. Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163–186, Dec 2000.