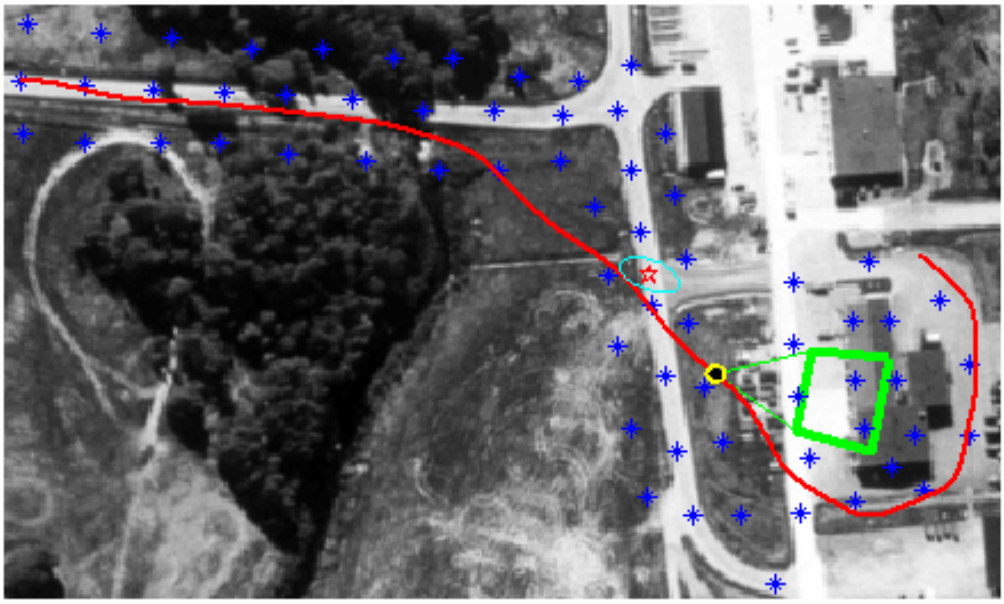
FOI-R--1685--SE
May 2005
ISSN 1650-1942

Scientific report

Per Skoglar, Rickard Björström, Jonas Nygårds, Morgan Ulvklo

Information-Theoretic Approach for Concurrent Path and Sensor Planning for a UAV with EO/IR sensors



SWEDISH DEFENCE RESEARCH AGENCY

FOI

P.O. Box 1165

SE-581 11 Linköping

Sweden

FOI-R--1685--SE

May 2005

ISSN 1650-1942

Scientific report

Per Skoglar, Rickard Björström, Jonas Nygårds, Morgan Ulvklo

Information-Theoretic Approach for Concurrent Path and Sensor Planning for a UAV with EO/IR sensors

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Sensor Technology P.O. Box 1165 SE-581 11 Linköping	FOI-R--1685--SE	Scientific report
	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)	Project manager
Per Skoglar Rickard Björström Jonas Nygårds Morgan Ulvklo	Morgan Ulvklo
	Approved by Lena Klasén
	Sponsoring agency FMV / Swedish Armed Forces
	Scientifically and technically responsible Jonas Nygårds

Report title

Information-Theoretic Approach for Concurrent Path and Sensor Planning for a UAV with EO/IR sensors

Abstract (not more than 200 words)

This report presents an information-theoretic approach to concurrent path and sensor planning for a UAV equipped with gimballed EO/IR sensors. The goal is autonomuos UAV surveillance, i.e., search for objects along a road or in a certain area and localization of discovered targets. The work is inspired by research in optimal observer trajectory computations for bearings-only tracking.

The path planning problem is first formulated and solved as an optimal control problem. A second method uses an interpolation method, e.g. splines, to solve the combined path and sensor planning problem. In the first method the optimization parameters are control signals for the UAV and in the second method the optimization parameters are waypoints of the UAV flight path and the sensor gazing directions. For both methods, an information matrix is constructed from the states representing targets and area points. A utility function is defined based on the information matrix and the planning process is to determine optimization parameters that maximize a utility function at the planning horizon.

Different parameters and aspects of the methods are discussed, e.g. different utility functions, length of time horizon and prior information. The methods are applied to single target localization, *n* target localization, and area exploration. The spline method is also extended to handle a limited field-of-view sensor, as well as scene occlusion. Finally, simulation results of a road surveillance scenario are presented.

Keywords

path planning, sensor planning, information theory, optimal control, localization, area coverage, exploration, UAV surveillance

Further bibliographic information	Language English
ISSN 1650-1942	**Pages** 72 p.
	Price acc. to pricelist

Utgivare Totalförsvarets Forskningsinstitut - FOI Sensorteknik Box 1165 581 11 Linköping	**Rapportnummer, ISRN** FOI-R--1685--SE	**Klassificering** Vetenskaplig rapport
	Forskningsområde 7. Farkost- och rymdteknik, inkl material	
	Månad, år Maj 2005	**Projektnummer** E3969
	Delområde 71 Obemannade farkoster	
	Delområde 2	

Författare/redaktör Per Skoglar Rickard Björström Jonas Nygårds Morgan Ulvklo	**Projektledare** Morgan Ulvklo
	Godkänd av Lena Klasén
	Uppdragsgivare/kundbeteckning FMV / Försvarsmakten
	Tekniskt och/eller vetenskapligt ansvarig Jonas Nygårds

Rapportens titel (i översättning)

Informationsteoretisk rutt- och sensorplanering för UAV med EO/IR sensorer

Sammanfattning (högst 200 ord)

Denna rapport presenterar en informationsteoretisk ansats för samtidig sensor- och ruttplanering för en UAV med EO/IR-sensorer. Det övergripande målet är autonom UAV-spaning, d.v.s. spaning efter mål längs en väg eller i ett särskilt område och positionsinmätning av upptäckta mål.

Ruttplaneringsproblemet är först formulerat och löst som ett optimalt styrningsproblem. En andra metod baserad på interpolation, med t.ex. splinefunktioner, används sedan för att lösa det kombinerade rutt- och sensorplaneringsproblemet. I den första metoden är optimeringsparametrarna styrsignaler för UAV:n och i den andra metoden är optimeringsparametrarna punkter längs UAV:ns flygbana samt sensorns stirr-riktning. För båda metoderna finns en informationsmatris baserad på tillstånd som representerar eventuella objektet och ytpunkter. Optimeringsproblemet är att bestämma värden på optimeringsparametrar som maximerar en målfunktion, baserad på informationsmatrisen, givet en viss tidshorisont.

Olika parameterval och aspekter hos metoderna diskuteras, t.ex. påverkan av olika målfunktioner, tidshorisonter och initial information. Metoderna är testade på lokalisering av ett mål, lokalisering av *n* mål, samt yttäckning. Spline-metoden är även utökad med en sensormodell med begränsat synfält och även ocklusionsproblemet berörs. Rapporten avslutas med simuleringsresultat från ett spaningsscenario där en väg och en byggnad ska övervakas och eventuella mål mätas in med en UAV.

Nyckelord

ruttplanering, sensorplanering, informationsteori, optimal styrning, lokalisering, yttäckning, UAV-spaning

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 72
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

Chapter 1

Introduction

This report is concerned with a information-theoretic approach to path and sensor planning for a UAV with gimballed EO/IR sensors.

1.1 Background

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 surveillance 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.

1.2 Objective

A survey of control and planning approaches from different scientific areas applicable to autonomous UAV surveillance is given in [6]. This report enters more deeply into one interesting approach presented in [3], namely, the information-theoretic approach. Information-theoretic approaches have been used in path planning problems before, however, we wanted to examine if this approach also is suitable for *concurrent sensor and*

path planning. *Path planning* means the planning of the sensor platform (vehicle) trajectory and *sensor planning* means, at first hand, the planning of the gazing direction of a EO/IR sensor. In future work, also other parameters will be considered, e.g. zoom.

1.3 Information vs. Detection and Identification

Throughout this report, the task of the UAV is to detect, localize and identify objects on the ground. To represent the ability to identify targets, the concept of information is introduced in the sense "the more information about an object, the more likely is the identification". When an object is said to be identified, there are sufficient number of pictures of the object that it could be identified with methods from image analysis. There are no real images in this work, instead the amount of information is connected to the uncertainty of the target's geometrical location. The terms "identify" and "detect" an object would now be that the information value of the object is larger than a pre-defined threshold values.

1.4 References and Further Readings

There are much research in UAV path planning and cooperation. On the other hand, there are very few results presented on the concurrent path and sensor planning problem similar to our. Nygårds [6] gives an overview of research fields and communities related to path planning and/or sensor planning. The report also presents methods, techniques and approaches to path and sensor planning. Ulvklo [11] gives an introduction to the UAV surveillance and reconnaissance problem. In particular, the path and sensor planning challenges in UAV surveillance are discussed.

A large amount of work has been performed in the last six decades on the problem of optimal observer trajectory for bearings-only tracking. Applications can be found in passive sonar and EO/IR tracking and in aircraft navigation systems. Typically, the problem is to estimate the position, and possible the velocity, of a target based on a sequence of noisy measurements from a bearings-only sensor mounted on a moving sensor platform. See e.g. [8, 10, 7] for rather recent papers on this subject. The path planning problem is formulated and solved as an optimal control problem by Grocholsky in [3]. The 2D single object localization example in Chapter 3 are from [3]. The area coverage idea in Chapter 4 is also from [3]. The gimballed sensor system that serves as pattern in Chapter 6.1 is presented by Skoglar in [9].

A detailed presentation of information theory, information filter and utility functions can be found in Manyika and Durrant-Whyte [5]. The solution of the optimal control problem by parametrization of the control signal is described by Bertsekas in [1].

1.5 Outline

Chapter 2 introduces the information theory and the optimal control problem. In Chapter 3 the problem of optimal observer trajectory for bearings-only localization is introduced. The problem is formulated as an optimal control problem. The effects of different utility function and time horizon are discussed. Furthermore, the model are extended to n objects and a 3D representation. Chapter 4 considers the area coverage problem and an area model is introduced. Chapter 5 introduces another method called spline

optimization. The optimization parameters are waypoints of the flight path instead control signals. Chapter 6 considers the problems with limited field-of-view and occlusion. Some basic models are presented and the spline optimization method is extended with a sensor gazing representation. Chapter 7 presents a road and building surveillance example and simulation results are given. Chapter 8 summarizes the results and suggestions for future work are given.

1.6 Acknowledgments

Thanks to Patrik Hermansson, Jörgen Karlholm, and Johan Hamberg for valuable comments and discussions.

Chapter 2

Information Theory and Optimal Control

In this chapter, the optimal control problem based on the information criterion is formulated. First the concept of information is introduced and how it is used in optimal control, and this results in a utility function from information theory, which should be optimized in the optimal control problem.

2.1 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*.

2.1.1 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 [4]

$$h(\mathbf{X}) \equiv -E\{\log p(\mathbf{x})\} = -\int_{\mathcal{R}^N} p(\mathbf{x}) \log p(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{2.1}$$

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

$$i(\mathbf{X}) \equiv -h(\mathbf{X}), \tag{2.2}$$

i.e., information is maximized when entropy is minimized. When the probability distribution of an n-dimensional state $\mathbf{X}$ is Gaussian, with mean $\mathbf{m}$ and covariance matrix P, the entropic information becomes [3]

$$i(\mathbf{X}) = -h(\mathbf{X}) = -\frac{1}{2} \log\Big((2\pi e)^n \ \mathrm{det} P\Big). \tag{2.3}$$

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

$$p(\mathbf{x}|\mathbf{Z}^k) = \frac{p(\mathbf{z}(k)|\mathbf{x}) p(\mathbf{x}|\mathbf{Z}^{k-1})}{p(\mathbf{z}(k)|\mathbf{Z}^{k-1})} \tag{2.4}$$

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

$$i(\mathbf{X}|\mathbf{Z}^k) = i(\mathbf{X}|\mathbf{Z}^{k-1}) + E\Big\{ \log \frac{p(\mathbf{z}(k)|\mathbf{x})}{p(\mathbf{z}(k)|\mathbf{Z}^{k-1})} \Big\} \tag{2.5}$$

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.

2.1.2 Fisher Information

The (expected) *Fisher information* of a n-dimensional family $p(\mathbf{y}, \theta)$ of distributions at the parameter value $\theta \in \mathcal{R}^n$, is defined as the matrix $\mathcal{I}$ with components

$$\mathcal{I}_{ij}(\theta) \equiv E_\theta\big\{ \partial_{\theta_i} \log p \;\; \partial_{\theta_j} \log p \big\} \tag{2.6}$$

where E_θ denotes expectation w.r.t $p(\mathbf{y}, \theta)$. This is well-defined for discrete as well as continuous distributions. For a single continuous distribution $p(\mathbf{y})$ on $\mathcal{R}^n$, this may be applied to the translational family $p_1(\mathbf{y}, \theta) = p(\mathbf{y} - \theta)$ leading to constant (i.e. θ-independent) values

$$\mathcal{I}_{ij} = \int \partial_{\theta_i} \log p_1 \;\; \partial_{\theta_j} \log p_1 \;\; p_1 \; dy = \int \partial_{y_i} \log p \;\; \partial_{y_j} \log p \;\; p \; dy \tag{2.7}$$

This is the case treated in the present report.

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 $\mathbf{x}$

$$\mathcal{I}(k) = -E\big\{ \nabla_x \nabla_x^T \log p(\mathbf{Z}^k, \mathbf{x}) \big\} = -E\big\{ \nabla_x \nabla_x^T \log p(\mathbf{x}|\mathbf{Z}^k) \big\} \tag{2.8}$$

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

$$\mathcal{I}(k) = -E\big\{ \nabla_x \nabla_x^T \log p(\mathbf{Z}^k|\mathbf{x}) \big\} \tag{2.9}$$

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

$$i(k) = -\frac{1}{2} \log\Big((2\pi e)^n \; \det P(k|k) \Big) = \frac{1}{2} \log\Big((2\pi e)^{-n} \; \det \mathcal{I}(k) \Big). \tag{2.10}$$

In this report we assume Gaussian distributions and the *information matrix* is defined as

$$Y = Y(t) = \mathcal{I}(\mathbf{x}(\mathbf{t})). \tag{2.11}$$

2.2 The Information Filter

The information matrix $Y(t)$ is updated by observations of objects. Let the states to be observed be the vector $\mathbf{x}(t)$ and the observations are in the observation vector $\mathbf{z}(t)$. The observation $\mathbf{z}(t)$ is a function of the observed states $\mathbf{x}(t)$ and some observation noise $\mathbf{v}(t)$ as [3]:

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{v}(t)). \tag{2.12}$$

The sensor making the observation is a bearings-only sensor, which can only make observations about the angle to the objects. The function $\mathbf{h}$ is a nonlinear function since it is the observed angle, $\tilde{\varphi}(t)$, which is the real angle, $\varphi(t)$, plus the noise:

$$\mathbf{h}(\mathbf{x}(t), \mathbf{v}(t)) = \tilde{\varphi}(\mathbf{x}(t), \mathbf{v}(t)) = \varphi(\mathbf{x}(t)) + \mathbf{v}(t). \tag{2.13}$$

The observation noise $\mathbf{v}(t)$ is modelled as white noise with covariance R. The nonlinear function is linearized about nominal states $\mathbf{x}_n(t)$ and $\mathbf{z}_n(t)$ as in Grocholsky [3]:

$$\delta\mathbf{z}(t) = H(t)\delta\mathbf{x}(t) + D(t)\mathbf{v}(t). \tag{2.14}$$

Where

$$H(t) = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t)\\ \mathbf{v}(t)=0}} \text{ and } D(t) = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{v}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t)\\ \mathbf{v}(t)=0}} \tag{2.15}$$

and the new linearized states are

$$\begin{aligned}\delta\mathbf{x}(t) &\equiv \mathbf{x}(t) - \mathbf{x}_n(t)\\ \delta\mathbf{z}(t) &\equiv \mathbf{z}(t) - \mathbf{z}_n(t).\end{aligned}$$

The H-matrix is called the linearized observation matrix and the D-matrix is the linearized observation noise matrix. The reason for the linearization, is that the information filter equations are linear. The linearized filter equations are detailed in Manyika and Durrant-Whyte [5] and simplified in Grocholsky [3], and describes how the *expected information* $I(t)$ is related to the observations, which is

$$I(t) = H(t)^T R^{-1} H(t). \tag{2.16}$$

The information matrix is updated not only by the observed information, but there are losses due to process noise and the system dynamics as well. First consider the system equations

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \tag{2.17}$$

where $\mathbf{x}(t)$ is the states, $\mathbf{u}(t)$ is known control inputs and $\mathbf{w}(t)$ is the process noise, the latter is assumed to be a zero mean uncorrelated Gaussian process with covariance Q. The function $\mathbf{f}$ is a system of non-linear differential equations, that could be linearized about nominal states $\mathbf{x}_n(t)$ and nominal control signals $\mathbf{u}_n(t)$ as

$$\delta\dot{\mathbf{x}}(t) = F(t)\delta\mathbf{x}(t) + B(t)\delta\mathbf{u}(t) + G(t)\mathbf{w}(t). \tag{2.18}$$

The matrix $F(t)$ is the linearized state transition matrix, $B(t)$ is the linearized input matrix and $G(t)$ is the linearized noise matrix. These are given by

$$F(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t)\\ \mathbf{u}(t)=\mathbf{u}_n(t)\\ \mathbf{w}(t)=0}}, \quad B(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t)\\ \mathbf{u}(t)=\mathbf{u}_n(t)\\ \mathbf{w}(t)=0}} \text{ and } G(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t)\\ \mathbf{u}(t)=\mathbf{u}_n(t)\\ \mathbf{w}(t)=0}} \tag{2.19}$$

where

$$\begin{aligned} \delta\mathbf{x}(t) &\equiv \mathbf{x}(t) - \mathbf{x}_n(t) \\ \delta\mathbf{u}(t) &\equiv \mathbf{u}(t) - \mathbf{u}_n(t). \end{aligned} \tag{2.20}$$

Given these matrices, the update law of the information matrix $Y(t)$ is given in [3] as:

$$\dot{Y}(t) = -Y(t)F(t) - F^T(t)Y(t) - Y(t)G(t)Q(t)G^T(t)Y(t) + I(t) \tag{2.21}$$

where $I(t)$ is given in (2.16). This is the continuous version of the prediction and update states of the information filter which is described in [3] and [5]. Since R and Q are positive semi-definite, the process noise cannot gain any information and observation cannot lose information. However, the system dynamics in F could lose or gain information over time.

2.3 Information Matrix and Utility Function

Let

$$Y = Y(t) = \mathcal{I}(\mathbf{x}(t)) \tag{2.22}$$

be the *information matrix* throughout this report. In order to maximize the "information" a *utility function*, giving a scalar value suitable for comparison, is needed. For a discussion of the properties of utility functions, see e.g. [4]. The utility function must combine or weigh the elements or eigenvalues of the information matrix or its inverse. The determinant of the information matrix is an example of a utility function and the problem can be expressed as

$$\max(\det Y) \tag{2.23}$$

Two alternatives are maximizing the trace and minimizing the trace of the inverse, i.e.,

$$\max(\mathrm{tr} Y) \tag{2.24}$$

and

$$\min(\mathrm{tr} Y^{-1}) \tag{2.25}$$

respectively.

Let $\{\lambda_1, \ldots, \lambda_n\}$ be the eigenvalues of the information matrix Y. The utility functions in (2.23), (2.24) and (2.25) can then be expressed in terms of the eigenvalues:

$$\det Y = \prod_{i=1}^{n} \lambda_i, \tag{2.26}$$

$$\mathrm{tr} Y = \sum_{i=1}^{n} \lambda_i, \tag{2.27}$$

$$\mathrm{tr} Y^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i}, \tag{2.28}$$

respectively. The information matrix has non-negative eigenvalues, since it is defined as the expectation of a non-negative matrix.

In the Gaussian distribution case, the utility function based on the determinant (2.23) is equivalent with maximizing the entropic information, see (2.10). Entropy is a appropriate measure due to its general applicability and resulting preference profiles [4]. However, as can be seen from (2.26), and (2.27), there are drawbacks. A poorly scaled information matrix could give high information even when some eigenvalues are small in comparison. In the object localization case, this corresponds to a position uncertainty that is very small in some directions, but very large in other. Thus, despite large information, the object can not be considered as satisfactory localized. This motivates the use of alternative utility functions, e.g. (2.28). An other utility function that is working on the smallest eigenvalue is

$$\max(\min(\text{eig}Y)). \tag{2.29}$$

However, this alternative works bad when the problem grows and several objects are included in the model.

2.4 Optimal Control

2.4.1 The General Optimal Control Problem

The general optimal control problem could be expressed simply as "choose the control signal such that the system behaves as good as possible" [2]. In mathematical terms, the problem can be formulated as in Glad and Ljung [2]:

Given initial conditions:

$$\mathbf{x}(0) = \mathbf{x}(t_0) \tag{2.30}$$

System equations:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.31}$$

Subject to constraints:

$$\psi(\mathbf{x}(t), \mathbf{u}(t)) = 0 \tag{2.32}$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leq 0 \tag{2.33}$$

The criterion function to minimize is:

$$J(\mathbf{x}(t), \mathbf{u}(t)) = \phi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t))\ dt \tag{2.34}$$

The solution to the optimal control problem would be:

$$\min_{\mathbf{u}(t)} J(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.35}$$

subject to the constraints (2.32) and (2.33).

2.4.2 Optimal Control In Path Planning

The criterion function J in (2.34) is the utility function discussed in Section 2.3, if the utility function is modified such that it is minimized. A maximum problem, could always be converted into a minimum problem,

$$\max g(x) = -\min(-g(x)), \tag{2.36}$$

and in the implementation, the utility functions that required a maximization, like maximizing the determinant, were converted into minimum problems.

The solution to the optimal control problem is affected by the choice of utility function and the idea is to find a control signal, or sequence of control signals, that maximizes information. However, the optimization is done over a pre-defined optimization time, called the *time horizon* denoted by t_f, and it is only of interest to consider the information at the time horizon t_f. Therefore the criterion function is a function of the time horizon only and L in (2.34) is equal to zero, i.e.,

$$J = J(t_f) = \phi(\mathbf{x}(t_f)). \tag{2.37}$$

Most optimal control problems requires a numerical solution and by parametrizing the control signal into m steps in every optimization, an approximate solution will be found, for details see Bertsekas [1]. The idea is now to find a sequence of control steps u_i that maximizes information at the time horizon t_f as in Grocholsky [3]:

$$\mathbf{u}_i(t) = \mathbf{p}_i \chi_i(t), \quad i = 1, \ldots, m. \tag{2.38}$$

Where χ_i simply holds the control variable over m equal time steps Δt_u as

$$\chi_i(t) = \begin{cases} 1 & \text{if } (j-1)\Delta t_u \leq t \leq j\Delta t_u \\ 0 & \text{otherwise} \end{cases}, \quad \Delta t_u = \frac{t_f - t_0}{m}. \tag{2.39}$$

The optimal control problem in (2.35) is now converted into a nonlinear programming problem [3]:

$$\min_{\mathbf{p}} J(\mathbf{p}) = \phi(\mathbf{x}(t_f)) + \frac{1}{2}\Delta t_x \sum_{i=1}^{m \cdot n_{steps}} (L_i(\mathbf{x}_i, \mathbf{u}_k) + L_{i-1}(\mathbf{x}_{i-1}, \mathbf{u}_k)), \tag{2.40}$$

subject to the constraints in (2.32) and (2.33), where $\mathbf{p} = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$ is the parameter vector and $\Delta t_x = \frac{\Delta t_u}{n_{steps}}$, $\{n_{steps} \geq 1\}$ is the time between the evaluations of the states, and k is the control index. The original optimal control problem is now in the form of mathematical programming problem [3]. In the path planning problem, L is zero, but it is given for generality.

There are no equality constraints as in equation (2.32), but there could be inequality constraints as in equation (2.33), with the control signals bounded as

$$u_{min} \leq u_i \leq u_{max}, \tag{2.41}$$

since it is reasonable that the movement of a UAV is restricted by its dynamics.

The optimal control problem is solved by *Matlab*'s Optimization Toolbox. If there are no bounds on the control signal, the problem is considered to be a unconstrained problem, solved by the function *fminunc*. When the problem are constrained, the problem is solved by the function *fmincon*. In this report constrained problems are also solved as a unconstrained optimization problem with the bounds included as penalties in the utility function.

2.4.3 Gradient Determination

The functions in *Matlab*'s toolbox uses the gradient $\nabla_p J$ and the Hessian $\nabla^2_{\mathbf{p}} J$ in order to find the local minimum. The use of the gradient will help the minimizer to reach a minimum in terms of efficiency and reliability. This can be done in two ways, either by letting the optimizer calculate the gradient and Hessian itself as a numerical solution, or by giving the analytical expressions explicitly. The first method requires no knowledge of the partial derivatives with respect to state and control vectors. However, the drawback is the computational load. For the latter method, the details are given here for the gradient, the details for calculating the Hessian are given in Grocholsky [3].

Differentiating (2.40) ($L = 0$) with respect to $\mathbf{p}$ gives:

$$(\nabla_p J) = \frac{\partial \phi(\mathbf{x}(t_f))}{\partial \mathbf{x}(t_f)} \frac{\partial \mathbf{x}(t_f)}{\partial \mathbf{p}} \tag{2.42}$$

The first part of (2.42) is simply the derivative of the criterion function with respect to each state at the time horizon t_f. The second part is derived from (2.31) by applying the chain rule, which gives:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{p}}, \text{ with } \left.\frac{\partial \mathbf{x}}{\partial \mathbf{p}}\right|_{t=t_0} = \frac{\partial \mathbf{x}_0}{\partial \mathbf{p}} \tag{2.43}$$

By the use of a Heun scheme, the first order sensitivities could be calculated as:

$$\frac{\partial \mathbf{x}_i}{\partial \mathbf{p}} = \left[\mathrm{I}_n - \frac{1}{2}\Delta t_x \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}\right]^{-1} \left[\left[\mathrm{I}_n + \frac{1}{2}\Delta t_x \frac{\partial \mathbf{f}_{i-1}}{\partial \mathbf{x}_{i-1}}\right]\frac{\partial \mathbf{x}_{i-1}}{\partial \mathbf{p}} + \frac{1}{2}\Delta t_x \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_k} + \frac{\partial \mathbf{f}_{i-1}}{\partial \mathbf{u}_k}\right)\frac{\partial \mathbf{u}_k}{\partial \mathbf{p}}\right],$$

where I_n is a $(n \times n)$ identity matrix, Δt_x is the time between each state evaluation, and $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ are the derivatives of the system equations with respect to each state and $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ with respect to the control signal.

Chapter 3

Optimal Path for Bearings-only Localization

In this chapter a single vehicle is considered, and the task is to localize a feature in the xy-plane with a bearings-only sensor. This is done by seeking control action of the vehicle that maximizes the information as described in Chapter 2.

The examples illustrate information as a performance metric and the effect of varied optimization time horizons. Different utility functions are evaluated and the difference between analytical implementation of the gradient and numerical calculation is examined. First the third dimension is ignored and the vehicle and the feature is defined in the xy-plane, then a constant altitude of flight is added. Also the case with n objects is considered, where the path is planned such that all objects are to be localized simultaneously.

3.1 System Models

This 2D single object localization example and the models are from Grocholsky [3]. The vehicle and the object are here assumed to be in the same plane.

3.1.1 Sensor Platform Model

The sensor is attached to a sensor platform (also called vehicle) 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 3.1. The vehicle's heading is modelled by ψ, the

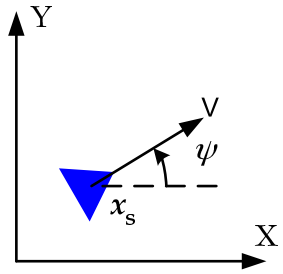


Figure 3.1. 2D vehicle model

angle between the head of the platform and the X-axis. The rate of change of the

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

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

$$u(t) = \dot{\psi} \tag{3.1}$$

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

In real life applications the coordinates of the vehicle is given by GPS/INS and navigation uncertainty should be included in the vehicle model. However, adding this to the model yields a much more complex problem to solve and is saved for future work.

3.1.2 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

$$Y(t) = P_f^{-1}(t). \tag{3.2}$$

Since a Gaussian distribution is assumed, the entropic information and the Fisher information are the same according to (2.10), and there is no need to distinguish between them.

The feature process model is

$$\dot{\mathbf{x}}_f(t) = \omega(t), \tag{3.3}$$

where the process noise $\omega(t)$ is a zero mean Gaussian process with uncorrelated covariance $Q(t)$. The process noise may have been omitted, if the object is stationary, but is included to allow flexible estimator design. The impact of incorrect information will be lower and the effect of the new information gained will be stronger.

The position uncertainty can be represented by an ellipse about the estimated location of the object. When planning the next optimization step, the path is calculated under the assumption that the true location of the object is the estimated location. This is not true in a real life application, since the object position is not known and could be in the outer range of its uncertainty ellipse, and the path would then not be optimal. This problem must be considered in future work. Instead of having a Gaussian distribution representing the uncertainty, one could instead use a sum of Gaussian with different weights which gives a more accurate model.

3.1.3 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, which could be calculated as the angle from the X-axis to the feature θ minus ψ as shown in Figure 3.2. The observation model is then, from (2.12) and (2.13),

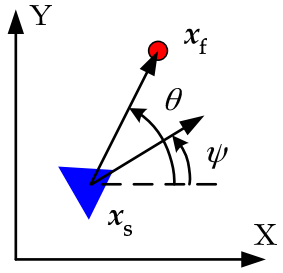


Figure 3.2. 2D observation model

$$z(t) = h(\mathbf{x}_f, \mathbf{x}_s) \tag{3.4}$$

$$h(t) = \theta(t) - \psi(t) + \nu(t) = \arctan_2(\frac{y_f - y_s}{x_f - x_s}) - \psi(t) + \nu(t) \tag{3.5}$$

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 according to (2.15):

$$\begin{aligned} H(t) &= \nabla_{\hat{\mathbf{x}}_\mathbf{f}} \mathbf{h}(\mathbf{x}_f, \mathbf{x}_s) \\ &= \Big[\frac{-(\hat{y}_f - y_s(t))}{(\hat{x}_f - x_s(t))^2 + (\hat{y}_f - y_s(t))^2}, \frac{\hat{x}_f - x_s(t)}{(\hat{x}_f - x_s(t))^2 + (\hat{y}_f - y_s(t))^2}\Big] \\ &= \frac{1}{\hat{r}(t)}\big[-\sin\hat{\theta}(t), \cos\hat{\theta}(t)\big], \end{aligned} \tag{3.6}$$

with $(\hat{x}_f, \hat{y}_f)$ as the estimated feature location, and $(\hat{r}, \hat{\theta})$ estimation of (r, θ) respectively. The resulting observed information is derived according to (2.16) as

$$I(t) = H^T(t) R^{-1} H(t). \tag{3.7}$$

3.1.4 System Equations

The state of the system consists of the platform model and the information matrix representing the uncertainty of the feature. The update of the vehicle states $\mathbf{x}_s$ is given in (3.1). The update of the information matrix is given by (2.21). A comparison between the feature model in (3.3) with the general expression in (2.18), yields $F(t) = 0$ and $G = I$ (and $B = 0$). The update of information is now reduced to

$$\dot{Y}(t) = -Y(t) Q Y(t) + I(t). \tag{3.8}$$

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

$$Y(t) = \begin{bmatrix} \mathrm{Y}_x & \mathrm{Y}_{xy} \\ \mathrm{Y}_{xy} & \mathrm{Y}_y \end{bmatrix}, \; I(t) = \begin{bmatrix} \mathrm{I}_x & \mathrm{I}_{xy} \\ \mathrm{I}_{xy} & \mathrm{I}_y \end{bmatrix} \text{ and } Q = \begin{bmatrix} \mathrm{Q}_x & 0 \\ 0 & \mathrm{Q}_y \end{bmatrix}. \tag{3.9}$$

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

$$\mathbf{x}_{info} = \begin{bmatrix} \mathrm{Y}_x \\ \mathrm{Y}_{xy} \\ \mathrm{Y}_y \end{bmatrix}. \tag{3.10}$$

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

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{x}}_s \\ \dot{\mathbf{x}}_{info} \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\psi}(t) \\ \dot{\mathrm{Y}}_x(t) \\ \dot{\mathrm{Y}}_{xy}(t) \\ \dot{\mathrm{Y}}_y(t) \end{bmatrix} = \begin{bmatrix} V\cos(\psi(t)) \\ V\sin(\psi(t)) \\ u(t) \\ -\mathrm{Y}_x^2(t)\mathrm{Q}_x - \mathrm{Y}_{xy}^2(t)\mathrm{Q}_y + \mathrm{I}_x(t) \\ -\mathrm{Y}_x(t)\mathrm{Q}_x\mathrm{Y}_{xy}(t) - \mathrm{Y}_{xy}(t)\mathrm{Q}_y\mathrm{Y}_y(t) + \mathrm{I}_{xy}(t) \\ -\mathrm{Y}_{xy}^2(t)\mathrm{Q}_x - \mathrm{Y}_y^2(t)\mathrm{Q}_y + \mathrm{I}_y(t) \end{bmatrix}. \tag{3.11}$$

The task is to reduce the uncertainty of the feature by maximizing information. Introduce for example the determinant as utility function $J(t)$ according to (2.26) as

$$J(t_f) = \det(Y(t_f)) = \mathrm{Y}_x(t_f)\mathrm{Y}_y(t_f) - \mathrm{Y}_{xy}^2(t_f). \tag{3.12}$$

and maximize it at the time horizon t_f.

3.1.5 Analytical Gradient

If the gradient would be given analytically, recall Section 2.4.3, the gradient of the utility function in (3.12) would be

$$\frac{\partial \phi(\mathbf{x}(t_f))}{\partial \mathbf{x}(t_f)} = \begin{bmatrix} 0 & 0 & 0 & \mathrm{Y}_y & -2\mathrm{Y}_{xy} & \mathrm{Y}_x \end{bmatrix}. \tag{3.13}$$

The gradient of the systems equations (3.11) is needed, and is given by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & -V\sin(\psi) & \dots \\ 0 & 0 & V\cos(\psi) & \dots \\ 0 & 0 & 0 & \dots \\ \frac{4(\hat{y}_f-y)^2(\hat{x}_f-x)}{\sigma^2 r^6} & \frac{4(\hat{y}_f-y)^3}{\sigma^2 r^6} - \frac{2(\hat{y}_f-y)}{\sigma^2 r^4} & 0 & \dots \\ -\frac{4(\hat{y}_f-y)(\hat{x}_f-x)^2}{\sigma^2 r^6} + \frac{(\hat{y}_f-y)}{\sigma^2 r^4} & -\frac{4(\hat{y}_f-y)^2(\hat{x}_f-x)}{\sigma^2 r^6} + \frac{(\hat{x}_f-x)}{\sigma^2 r^4} & 0 & \dots \\ \frac{4(\hat{x}_f-x)^3}{\sigma^2 r^6} - \frac{2(\hat{x}_f-x)}{\sigma^2 r^4} & \frac{4(\hat{y}_f-y)(\hat{x}_f-x)^2}{\sigma^2 r^6} & 0 & \dots \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -2\mathrm{Y}_x\mathrm{Q}_x & -2\mathrm{Y}_{xy}\mathrm{Q}_y & 0 \\ -\mathrm{Y}_{xy}\mathrm{Q}_x & -\mathrm{Y}_x\mathrm{Q}_x - \mathrm{Y}_y\mathrm{Q}_y & -\mathrm{Y}_{xy}\mathrm{Q}_y \\ 0 & -2\mathrm{Y}_{xy}\mathrm{Q}_x & -2\mathrm{Y}_y\mathrm{Q}_y \end{bmatrix} \tag{3.14}$$

where $[x \quad y]^T$ is the vehicle's position, $[\hat{x}_f \quad \hat{y}_f]^T$ is the feature's estimated position and r is the distance between the vehicle and the target, and

$$\frac{\partial \mathbf{f}}{\partial u} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T. \tag{3.15}$$

These are the equations needed to calculate the gradient $\nabla_{\mathbf{p}} J$ according to (2.42). One could realize that if the model is further complicated, it would be quite difficult to derive the expressions needed for the analytical gradient. Instead one could let *Matlab* calculate the gradient numerically.

3.2 Simulations

The initial conditions in (3.1) give the starting position of the vehicle and the starting angle as the angle between the heading of the vehicle and the x-axis. Let the vehicle start at the origin, and specify a desired starting angle. In the simulation examples the starting angle was set to $\pi/2$ [rad]

$$\mathbf{x}_s(0) = \begin{bmatrix} 0 & 0 & \pi/2 \end{bmatrix}^T.$$

There must also be some starting information, since in order to plan how to localize a target, some information is needed. In other words, there is a need to know that the target exists, otherwise it is hard to plan the path. The settings used in the simulations are:

$\mathbf{x}_f = [10 \quad 10]\ m$	Feature location (stationary).
$V = 1\ m/s$	Constant speed.
$R = \sigma^2$, $\sigma = 2.5^\circ$	Observation noise.
$Y(0) = \mathrm{I}_{2\times 2} \cdot 10^{-3}$	Low starting information.
$Q = \mathrm{I}_{2\times 2} \cdot 10^{-6}$	Process noise.
$m = 2$	Number of parameterized control signals. in each optimization step.
$t_f = 1\ s$	Optimization time horizon.
$\lvert u_i \rvert \leq 1\ rad/s$	Control signal bounds.
$J(t_f) = \det(Y(t_f))$	Utility function.

These values are chosen for simplicity and are not values for real applications, it is the principle that is interesting. The resulting path is presented in Figure 3.3 and the observed information and the parameterized control signal are shown in Figure 3.4. The control signal is bounded and then the optimization is solved by the function *fmincon* in Matlab Optimization Toolbox.

The solution is calculated as following. In each step, a control signal, parameterized into m equal long parts, will be calculated such that the information at the time horizon t_f is maximized, given the system equations and subject to constraints. The first optimum will be a trajectory from the origin of length $V \cdot t_f = 1\ m$, and this point is reached by the optimal control signal consisting of $m = 2$ steps. The first part of the control signal u_1 is valid between the time 0 and $t_f/2$ and the next part u_2 is valid between $t_f/2$ and t_f. After the first optimum is reached, and the states $\mathbf{x}$ have been updated, a new optimization is done. The uncertainty of the feature location is plotted as an ellipse about the feature, and the procedure is terminated when the uncertainty

is so low that the object could be said is localized. The total number of optimizations are the number of 'x':s in the figure, a total of 25, that is the number of optimizations required for localizing the target in this case. The target is said to be localized if the information is higher than a pre-defined value.

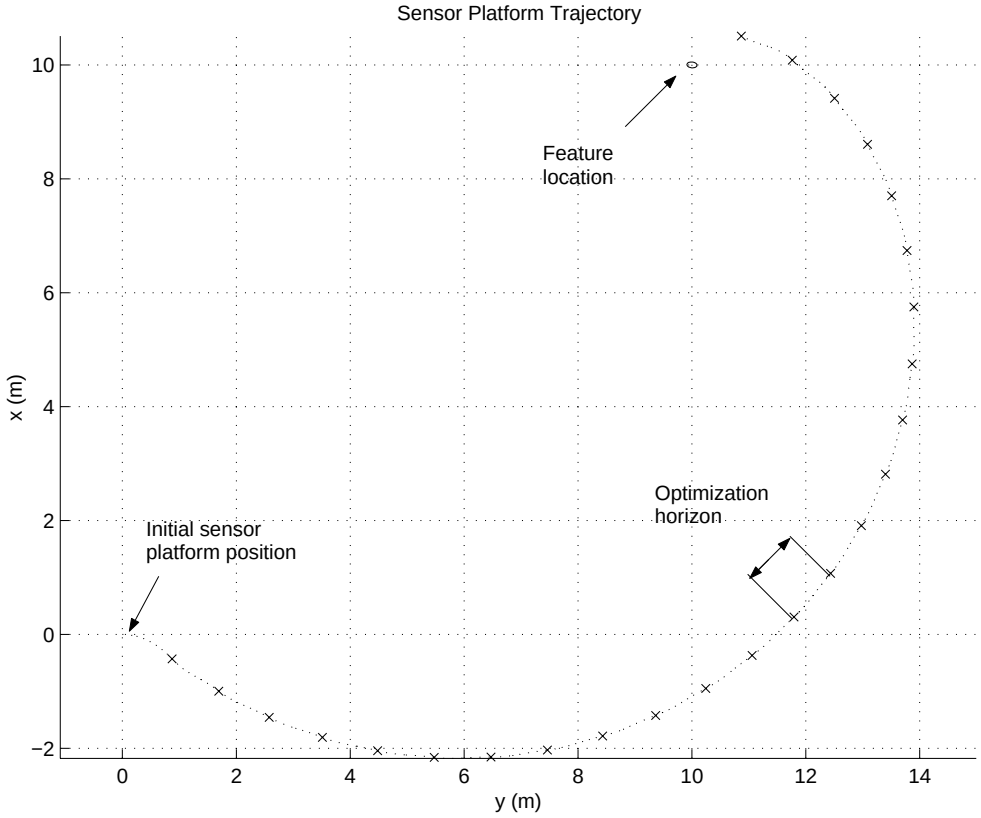


Figure 3.3. Trajectory of the sensor platform. Each 'x' marks a new optimization. The feature location's uncertainty is an ellipse about its true location.

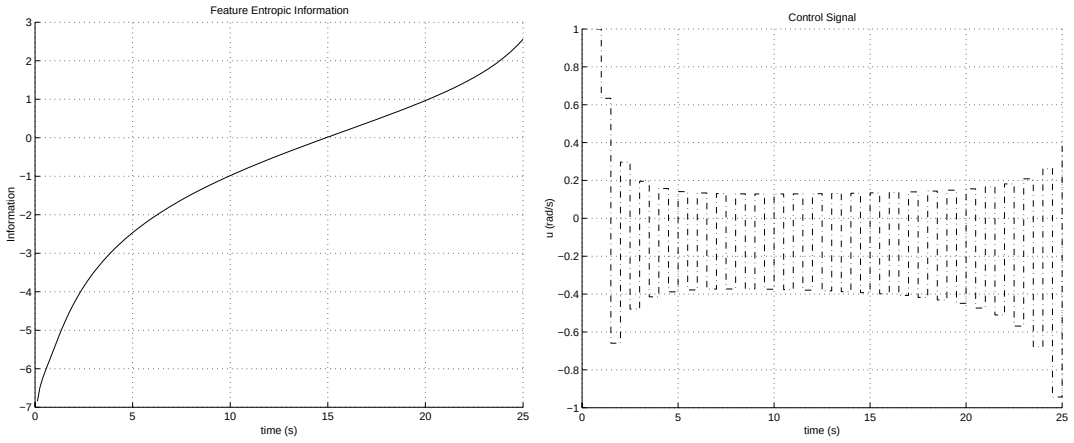


Figure 3.4. Feature entropic information and the parameterized control signal.

The path is shaped as a spiral, which is reasonable since the sensor is a bearings-only sensor. In the beginning, the planning steps are approximately in the orthogonal

direction with respect to the object. In addition, the sensor platform is also moving closer to object.

The path is the planned path for the sensor platform. However, since there are no uncertainties of the platform's position, it will also be the performed path by the platform. The control signal varies all the time and it is undesirable since the controller would be worn out. But it is understandable that the solution gives a varying control signal, since it plans very short ahead. To avoid this problem, one could formulate penalty functions on varying control signals or use a different control signal parametrization.

The calculation time is highly dependent on the number of control parameterizations m. The calculation time is also dependent on the tolerance chosen for the optimization and the ODE solver.

The use of information as a performance index is intuitive, since the more information you have of a feature, the more certain are you of its location. It is also a convenient criterion for the optimization process, since when the information is maximized, the uncertainty of the feature location is minimized.

3.2.1 Plotting the Criterion Function

To illustrate the complexity of the optimization problem, the criterion function could be plotted over the first optimization step, that is over the first parameterized control signal $[u_1\ u_2]$. Recall the criterion function used here as the determinant of the information matrix from (3.12)

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

This function is plotted for two cases of bound on the control signal, first bounded as $-10\ rad/s \leq u_i \leq 10\ rad/s,\ i = 1, 2$, plotted in Figure 3.5. The optimizer is trying to find the maximum, and the problem is that there are several local maxima, and there is a risk that the optimizer will find a local maximum instead of the global. The reason why there are so many maxima is that when having a large bound on the control signal, the vehicle could either turn around with a high control signal and end up about the same place as a low control signal. For this example, tighten the bound on the control signal to $-1\ rad/s \leq u_i \leq 1\ rad/s,\ i = 1, 2$, for the same situation yields the situation in Figure 3.6. The problem of many maxima is not solved by this, but there are less maxima than for the case with looser bounds.

The reason why the criterion function was plotted, is to illustrate that the optimization is not perfect, but since it is an application of engineering, one would be satisfied with a good solution and not the best in the theoretical sense. The *Matlab Optimization Toolbox* functions *fminunc* and *fmincon* use line search when they cannot solve the problem otherwise. The line search algorithm has problem with local maximum, and instead one should implement and test other optimization methods. This is a very important task in the future work.

3.2.2 The Effect of Optimization Time Horizon

The sensor platform's trajectory is affected by the time horizon t_f in the optimization. Figure 3.7 shows a comparison between three different time horizons. The number of parameterizations of the control signal m differs, since in a longer optimization, each parameterization of the control signal is valid longer. The three time horizons are short

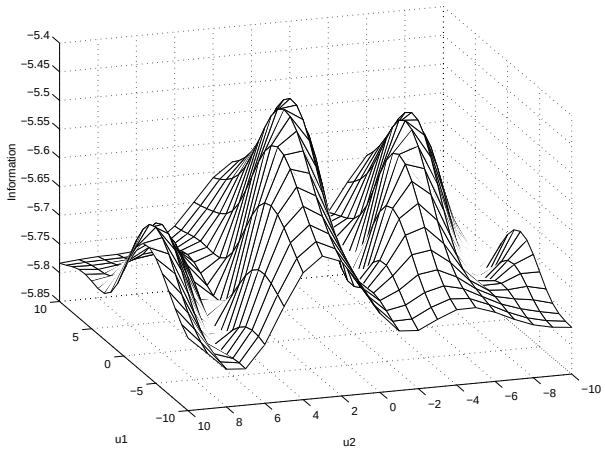


Figure 3.5. The criterion function over a parameterized control signal $[u_1\ u_2], -10 \leq u_i \leq 10$

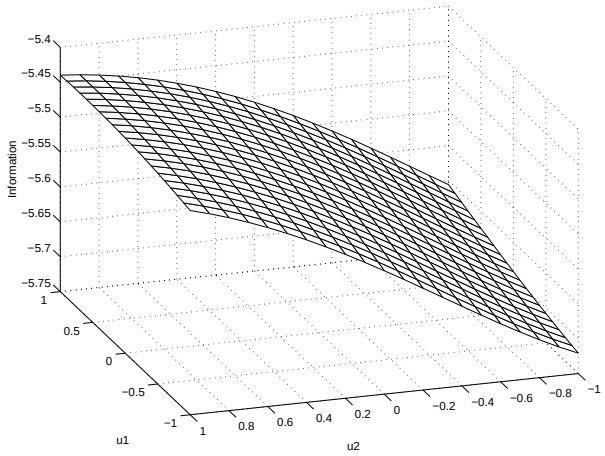


Figure 3.6. The criterion function over a parameterized control signal $[u_1\ u_2], -1 \leq u_i \leq 1$

(1 s) for 16 optimization steps, intermediate (4 s) for 4 optimization steps and long (8 s) for 2 optimization steps. The reason is that the total time would be 16 s for all cases. The cases can be summarized in Table 3.1.

Case	1	2	3
Time horizon $t_f(s)$	1	4	8
Number of optimizations	16	4	2
Total time (s)	16	16	16
Control parameters m	2	4	8

Table 3.1. Details of the three cases used to investigate the effect of optimization time horizon.

As can been seen, the platform with long optimization time travels more directly to the feature. It will not gain as much information in the beginning as in the case with short optimization time, but since it plans further in the future, the information at the time horizon will increase as seen in Figure 3.8. According to Figure 3.8, a long time

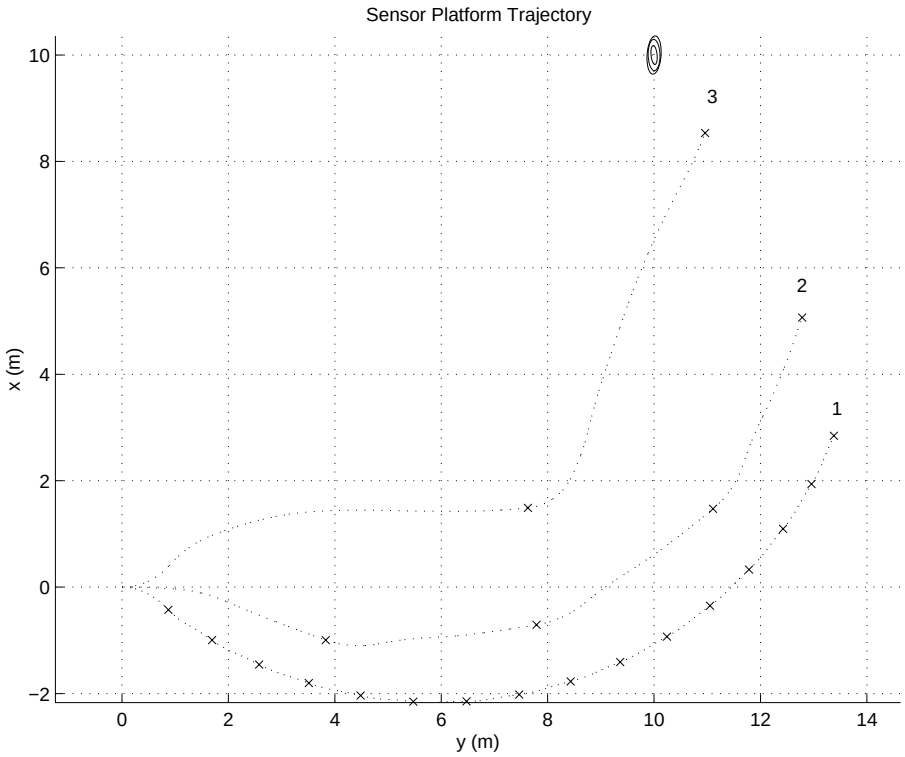


Figure 3.7. The trajectories for platforms with different optimization times for the three cases in Table 3.1.

horizon is preferable compared to a short time horizon. But there are problems with a long horizon. The calculation time was different for the three cases. The computation time of the second and third case was 30% and 350% longer than for the first case, respectively.

3.2.3 The Effect of Prior Information

In the simulations, the starting information was set to 10^{-3} in both the x- and y-direction. The starting information is a measure of how much is known about the object when planning the first step in terms of uncertainty. When knowing more about the object, *i.e.* the starting information was set to 1 in both directions, the resulting path is seen in the left part of Figure 3.9. It will travel in the direction of the object at first, since it will not gain that much information by taking a step orthogonally. Instead it needs to move closer, which it does until one point where it turns and continues like in the first simulation in Figure 3.3.

In the simulation shown in the right part of Figure 3.9 the starting information was set to 1 in the x-direction and 10^{-3} in the y-direction. This is the case when knowing one coordinate of the object's position and it is needed to reduce the uncertainty in the other direction. The vehicle is trying to reduce the uncertainty in the y-direction by moving in the direction which is orthogonal to the initial uncertainty, that is in the y-axis. The vehicle continues until the uncertainty in the y-direction is about the same as the x-axis and it turns to reduce the uncertainty in both directions.

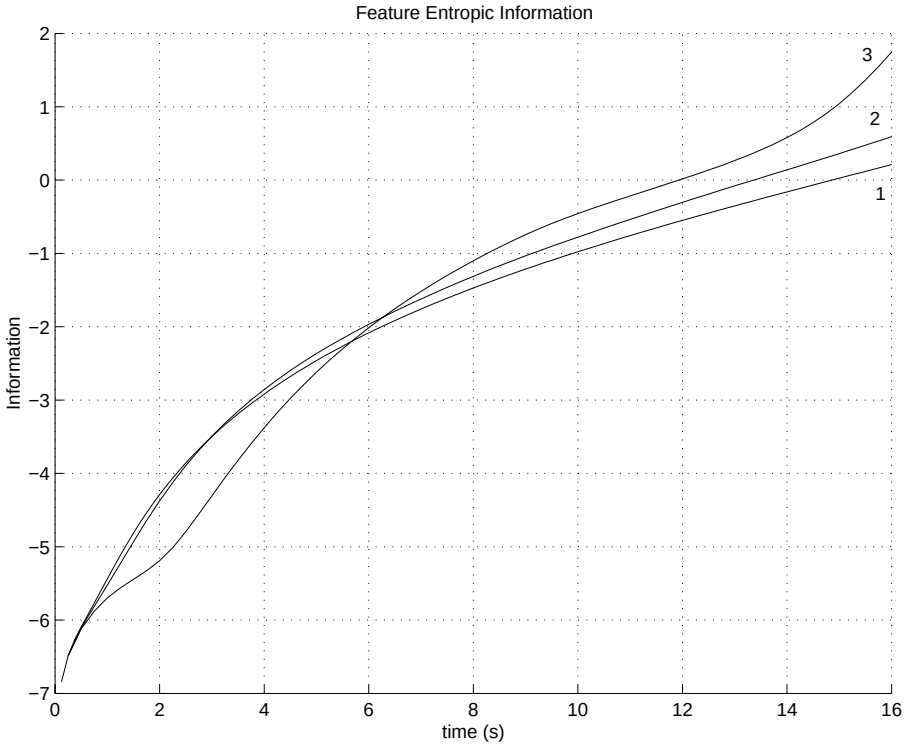


Figure 3.8. The information for platforms with different optimization times for the three cases in Table 3.1.

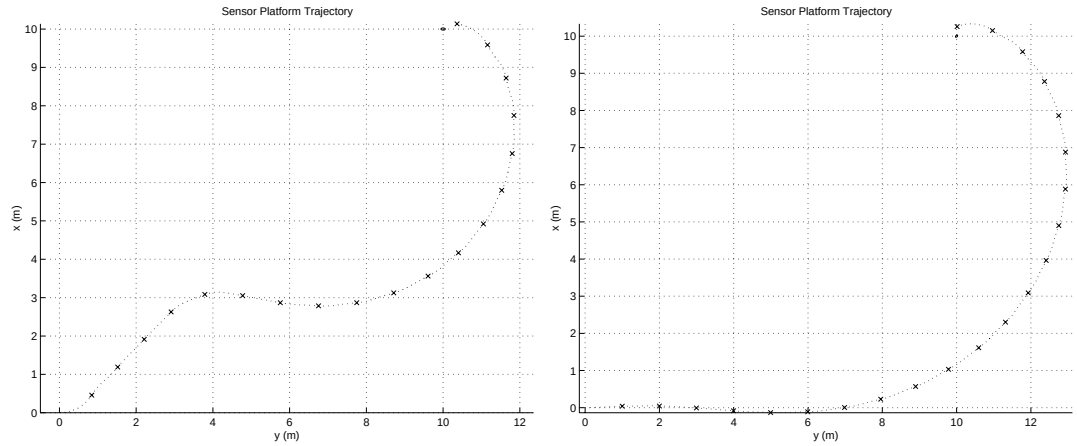


Figure 3.9. *Left:* The effect of high starting information. *Right:* High starting information in x-direction and low starting information in y-direction.

3.2.4 Evaluation of the Utility Function

In the simulations below, three different utility functions are evaluated. The details of the different simulations are given in Table 3.2. The first one is maximizing the determinant of the information matrix, and tried for two cases where the gradient of the utility function is given analytically, case 1, and by letting *Matlab* approximate the gradient numerically, case 2. The optimization is made by *Matlab*'s function *fmincon*, where it is possible for the user to choose whether the gradient is given explicitly or not,

and for a simple case like this with only one object, the gradient could be calculated as described in Section 2.4.3. Case 3 is the path from the trace of the information matrix's inverse and case 4 is maximizing the minimum of the eigenvalues. The resulting paths are shown in Figure 3.10.

Case	1	2	3	4
Utility function	$\max(\det Y)$	$\max(\det Y)$	$\min(\mathrm{tr} Y^{-1})$	$\max(\min(\mathrm{eig} Y))$
Gradient computation	Analytical	Numerical	Numerical	Numerical

Table 3.2. Details of the four cases used to investigate the effect of different utility functions.

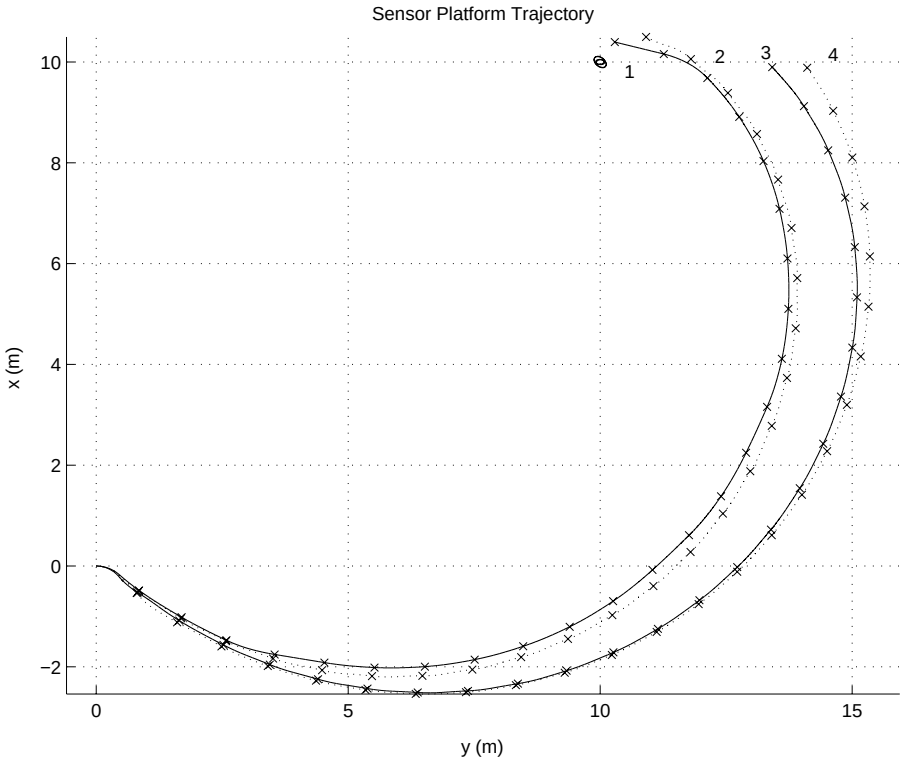


Figure 3.10. Sensor platform trajectory calculated from different utility functions: 1) $\max(\det Y)$ with gradient analytically, 2) $\max(\det Y)$ with gradient numerically, 3) min(trace Y^{-1}), 4) $\max(\min(\mathrm{eig} Y))$

All paths have the same basic shape, a spiral. The time horizon was set to 1 s for simplicity and except for the utility function and gradient, all other simulation parameters are the same. The entropic information values are plotted in Figure 3.11. However, this is an unfair comparison tool. Not very surprising $detY$ gives the largest final information value, but it is more difficult to decide which utility function that solves the localization problem best.

However, the analytical and numerical implementation of the gradient in the $detY$ case can be compared. The analytical implementation is better, since information is higher, but the difference is small. The major advantage of the analytical case is the computational time, but when the model gets more complicated, it is difficult or impossible to derive the analytical expression explicitly.

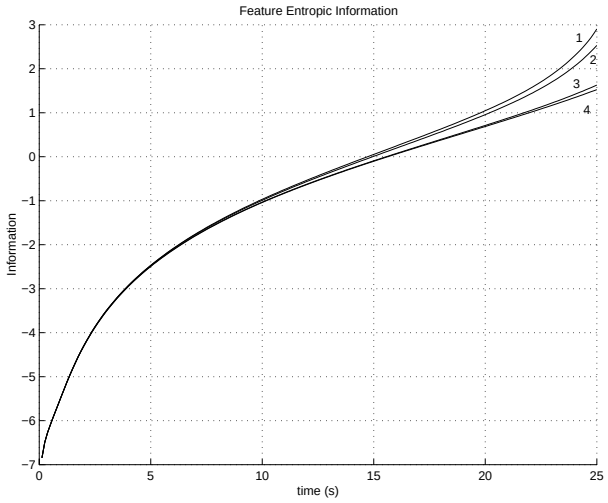


Figure 3.11. Information from the different utility functions: 1. max(det$Y$) with gradient analytically, 2. max(det$Y$) with gradient numerically, 3. min(traceY^{-1}), 4. max(min(eig$Y$))

3.3 n Objects

Following the methods of this chapter, it seems that the vehicle could successfully localize an object. To make things more interesting, the vehicle must be able to handle the case of n objects on the ground.

3.3.1 Modelling two Objects

First the model should be extended to the case with two objects. A symmetric information matrix Y is now of (4×4) as:

$$Y(t) = \begin{bmatrix} \mathrm{Y}_{x_1} & \mathrm{Y}_{x_1y_1} & \mathrm{Y}_{x_1x_2} & \mathrm{Y}_{x_1y_2} \\ \mathrm{Y}_{x_1y_1} & \mathrm{Y}_{y_1} & \mathrm{Y}_{x_2y_1} & \mathrm{Y}_{y_1y_2} \\ \mathrm{Y}_{x_1x_2} & \mathrm{Y}_{x_2y_1} & \mathrm{Y}_{x_2} & \mathrm{Y}_{x_2y_2} \\ \mathrm{Y}_{x_1y_2} & \mathrm{Y}_{y_1y_2} & \mathrm{Y}_{x_2y_2} & \mathrm{Y}_{y_2} \end{bmatrix}. \tag{3.16}$$

With just one more object, there were seven new states introduced. There is now three states representing the vehicle, same as in (3.1) and ten states from the information matrix above. The question arises whether some states, especially the cross states $\mathrm{Y}_{i_1j_2}$, could be zero. Recall the information rate of change from (3.8) as

$$\dot{Y}(t) = -Y(t)QY(t) + I(t), \tag{3.17}$$

and the same observation model as for one object from (3.4) now extended as

$$\begin{aligned} \mathbf{z}(t) &= [h_1(\mathbf{x}_{f_1}, \mathbf{x}_s), h_2(\mathbf{x}_{f_2}, \mathbf{x}_s)]^T \\ h_1(t) &= \theta_1(t) - \psi(t) + \nu_1(t) \\ h_2(t) &= \theta_2(t) - \psi(t) + \nu_2(t). \end{aligned} \tag{3.18}$$

The linearized observation matrix $H(t)$ is

$$
\begin{aligned}
H(t) &= \nabla_{\hat{\mathbf{x}}_\mathbf{f}} \mathbf{h}(\mathbf{x}_f, \mathbf{x}_s) \\
&= \begin{bmatrix} -\frac{\sin\theta_1(t)}{r_1(t)} & \frac{\cos\theta_1(t)}{r_1(t)} & 0 & 0 \\ 0 & 0 & -\frac{\sin\theta_2(t)}{r_2(t)} & \frac{\cos\theta_2(t)}{r_2(t)} \end{bmatrix}.
\end{aligned} \tag{3.19}
$$

The R matrix is the covariance matrix of the two noise processes $\nu_1(t)$ and $\nu_2(t)$ as

$$
R = \begin{bmatrix} \text{var}(\nu_1) & \text{cov}(\nu_1, \nu_2) \\ \text{cov}(\nu_1, \nu_2) & \text{var}(\nu_2) \end{bmatrix}. \tag{3.20}
$$

Combining (3.19) and (3.20) gives the observed information according to (3.7) as

$$
I(t) = H^T(t) R^{-1} H(t) \tag{3.21}
$$

If the two noise processes $\nu_1(t)$ and $\nu_2(t)$ are independent, there are no covariances and the observed information matrix $I(t)$ would become a block matrix as

$$
I(t) = \begin{bmatrix} \mathrm{I}_1(t) & 0 \\ 0 & \mathrm{I}_2(t) \end{bmatrix}, \tag{3.22}
$$

where $\mathrm{I}_\mathrm{i}(t), i = 1, 2$ are (2×2) matrices representing the observed information from object i. From this, a model with n objects could be easily be derived by just extending the observed information with block matrices. This is not sufficient for the cross states $\mathrm{Y}_{i_1 j_2}$ in (3.16) to be zero. The term in the update law corresponding to the loss due to the process noise

$$
-Y(t)QY(t)
$$

must also be considered. The process noise is modelled with a block matrix, each block containing process noise for the two objects, extended from (3.9) to

$$
Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} = \begin{bmatrix} \mathrm{Q}_{x_1} & 0 & 0 & 0 \\ 0 & \mathrm{Q}_{y_1} & 0 & 0 \\ 0 & 0 & \mathrm{Q}_{x_2} & 0 \\ 0 & 0 & 0 & \mathrm{Q}_{y_2} \end{bmatrix}. \tag{3.23}
$$

If the cross states $\mathrm{Y}_{i_1 j_2}$ are set to zero, then the term $Y(t)QY(t)$ would be a block matrix as

$$
Y(t)QY(t) = \begin{bmatrix} Y_1(t)Q_1Y_1(t) & 0 \\ 0 & Y_2(t)Q_2Y_2(t) \end{bmatrix}. \tag{3.24}
$$

Since both terms, $I(t)$ and $Y(t)QY(t)$, in the update of the information matrix in (3.17) are block matrices with non-diagonal blocks as zeros, there will not be any information update from the cross states $\mathrm{Y}_{i_1 j_2}$. With no update of those states, they will not affect the optimization and therefore those states could be set to zero. Then the information matrix in (3.16) would be a diagonal matrix, with (2×2) block matrices on the diagonal representing the information of each object as

$$
Y(t) = \begin{bmatrix} \mathrm{Y}_1(t) & 0 \\ 0 & \mathrm{Y}_2(t) \end{bmatrix}, \tag{3.25}
$$

where

$$Y_i(t) = \begin{bmatrix} \mathrm{Y}_{x_i} & \mathrm{Y}_{x_i y_i} \\ \mathrm{Y}_{x_i y_i} & \mathrm{Y}_{y_i} \end{bmatrix}.$$

The matrix in (3.25) could easily be extended to n objects, with adding block matrices for each object.

3.3.2 Simulation with two Objects

In these simulations, the noise processes $\nu_1(t)$ and $\nu_2(t)$ are independent of each other, and the resulting path is examined. An information matrix is constructed as in (3.25), and from this the determinant is taken as utility function, and information could be maximized.

The observed information is inverse proportional to the squared distance and angle dependent, and there is a risk that the optimizer will only consider the nearest object, since it will gain much information by getting closer all the time. In order to avoid such problem, the object is removed from the model when localized. Thus, the states representing the information from the objects are removed from the system equations and the new information matrix is simply the information matrix from the non-localized object.

In the following simulation two features was placed in [10 10] and [10 0], and the vehicle starts in the origin with heading in the Y-axis direction. The optimization time horizon t_f is set to 1 s in the first simulation and to 8 [s] in the second simulation, and the utility function is the determinant of the information matrix. The noise is the same for the two objects.

The resulting paths are shown in Figure 3.12, where the short time horizon to the left and the long time horizon to the right. In both cases, the resulting path is where information is gained the most from both objects. For the short time horizon, the first step is as much orthogonal to both objects as possible, and at a certain point, it will gain more information to "zoom" the object closest and will continue until the first object is localized. Next the full attention will focus on the remaining object and it will be the case of localizing one object. For the long time horizon, it could place its first optimum close and orthogonal to both objects.

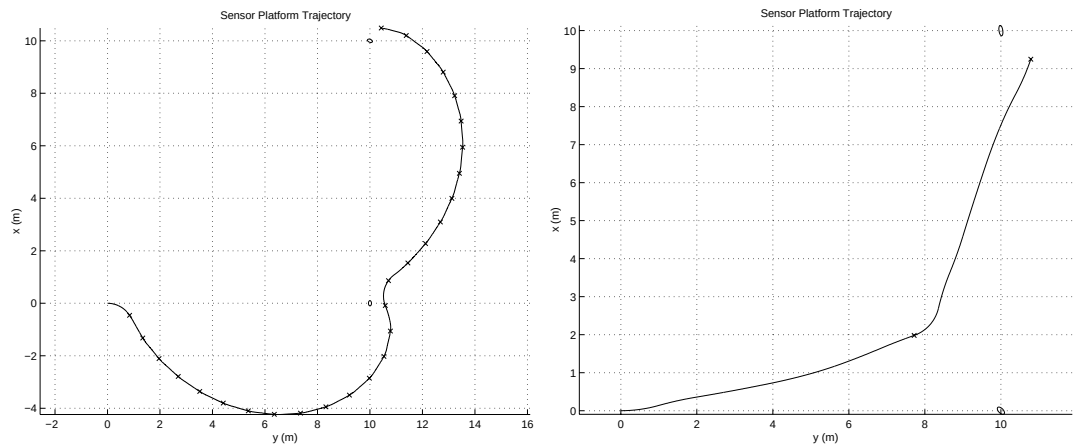


Figure 3.12. Left: Simulated sensor platform trajectory with two objects, time horizon 1 s. Right: Simulated sensor platform trajectory with two objects, time horizon 8 s.

From the two paths, it is very tempting to make the conclusion that the long time horizon is better than the short time horizon. But if the threshold for localization is set higher, the long time horizon must plan a long time ahead close to an object. In Figure 3.13 a path is shown, where the problem with a long time horizon is seen. The first two points are the same as the right part of Figure 3.12, but then the path must be planned a long time ahead close to the target, until it is localized and then turns back to the other object, which is not very effective. The short optimization horizon is not much more effective, but it takes less time to calculate. This example was given to illustrate that a long optimization horizon is not always preferable as one could assume be just looking at Figure 3.12.

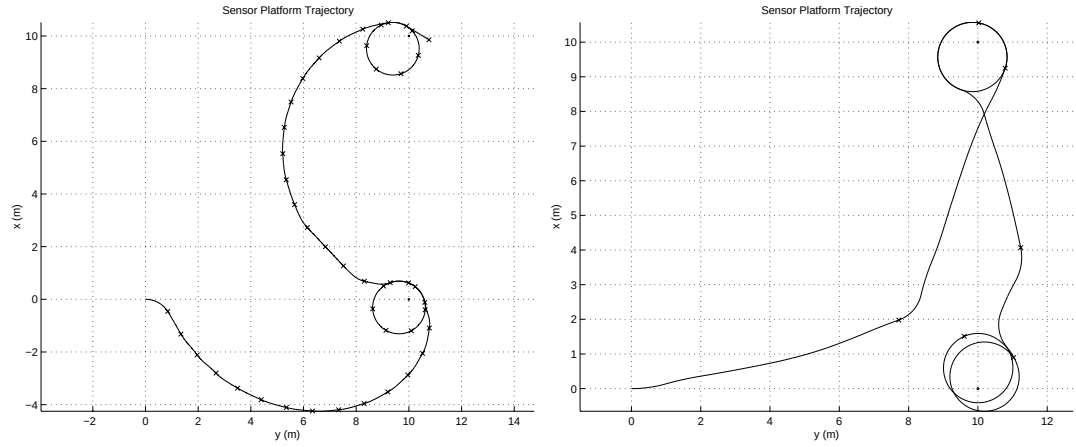


Figure 3.13. With long time horizon (right), the UAV must turn back to the initially nearest object compared to the short time horizon (left).

3.3.3 Extension to n Objects

The model can now be extended to handle n objects, since it is about adding and removing states to the model and hence to the information matrix. When modelling n objects, each object i has its own (2×2) information matrix $\mathrm{Y_i}$. Then a global information matrix could be constructed by putting these on the diagonal as

$$Y = \begin{bmatrix} \mathrm{Y}_1 & 0 & \dots & 0 \\ 0 & \mathrm{Y}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{Y}_\mathrm{n} \end{bmatrix}. \tag{3.26}$$

From this matrix, all discussed utility functions could be formulated such as maximizing the determinant and minimizing the trace of its inverse.

A simulation is made with five objects randomly placed in the xy-plane. The same parameters is used as for the simulation with two objects and the result is shown in Figure 3.14.

There are most certainly many maxima of the utility function, but as said before, any local maximum would be good enough and new optimizations are done until localization.

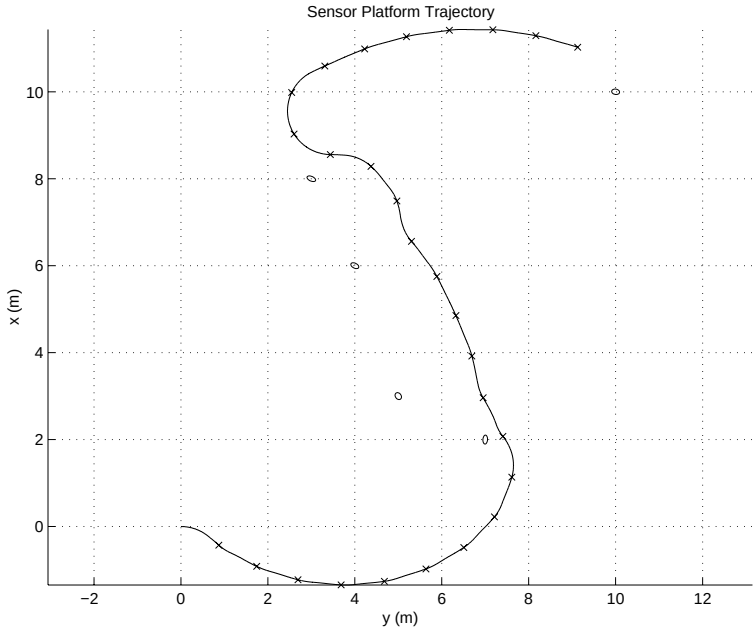


Figure 3.14. Sensor platform trajectory with five objects.

3.4 3D-modelling

The previous simulations has been in two dimensions, but since a UAV has a certain height of flight, it is natural to extend the models to three dimensions. A new control signal of the vehicle that controls the flight altitude could be introduced, but to keep the optimization complexity down we assume that the flight altitude is constant.

There is also a singularity in the 2D model, since the observed information is inverse proportional to the squared distance to the feature. This means that the vehicle could not be close to the object. The problem was solved by removing object when localized or by bounding the control signals, so the vehicle never could come too close. In this section, a method is derived where the information is observed in a different coordinate system and then transformed into Cartesian coordinates, and the singularity is avoided.

3.4.1 3D Object

As described previously, the uncertainty of an object's location is represented by its information matrix $Y(t)$. The concept is taken into three dimensions with an extended information matrix:

$$Y(t) = \begin{bmatrix} \mathrm{Y}_x & \mathrm{Y}_{xy} & \mathrm{Y}_{xz} \\ \mathrm{Y}_{xy} & \mathrm{Y}_y & \mathrm{Y}_{yz} \\ \mathrm{Y}_{xz} & \mathrm{Y}_{yz} & \mathrm{Y}_z \end{bmatrix}. \tag{3.27}$$

By modelling an object in three dimensions, the uncertainty in the z-direction will also be considered.

The information matrix is symmetric as before and it is sufficient with six states to represent an object. The update of the information matrix is still:

$$\dot{Y}(t) = -Y(t)QY(t) + I(t), \tag{3.28}$$

but both the process noise Q and the observed information I are now (3×3)-matrices.

3.4.2 Information in 3D

Let the UAV fly in the xy-plane and the distance to the ground, z_s, is kept constant. An object on the ground could be described by the angle α same as θ in the two dimensional case and the angle β, which is the angle from the xy-plane to the object, shown in Figure 3.15.

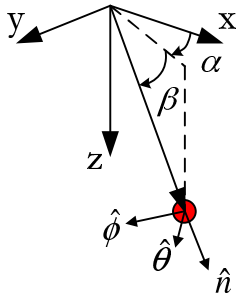


Figure 3.15. 3D modelling of a feature described by the angles α and β. Local coordinates for observing information.

Introduce a sensor direction $\hat{n}$ as the vector from the vehicle to the feature. At this moment, the sensor is considered to "point" at every direction and a camera will be introduced later on in Chapter 6. For now, the sensor has unlimited field of view. A coordinate system is set in the end of the vector at the object, as in Figure 3.15. The information is observed orthogonal to the direction of the camera, that is in the $\hat{\theta}$ and the $\hat{\phi}$-direction. In other words, the variance of the noise process in the $\hat{n}$-direction is infinite. The variance in the $\hat{\theta}$- and the $\hat{\phi}$-directions are σ_θ^2 and σ_ϕ^2 respectively. The inverse of the covariance matrix R could be set as

$$R^{-1} \approx \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_\theta^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_\phi^2} \end{bmatrix}, \tag{3.29}$$

if the noise processes are uncorrelated. The information could now be transformed into the global Cartesian coordinates, by using the angles α and β. The angles α and β are the rotation about the z-axis and the y-axis, respectively. The transformation matrices are described as

$$T_2(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \text{ and} \tag{3.30}$$

$$T_3(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.31}$$

The resulting rotation matrix which transforms the local coordinates into the global coordinates would be

$$T = T_3(-\alpha)T_2(-\beta) \tag{3.32}$$

The information expressed in the local reference system can now be transformed to the global reference system

$$I_{global} = T I_{local} T^T. \tag{3.33}$$

This yields in a (3×3) information matrix, used in the update law in (3.28), without a singularity. The H in this case is simply the identity matrix in the local coordinates, since the feature location and the observed location is the same point.

A simulation with the same properties as for the two dimensional model in Figure 3.3 is shown in Figure 3.16 for two different heights. In the left picture the height of flight z_s is 1 m. Hence, the β-angle is small and the simulation result is similar to the 2D case. The flight height affects more when it is higher, and in the right picture of Figure 3.16 it was set to 10 m. The rotation matrix about the y-axis differs now from the identity matrix, and there will be essential amount of information in all states in the information matrix (3.27) and this will affect the eigenvalues of the information matrix and hence the utility function. The scalar measure of the utility function will be lower with increasing flight height and could be interpreted as the higher the UAV flies, the less information is seen about the object since it will be farther away. The resulting path with higher flight height is longer, since there will be less information gained.

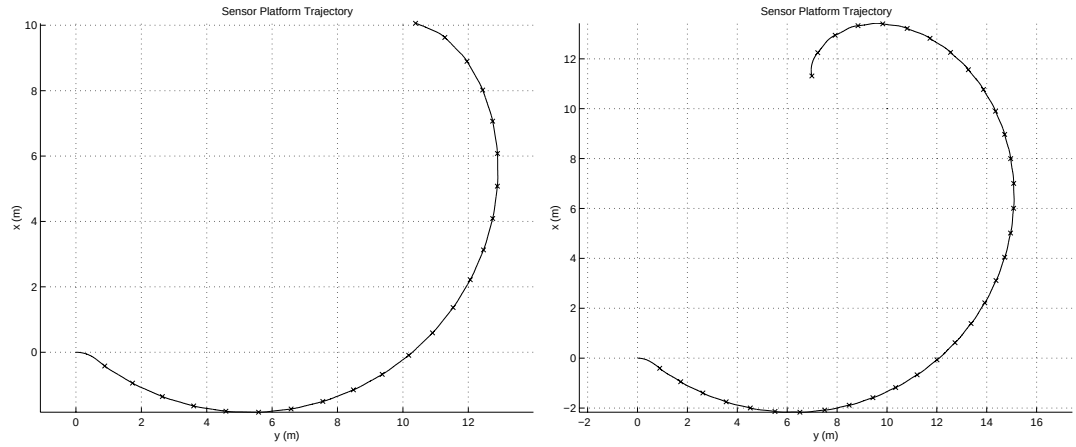


Figure 3.16. The trajectory for an UAV with different flight heights. Left: $z_s = 1$ m. Right: $z_s = 10$ m. Feature located in $(10, 10)$.

3.5 Conclusions and Remarks

The use of information as a performance metric is reasonable since the more information about an object is better when trying to localize it, and the path of the moving platform is planned such that information is maximized.

The path is planned by optimization in each step. That is the reason the control signal varies a lot, and there is no penalty in choosing the control signal allowing the most greedy solution in terms of information. The optimization uses the *Matlab Optimization Toolbox*, which uses line search in the examples and there is a risk to get stuck in local optimum when the optimization is done to a non-convex and non-linear utility function. Introduce bounds on the control signal in the optimizer, reduces the number of local optima but the problem does not disappear, instead one could accept any local minima

as a good solution and make another optimization step, until the information is high enough.

There is the difficulty of choosing the optimization parameters. The longer time horizon, the faster information gain, but it takes longer time to compute and the there is the problem of planning the path close to an object with long time horizon. Therefore the short time horizon is chosen in the next chapter, where an area is about to be explored. The utility function which gives the best result with respect to information is the determinant, with the trace of the inverse as second best. By calculating the gradient analytically, the calculation time decreases, but the resulting path is about the same as for the numerical approximation of the gradient. However, the analytical gradient is hard to implement in an area search in the next chapter, and therefore the numerical approximation is chosen. The vehicle was also able to localize n objects, and the resulting path considered that all objects were to be localized simultaneously.

Finally the model was extended to three dimensions where the information is observed in a local coordinate system about the object and then transformed into global coordinates and thus avoiding the singularity in the linearization.

Chapter 4

Area Exploration

A common task in surveillance and reconnaissance is area coverage. The area is represented by a number of grid points, and with each grid point is associated a quantity which may be measured under the influence of noise, so an information matrix could be constructed. The flight path is calculated from utility functions derived from the information matrix.

4.1 Area Coverage

The area coverage is usually done after a pre-defined path. The UAV is sweeping the area back and forth until the whole area has been covered. However, we want the UAV to search for objects in the area and once a object is found the UAV should localize it. By calculating the next step from an information matrix consisting of information about both the area and the objects, the UAV is able to react on objects on the ground. The vehicle will now take the best possible step at the time horizon in terms of information.

Consider for example a game of battleships. In those games, you are sweeping your opponents area, and if you hit something, you know that there is a ship and you continue to search that part of the area until the ship has been sunken. Instead of ships and sinking, change the terms to objects and localizing and that will be the case of the UAV.

4.1.1 Area Model

The area is represented by a number of *grid points*, thus the area is discretized. The information from an object was modeled in (3.27) by a (3×3) information matrix, that is by six states. It would be possible to use the object model as the grid point model. Grid points are not going to be localized as the objects, since the grid points are set out and therefore completely known. However, the (3×3) information matrix is suitable since it will capture the properties of how well the grid point have been seen, including if it have been seen from significantly different directions.

The downside of using the six states model for the grid points is, of course, that the complexity of a discretized area with N grid points will grow large to $6N$ states. An alternative is to let each grid point be represented by just one state. This one state has no physical interpretation. We lost the strive for seeing the grid points from significantly different directions, but we keep the complexity down, the total number of states will be N. Hence, calculation time will be much shorter.

In this report we use the one state grid point model. Thus, the state vector $\mathbf{x}$ consists of the three states representing the vehicle, that is $\mathbf{x}_s$ and the N states representing the grid points $\mathbf{x}_{gp}$ as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_s \\ \mathbf{x}_{gp} \end{bmatrix}. \tag{4.1}$$

The state vector consists now of $(3 + N)$ states.

The information matrix consists of the states representing the information and is constructed in the same manner as for n objects as

$$Y_{gp}(t) = \begin{bmatrix} \mathrm{Y}_{\mathrm{gp_1}}(t) & 0 & \dots & 0 \\ 0 & \mathrm{Y}_{\mathrm{gp_2}}(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathrm{Y}_{\mathrm{gp_N}}(t) \end{bmatrix}, \tag{4.2}$$

where the index $\mathrm{gp_i}$ denotes grid point i. The noise in each observation is considered uncorrelated and the information matrix becomes a diagonal matrix. The dimension of the information matrix is $(N \times N)$.

4.1.2 Limited Sensor Range

The sensor is attached to the vehicle and has its limitations. It is reasonable that the sensor has a limited range, since it has a certain resolution, and it is needed to find a mathematical representation of this limitation. Grocholsky [3] argues that real world sensors typically exhibit an exponential variation in measurement uncertainty up to the maximum range. Introduce a distance r_{max} which is the maximum range of the sensor and introduce an exponential penalty function, such that the covariance of the noise becomes

$$R = \sigma_0^2 \exp\left(k_R(\frac{r}{r_{max}})^2\right), \tag{4.3}$$

and the expected information is proportional to the inverse of the covariance as

$$R^{-1} = \frac{1}{\sigma_0^2}\exp\left(-k_R(\frac{r}{r_{max}})^2\right), \tag{4.4}$$

where σ_0 is the standard deviation at zero range, r is the distance between the sensor and the grid point, r_{max} is the maximum range of the sensor and k_R is a constant, $k_R = 4.6$ is used in this chapter. The penalty function is the exponential function of r

$$\exp\left(-k_R(\frac{r}{r_{max}})^2\right),$$

which does not penalize at zero distance, and at distance r_{max} there will hardly be any information observed, and is plotted in Figure 4.1. Other functions could be used and for examples see Grocholsky [3].

4.1.3 Observed Information

The observation model for an area exploration is according to Grocholsky [3]:

$$\mathbf{z}(t) = \mathbf{T}(x, y) + \mathbf{v}(t), \tag{4.5}$$

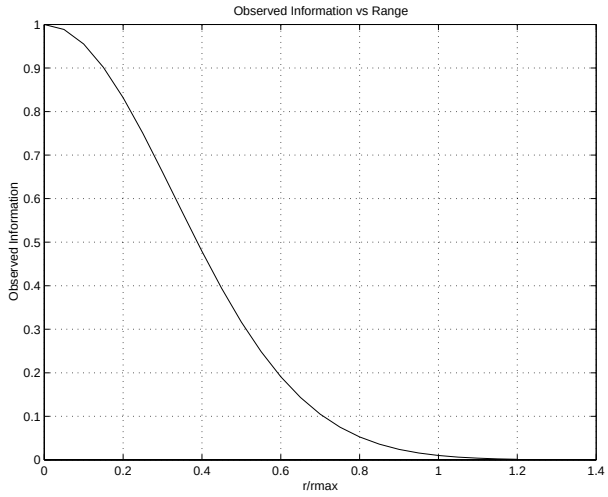


Figure 4.1. Exponential modelling of range dependent measurement errors in a mathematical model of a realistic bearings-only sensor.

where $\mathbf{T}(x, y)$ is a function describing the terrain characteristic and $\mathbf{v}(t)$ zero-mean uncorrelated Gaussian sequence with variance R as a function of the distance r as in (4.3).

In this model the terrain characteristics are said to be the states representing the grid points. For simplicity, we make the assumption that these states could be observed directly by the camera. This means that there are no transformation between the measurements and the states. The reason for making this assumption is that the linear observation matrix H then becomes simply the identity matrix according to (2.14) as

$$\begin{aligned} \mathbf{T}(x, y) &= \mathbf{x}_{gp} \\ \mathbf{z}(t) &= \mathbf{x}_{gp} + \mathbf{v}(t). \end{aligned} \tag{4.6}$$

The information observed from the grid points follows the usual update law as

$$\begin{aligned} I(t) &= H^T(t)R^{-1}H(t) = R^{-1} \\ &= \frac{1}{\sigma_0^2} \begin{bmatrix} \exp\left(-k_R(\frac{r_1}{r_{max}})^2\right) & 0 & \dots & 0 \\ 0 & \ddots & & 0 \\ \vdots & & & 0 \\ 0 & \dots & 0 & \exp\left(-k_R(\frac{r_N}{r_{max}})^2\right) \end{bmatrix}. \end{aligned}$$

Each grid point is stationary, but as for stationary objects, a small process noise is included to improve numerical conditioning, as described in Section 3.1.2. The model for a grid point i is

$$\dot{\mathbf{x}}_{gp_i} = \omega(t), \tag{4.7}$$

where $\omega(t)$ is a zero mean Gaussian process with uncorrelated covariance $Q(t)$. The observed information matrix $I(t)$ is also an $(N \times N)$ matrix and the update law of the information matrix is given by

$$\dot{Y}(t) = -Y(t)QY(t) + I(t).$$

4.1.4 Comments on the Utility Function

A problem with using the determinant as utility function is the numerical difficulties. If the area consists of N grid points with low starting information 10^{-m}. The first determinant calculated would be in the range of 10^{-mN}, e.g., if we have 121 grid points with starting information 10^{-3} the determinant is 10^{-363} which is considered to be zero by Matlab. However, maximizing $\log(\det Y)$ would also maximize the entropic information, recall the definition of entropic information from (2.10)

$$i(\mathbf{x}) = \frac{1}{2}\log\Big((2\pi e)^{-n}\ \det Y\Big).$$

The numerical problems are smaller when using $\log(\det Y)$, but there is still a risk when the number of states is large.

When calculating trace of the inverse information matrix, a summation is done instead of a multiplication and the numerical difficulties could be avoided.

4.1.5 Simulations

In the simulations a quadratic area (10×10) was created, and discretisized with a distance of one between each grid point. This means that there are $11^2 = 121$ points representing the area.

The complexity of the problem is now increased, compared to the simulations in Chapter 3. There are now 121 states for the area plus the original three representing the vehicle, to a total of 124 states, to be compared with for example nine states when localizing an 3D object. There is again need to evaluate the performance of the different utility functions.

Figure 4.2 shows two area searches with different bounds on the control signal. The vehicle starts at the origin with heading upwards and the time horizon is set to 1 s, otherwise the calculation time would be too long. For the same reason the control signal is parameterized into two steps $m = 2$. The background is the information level of the surface, the brighter the more information. The area search is terminated when the information for all grid points is higher than a predefined threshold value.

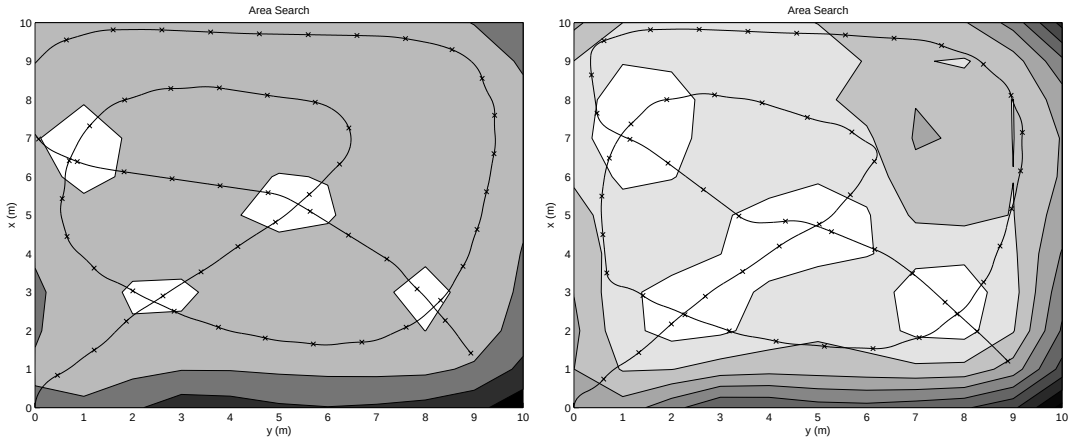


Figure 4.2. Trajectory of sensor platform for an area search with different bounds on the control signals. Left: $-1 \leq u \leq 1$. Right: $-\pi \leq u \leq \pi$

The paths are about the same for the two simulations, but with the tighter bound on the control signal, the vehicle will not turn as sharp. The flight path will begin on the diagonal since there is the most information to gain, but when the sensor's maximum range reaches the outer rim of the area, it turns and continues the search in the direction of maximum information.

4.2 Combined Object and Grid Point Model

It is now possible to define a model which treats the problem of searching an area with n objects. The UAV decides where to go, by solving the optimal control problem while flying, with respect to the grid points and the objects.

4.2.1 Global Information Matrix

The locations of the objects are unknown to the sensor initially and first when an object is in the sensor's range, the information matrix from the grid points are extended with the information matrix from the object. So the global information matrix would be

$$Y_{tot} = \begin{bmatrix} \mathrm{Y}_{gp} & 0 \\ 0 & \mathrm{Y}_{obj} \end{bmatrix} \tag{4.8}$$

where Y_{gp} is defined in (4.2) as

$$Y_{gp}(t) = \begin{bmatrix} \mathrm{Y}_{\mathrm{gp}_1}(t) & 0 & \dots & 0 \\ 0 & \mathrm{Y}_{\mathrm{gp}_2}(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathrm{Y}_{\mathrm{gp}_\mathrm{N}}(t) \end{bmatrix}, \tag{4.9}$$

and

$$Y_{obj} = \begin{bmatrix} \mathrm{Y}_{obj_1} & 0 & \dots & 0 \\ 0 & \mathrm{Y}_{obj_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{Y}_{obj_n} \end{bmatrix}. \tag{4.10}$$

All Y_{gp_i} are scalar values, and the Y_{obj_i} are (3×3) matrices.

As mentioned earlier in (4.4), the observations of grid points are range dependent. This is also true for objects, and the same exponential behaviour is applied to the objects. The information observed from objects is computed in local coordinates, and a measurement provides no information in the direction of the camera. The new range dependent inverse covariance matrix is the same as in (3.29) with the penalty function added as

$$R^{-1}(t) \approx \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_\theta^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_\phi^2} \end{bmatrix} \exp(-k_R(\frac{r}{r_{max}})^2). \tag{4.11}$$

4.2.2 Simulations

Since the grid points and the objects are modelled differently, there is a problem to get balance in the simulations. The first simulation is an area search where five objects

has been placed randomly. The path resembles the paths in Figure 4.2, which could be explained by instead of concentrating on an object, there are more information to be gained by searching the unexplored area. It seems that the model for the grid points gives higher information than the model for the objects, and therefore it is more information to gain by searching the area.

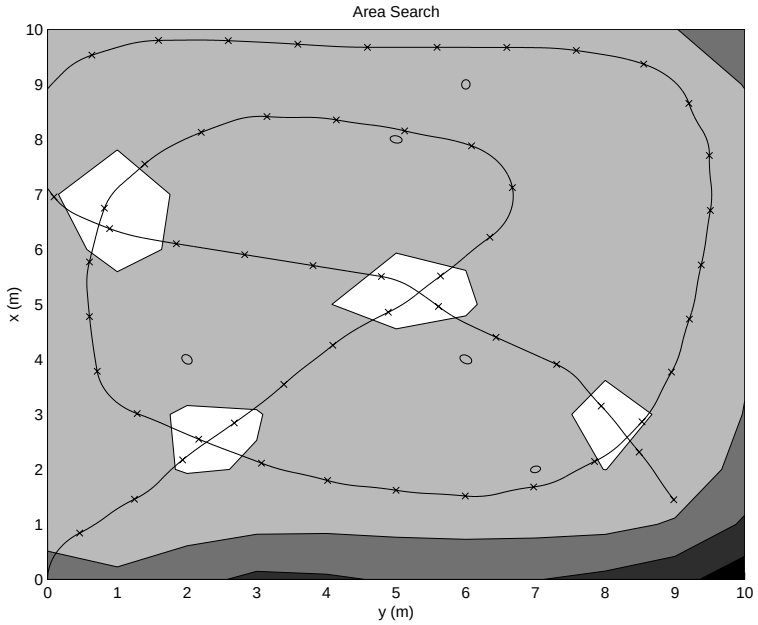


Figure 4.3. Trajectory of sensor platform for an area search with five objects. $-1 \leq u \leq 1$

However, one could set different priorities for objects and grid points. The priorities could be chosen such that the information values from the grid points could be compared to the information values from the objects.

As comparison, a simulation with a weighing matrix W is including in the utility function as

$$J = \text{trace}(W\ P), \tag{4.12}$$

where P is the covariance matrix Y^{-1}. The states representing the objects are weighted higher than the states representing the grid points. The resulting path is shown in Figure 4.4 and the vehicle strives to localize a target within the sensor's range. However, the question of how to choose weights remains open.

4.3 Summary

In this chapter the model was extended so that the vehicle searched an area. Area searches are usually done after a pre-defined path, where the vehicle sweeps the area. The information-theoretic approach lets the vehicle optimize a trajectory at a time horizon, do that it can take a good step depending on where it is and what it has already seen. In this way, the vehicle can act autonomously.

The combined model of an area with objects was also evaluated. The models for grid points and objects differs, and by introducing weight matrices, these weights or priorities could be set by the user according to the mission.

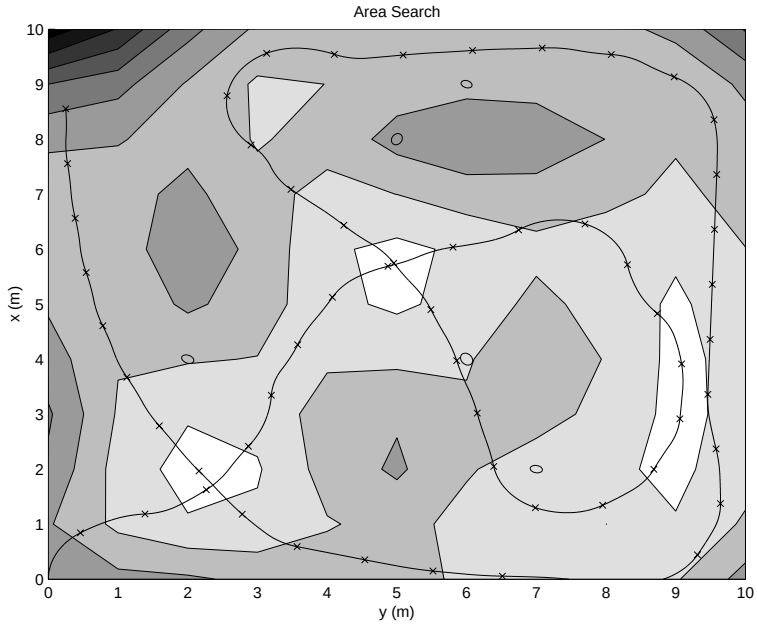


Figure 4.4. Trajectory of sensor platform for an area search consisting of five objects with weighted utility function. $-1 \leq u \leq 1$

Chapter 5

Spline Optimization

In previous chapters the optimization has been over a single parameterized control signal. In Chapter 6 a gimballed sensor system with two degrees of freedom (pan and tilt) is introduced. The optimal control problem would then be over three parameterized control signals, which implies a heavy computational load.

In this chapter we will introduce an alternative approach, where the optimizer searches a number of path points. The complete trajectory is computed with an interpolation method, e.g. splines. This approach is referred to *spline optimization* in this report (even if it is possible to use other interpolation methods) and the result is compared with the optimal control method.

In this chapter, we consider one path representing the flight path of the vehicle, but in Chapter 6 the approach is extended with a second path representing the pointing directions of the sensor. Thus, the optimization result would be a number of vehicle points and sensor points describing how both the vehicle and sensor should move respectively.

5.1 Properties of Splines Optimization

If a limited field-of-view and a specific gaze direction of the sensor are introduced, the optimal control problem must include three parameterized control signals, one for the vehicle and two for the pan and tilt axes of the sensor gimbal. However, if the flight path of the vehicle instead is directly represented by vehicle points at constant altitude and the positions where the sensor are gazing are represented by sensor points on the ground, the number of control signals are reduced from three to two, which would hopefully decrease the computational time. Another advantage is that it is now easier to force the vehicle to pass through certain positions, by making them path points.

The path points must be placed in such way that the path is feasible and the control signals corresponding to the path are bounded. This is done by introducing geometrical bound on the path points as shown in Figure 5.1. The path points are the bigger points, and the smaller points are the interpolated points in between. The first constraint is that the path points must be at a certain distance d from each other. The second constraint is that the angle φ is bounded. φ_i is the angle between two lines, one line going through path points $i-2$ and $i-1$ and one line going through path points $i-1$ and i.

Thus, this approach simplifies the utility function, since the optimizer search for a path instead of control action. However, geometrical nonlinear constraints are more complex than the constant bounds of the control signal in the optimal control approach.

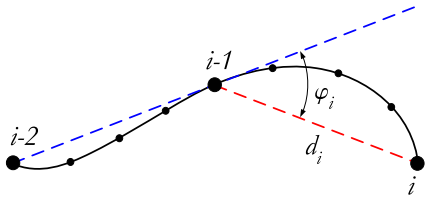


Figure 5.1. Geometrical bounds on the path. The bigger points are the path points and the smaller points interpolated (e.g. spline) points in between. The distance d between the path points is bounded, as well as the angle φ in the intersection of two splines.

5.2 Optimal Path for Bearings-only Localization

Consider the example in Chapter 3, but now optimizing path points instead of control signals. The information is in three dimensions, with flight height 1 m and it would be the same situation as the left plot of Figure 3.16.

The feature and sensor is modelled as in Chapter 3. The states representing the vehicle are removed from the model and the spline is used. The position could be taken directly from the spline and the angle ψ could be calculated by using the path points.

Instead of having a parameterized control signal as variable in the optimization process, a number of points is used which should be placed according to a given start point and the direction of the previous spline. The output of the optimizer is the path points, who together build up the resulting trajectory. The problem with several minima of the utility function is still there, and the resulting trajectory may not be optimal, but hopefully good enough. Then the states are calculated by solving the update law of the information matrix, where observations are made at the path points and interpolated points in between. The nonlinear bounds on the spline are introduced in order to get a feasible flight path.

In the first simulations, the parameters describing the spline are

$d = 1\ m$	Distance between the path points constant.
$\varphi = 30°$	Angle bound between the splines.
$n_s = 3$	Number of path points in each optimization step.

And the other simulations parameters are

$[x_f \quad y_f] = [10 \quad 10]\ m$	Feature location
$Y(0) = \mathrm{I}_{3\times3} \cdot 10^{-3}$	Low starting information
$Q = \mathrm{I}_{3\times3} \cdot 10^{-6}$	Process noise
$R = \sigma^2$, $\sigma = 2.5°$	Observation noise

The distance between the path points is kept constant to resemble constant velocity, and the angle bound is there to avoid large control signals. $M = 3$ means that three path points are placed and they are building up the trajectory for the vehicle. The number of path points and how long they may be placed from each other are setting the length of the trajectory which resembles the optimization time horizon. The information at the final path point, which is depending on the trajectory, is building up the information matrix from where the utility function is calculated.

The resulting trajectory is still shaped as a spiral since the information gain is most favorable orthogonal to the last observed direction and inverse proportional to

the squared distance. The spline optimization gives about the same result as for the control signal optimization, which is expected since the same information model are used.

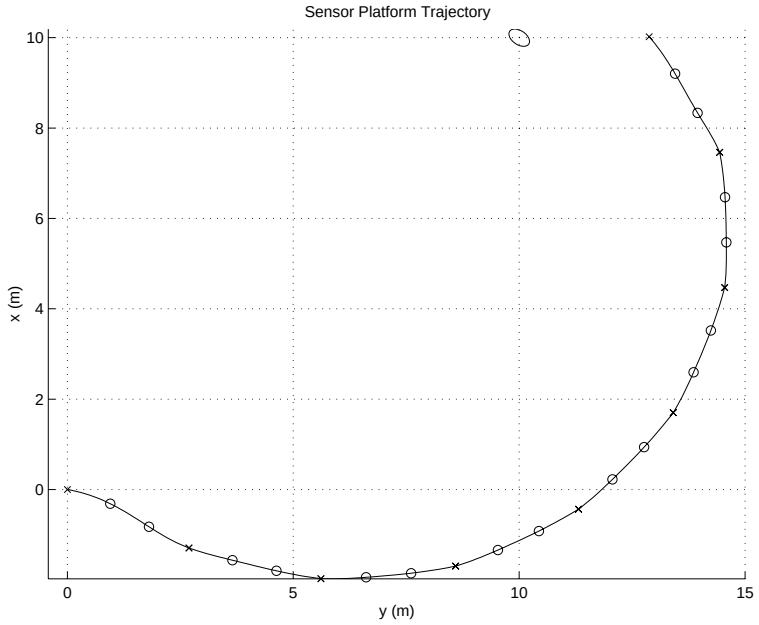


Figure 5.2. The optimal trajectory with constant distance 1 m between the path points. Each path point is marked with an 'x' or a 'o', the 'x' points indicates a new optimization.

The distance between the path points could vary and is shown in the left part of Figure 5.3. The distance is set to 2 m and 3 m respectively. The situation resembles much of the effect of different optimization time.

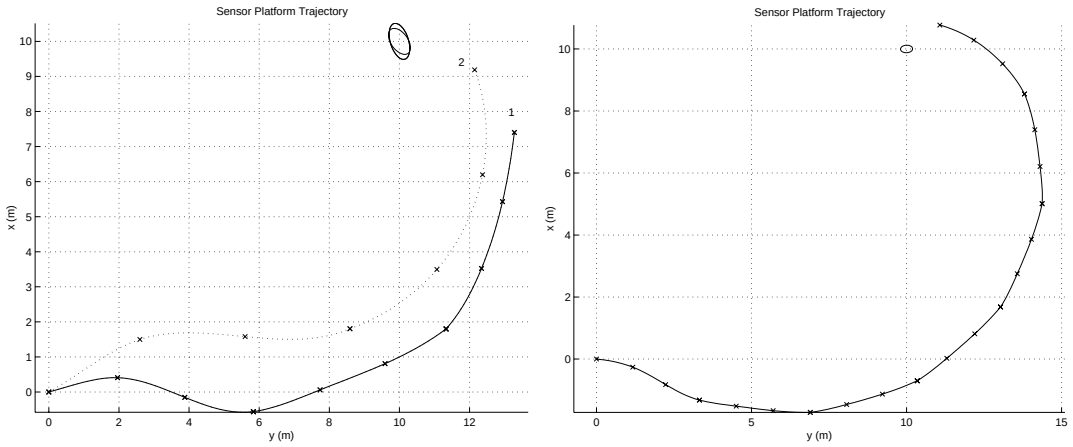


Figure 5.3. Left: The resulting trajectory for distance between path points: 1) 2 m and 2) 3 m. Right: The resulting trajectory for a path with distance varying 0.8 and 1.2 m between two spline points. All path points are marked with an 'x'.

The simulations are so far very similar to the simulations in Chapter 3. Arguing for the use of splines, it is very easy to place the path points with different distance in between. The next simulation is the same as for in Figure 5.2 but the distance is allowed

to vary within 20%, i.e., the path point distance is between 0.8 m and 1.2 m. This would be harder to implement in the original model. The bound on the angle is tightened to 20° in order to avoid some local minima, the same problem as in Section 3.2.1. The resulting path is shown in the right part of Figure 5.3 which requires less optimization steps than for the simulation in Figure 5.2 since the vehicle will take longer step far away from the feature and could use shorter step if necessary.

A problem with splines is overplanning, which means that a spline is too long and path points must be placed far from the feature. By letting the length between the path points vary with the distance to the target as seen in Figure 5.4 but not the angular bound, there is no risk to overplan and instead the vehicle will localize the object faster. This opens up for two interpretations. The first is that the vehicle will no longer travel with constant velocity, but instead travel fast when it is far away and slow when it is close to the object. The number of observations is the same between any two path points and more observations are made closer to the object. The other interpretation is that the velocity is still constant and instead the number of observations varies with the distance to the target.

A spline optimization simulation is faster than the optimal control problem. This is promising since the main reason for introducing splines was the computational time.

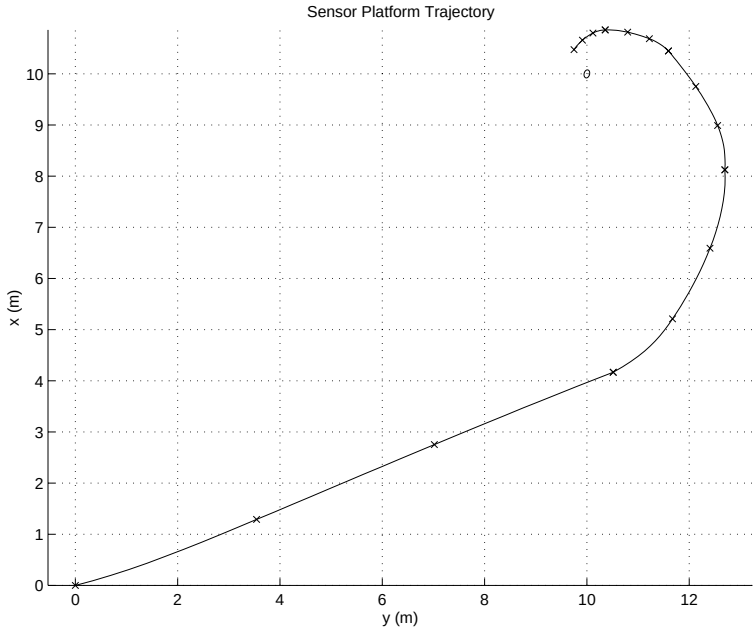


Figure 5.4. The optimal trajectory with varying distance in each subspline.

5.3 Comments

The method of using splines is appropriate since the resulting trajectory is about the same as for optimizing over control signals. Path points are very flexible since it is easy placing the points at different distance between each other. Splines are also preferable if the user knows that the UAV must pass a certain point.

There is still the problem with many optimum in the utility function, and the number of local optimum increases with more path points or less tighten bounds. The starting

values are also important for a successful result. Future work will include a coarse planning approach that will produce starting points.

A recursive filter algorithm is also replacing the ODE solvers. In the remaining chapters of this report the recursive filter is used.

Chapter 6

Limited Field-of-View and Occlusion

So far in this report we have assumed that the sensor is able to see in all directions all the time. Furthermore, we have assumed that all objects and grid points are visible from all positions. The only limitation has been an distance dependent noise, i.e. observations contains more information the closer the feature is. In this chapter we introduce a more realistic sensor model together with a basic occlusion concept in order to be able to handle more realistic scenarios.

6.1 Experimental Sensor System

The sensor system that serves as pattern is quickly presented here. The sensor system is a gimballed EO/IR system with an IR QWIP sensor and a CCD video sensor, see Figure 6.1. Field-of-view of the QWIP sensor is $20° \times 15°$. The gimbal has two controllable axes, pan and tilt. Pan axis is able to rotate 360°, and the tilt axis is able to rotate approximately $[10° - 90°]$.

Figure 6.1. The gimbal system. Left: Inner gimbal consists of an IR camera and a color CCD. Middle: The gimbal with demounted front. Right: Gimbal with mounted front.

6.2 Sensor Pointing Path

The sensor pointing path is obtained in the same way as the vehicle path described in Chapter 5. The optimizer searches for n_s vehicle points and n_c sensor pointing points at the same time, see Figure 6.2. The vehicle and sensor points are in two parallel planes, π_s

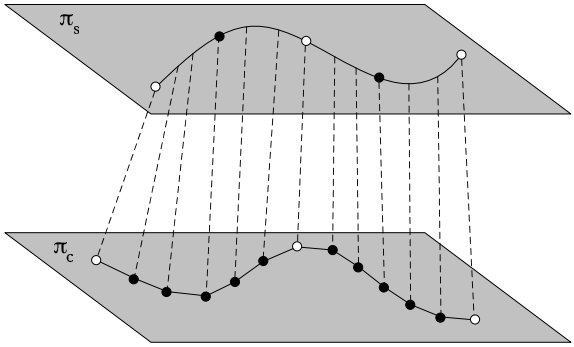


Figure 6.2. Spline optimization with the vehicle path in the upper plane and the sensor gazing path in the lower plane.

and π_c respectively. It is not necessary that $n_s = n_c$, in general $n_s < n_c$. An interpolating method is used to compute the complete vehicle and sensor pointing trajectories. If not stated otherwise, a linear interpolating method is used to compute the sensor pointing trajectory. It is possible to add constraints on where to put the sensor points, as for the vehicle points, but in the simulations in this report no constraints for the sensor points were applied. This is reasonable since the gimbal system has "faster" dynamics than the vehicle, i.e. the sensor has an much higher ability to follow an pointing trajectory on the ground than the vehicle has to follow an flight trajectory.

6.3 Limited Field-of-View

The sensor has a limited field-of-view (FOV) and therefore it is only possible to see features (objects and grid points) that are inside the sensor footprint on the ground, Figure 6.3.

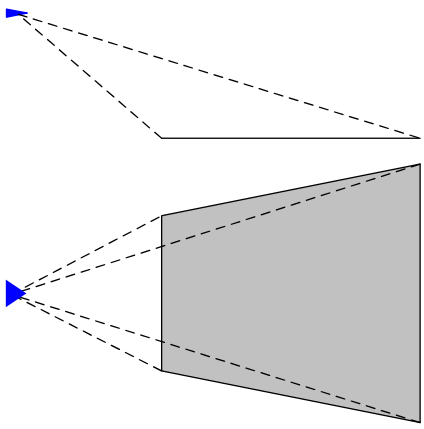


Figure 6.3.

Let $\mathbf{x}_c = [x_c \quad y_c]^T$ be the sensor pointing point in the plane π_c. Furthermore, let $\mathbf{x}_s = [x_s \quad y_s]^T$ be the vehicle point in the plane π_s. Thus, the sensor gaze direction is described by the vector $\mathbf{r} = \mathbf{x}_c - \mathbf{x}_s$, let $\hat{\mathbf{r}}$ be a normalized vector in the direction of $\mathbf{r}$. Assume that the sensor is a pin-hole camera and project the image plane *abcd* on the plane π_c, see Figure 6.4. The resulting polygon $ABCD$ describes the sensor footprint.

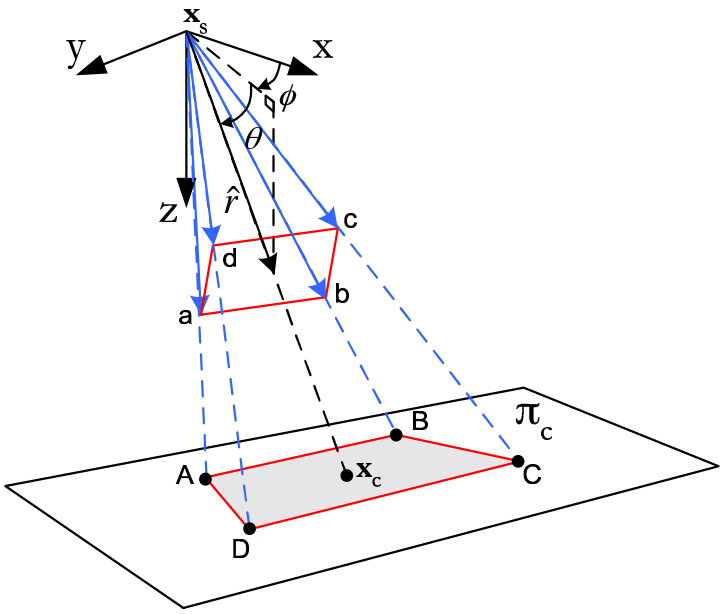


Figure 6.4.

The optimization process will fail if we state that all features inside $ABCD$ is fully visible and all features outside $ABCD$ is not seen at all. The optimization requires an smooth transition between visibility and invisibility, or in other words, the optimizer requires an non-zero gradient on the information surface to know in which direction the information increases.

In practice, a smooth and continuous threshold function is defined for each edge. Let w be the perpendicular distance to the line going through the edge, i.e $w > 0$ "outside", $w = 0$ on the line/edge and $w < 0$ "inside", see Figure 6.5. The threshold function is

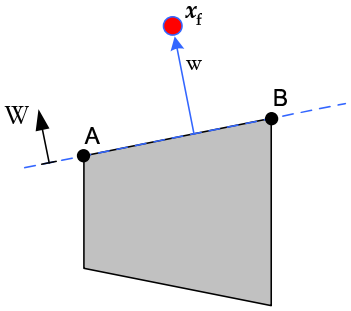


Figure 6.5.

a product of an arctan and and exponential function that is near 1 when $w \ll 0$ near 0 $w \gg 0$. Thus,

$$f_{edge}(w) = f_{atan}(w) f_{exp}(w) \tag{6.1}$$

where

$$f_{atan}(w) = 1/2 - \frac{1}{\pi}\arctan(k_1 w) \tag{6.2}$$

and

$$f_{exp}(w) = \begin{cases} 1, & w < 0 \\ \exp(-k_2 w^2), & w \geq 0 \end{cases} \tag{6.3}$$

The threshold behaviour comes from the arctan function, the exponential function is added to force the values faster to zero when w increases. The parameters k_1 and k_2 affects the slope and may also be dependent on w.

Finally, by taking the product of all four edge threshold functions a new smooth and continuous threshold function f_{FOV} is obtained with values near 1 inside $ABCD$ and near 0 outside, i.e.,

$$f_{FOV}(\mathbf{x}_s, \mathbf{x}_c) = f_{edge}(w_{AB}) f_{edge}(w_{BC}) f_{edge}(w_{CD}) f_{edge}(w_{DA}) \tag{6.4}$$

Figures 6.6 shows some examples of the sensor footprint and figures 6.7 shows the visibility value in the point (30,0) as a function of $\mathbf{x}_c$.

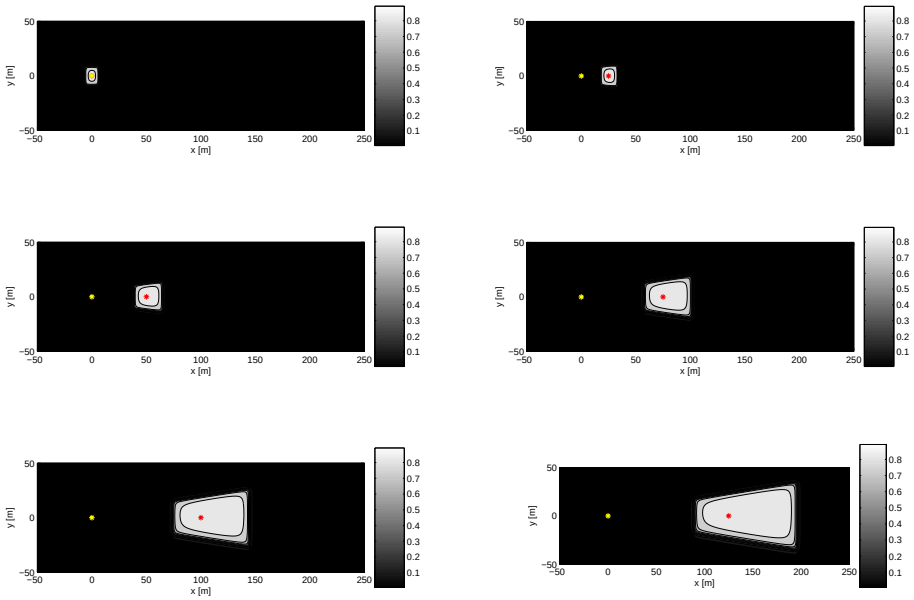


Figure 6.6. Sensor footprint examples. Sensor located in (0,0) at height 50 [m]. Sensor pointing points are $\mathbf{x}_c = (0,0)$, $\mathbf{x}_c = (25,0)$, $\mathbf{x}_c = (50,0)$, $\mathbf{x}_c = (75,0)$, $\mathbf{x}_c = (100,0)$, $\mathbf{x}_c = (125,0)$.

6.4 Occlusion

In high-altitude surveillance the 3d structure of the ground may be ignored and the ground can be approximated with a plane. However, in low-altitude surveillance, especially in urban environments, the 3d structure give rise to occlusion that can not be ignored, Figure 6.8.

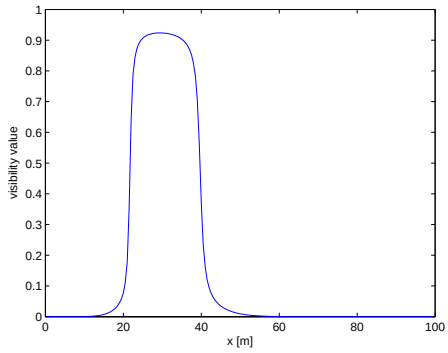


Figure 6.7. Visibility value in the point (30,0) as a function of $\mathbf{x}_c$, $x_c \in \{0, 100\}$, $y_c = 0$.

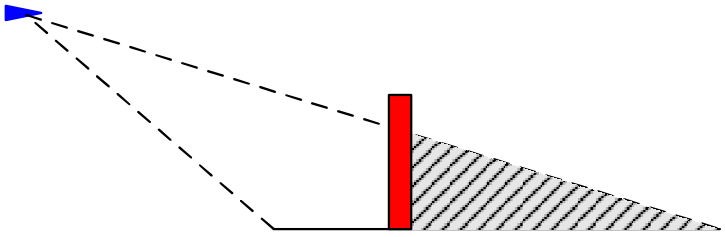


Figure 6.8. Occlusion.

In this report only a simple 2D angle-dependent occlusion model is used, see below. It is possible to achieve an occlusion model that also depends on the angle of depression by using the range parameters r_{max} and k_R introduced in Chapter 4. More complex models will be developed in the future. OpenGL and OSG tools will be used to achieve more realistic models of the occlusion.

6.4.1 2D Angle-Dependent Occlusion Model

One simple model of occlusion is to consider the angle of incidence to a feature in the horizontal plane, i.e. the angle of depression is ignored.

Each feature has an visibility function $f_{occ}(\gamma)$ where γ is the angle of incidence, see Figur 6.9. Furthermore, the function is bounded as $0 \leq f_{occ}(\gamma) \leq 1$ and defined for

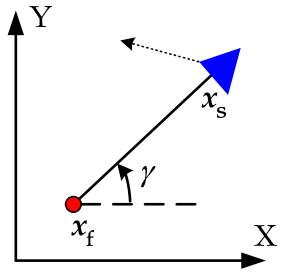


Figure 6.9. γ, angle of incidence.

$\gamma \in [-\pi, \pi)$. f_{occ} near one means that the feature is seen and f_{occ} near zero that the feature is occluded. Figure 6.10 shows four examples of the visibility function $f_{occ}(\gamma)$ and the corresponding symbol that will be used in the next chapter. Figure 6.11 shows

an example of how the occlusion may be defined in a road surveillance example with foliage occlusion.

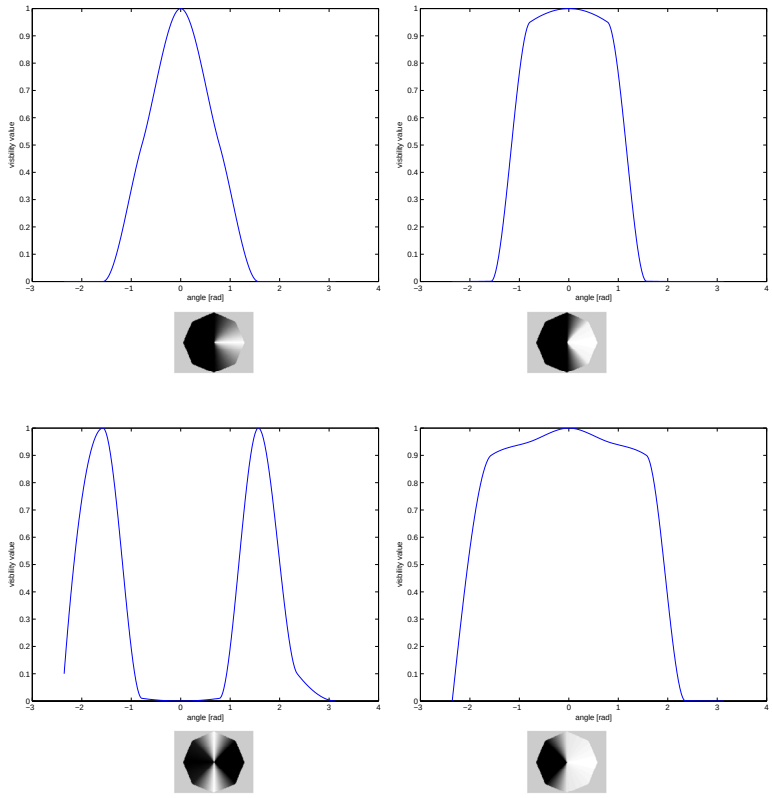


Figure 6.10. Four examples of the visibility function $f_{occ}(\gamma)$.

6.5 Observed Information

The visibility functions f_{FOV} and f_{occ} are used to reduce the observed information I_f of the feature f, or equivalently, varying the observation noise as in Section 4.1.2 where a distance dependent function was used to model the limited sensor range. Denote this factor as f_{range} and the reduced observed information used in the filter update can be written as

$$I_f^{reduced} = f_{FOV}(\mathbf{x}_s, \mathbf{x}_c) f_{occ}(\mathbf{x}_s, \mathbf{x}_c) f_{range}(\mathbf{x}_s, \mathbf{x}_c) I_f \tag{6.5}$$

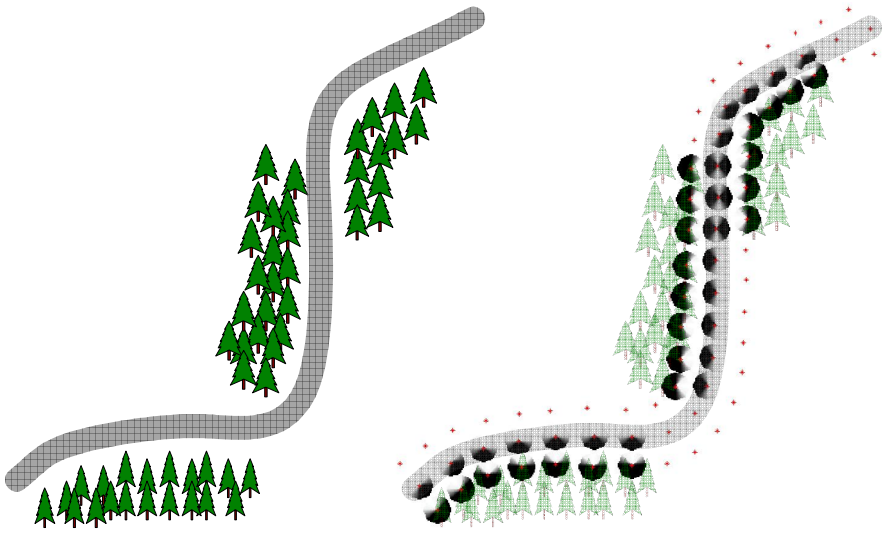


Figure 6.11. An example of how the 2D occlusion models may be defined in a road surveillance example. A number of grid points are placed along the road. Each grid point has an occlusion model, indicated by the circular symbol.

Chapter 7

Road and Building Surveillance

In this chapter we put all pieces presented in this report together. We use concurrent path and sensor planning to solve a road and building surveillance task.

7.1 The Road and Building Surveillance Scenario

The scenario in this chapter is based on a real world environment, Figure 7.1. The task

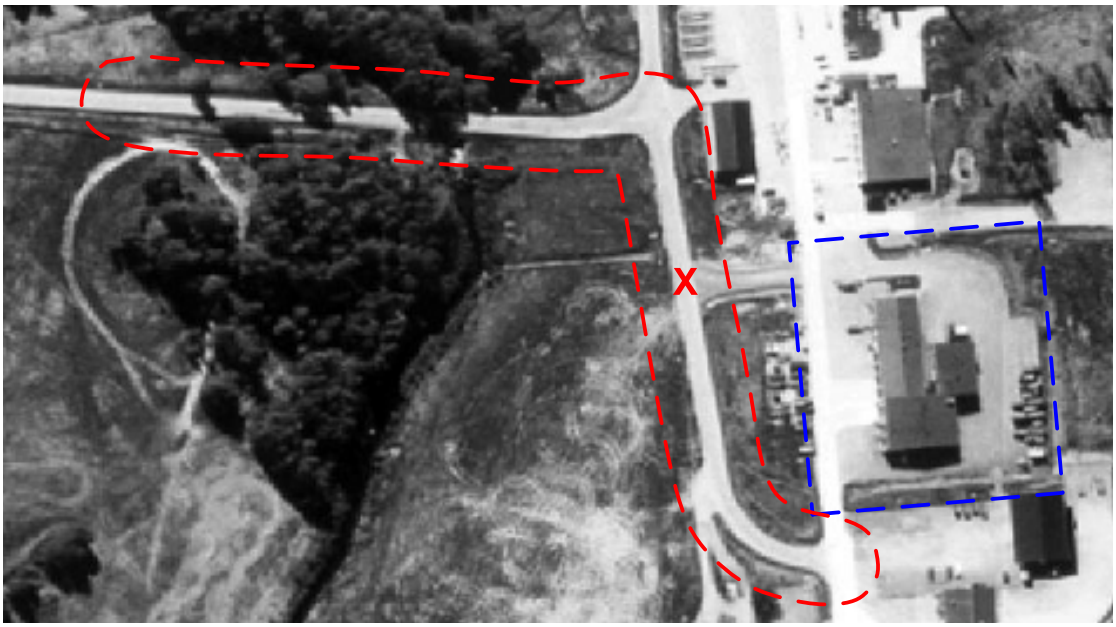


Figure 7.1. Road surveillance scenario. A road segment (red lines) and a building (blue rectangle). In addition, there is an object on the road.

is to survey a road segment and a building, and to detect and localize an object on the road. A number of grid points along the road and around the building are set out. The number of grid points is a compromise between long computational time and fair area coverage. An occlusion model is associated with each grid point, and points on the edge of a forest and around the building are partly occluded, see Figure 7.2.

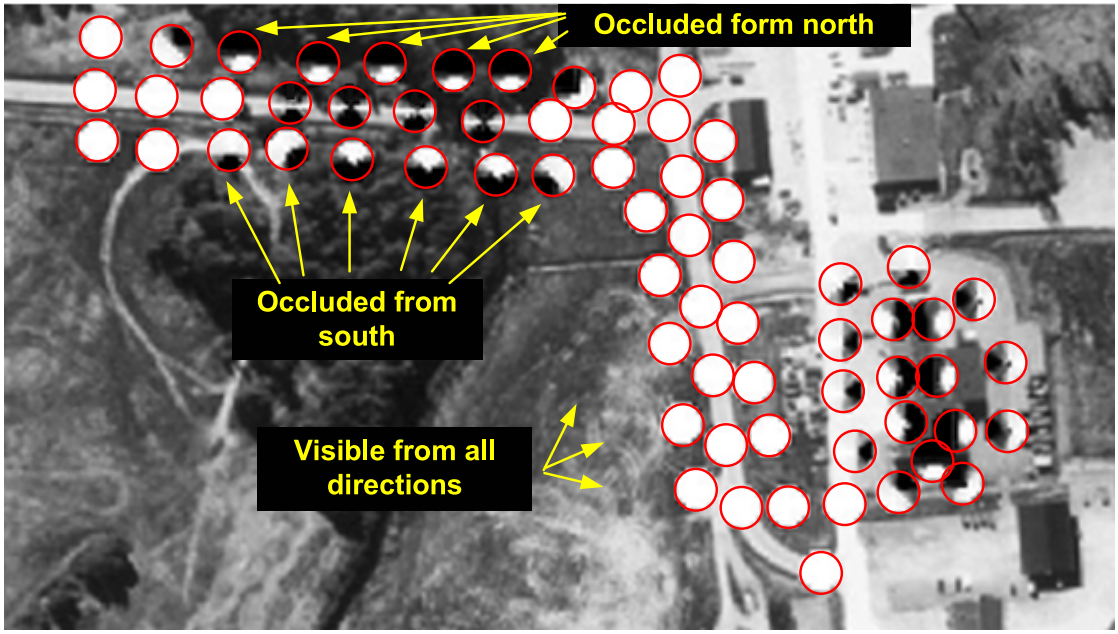


Figure 7.2. An occlusion model is associated with each grid point.

7.2 Replanning

In real world applications our knowledge could never be complete. The plan at time t_0 is based on the knowledge about objects and environment at time t_0. However, during the execution of the plan, new knowledge is obtained through observations of the world. Thus, we need to replan.

In the simulations in this chapter a replanning is done if a object is "detected" or "localized". A object is "detected" and "localized" if its information level (determinant or trace inverse of the information matrix) is above some threshold $I_{detected}$ and $I_{localized}$, respectively. For instance, at time t_0 a new plan for the time span $[t_0, t_0+t_T]$ is generated. During the execution of this plan, a new object is discovered at time t_1. Then a new plan is generated for $[t_1, t_1 + t_T]$, see Figure 7.3.

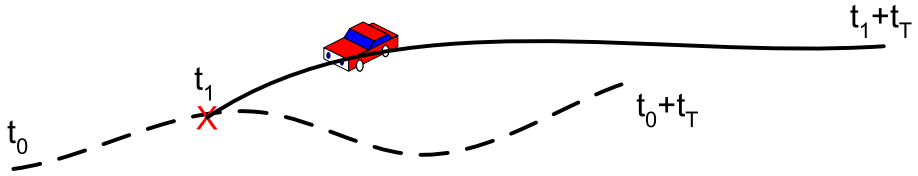


Figure 7.3. Replanning at time = t_1.

7.3 Simulation Result

The simulation result is highly dependent on the simulation settings. The purpose of this section is not to show how different parameter sets affects the result. Only one example is given to illustrate how the methods presented in this report may solve this road surveillance task.

Figure 7.4 shows the path of the vehicle together with the information values at each grid point. Figure 7.5 shows, besides the vehicle path, the gaze direction of the sensor at some points. In Figures 7.6 and Figure 7.7 two areas of Figure 7.5 is zoomed in. Figures 7.6 shows the area around the object, it can be seen how the sensor is gazing at the objects as the vehicle is approaching and then leaving the object. In Figures 7.7 the vehicle is flying round the building as the sensor is pointing at the building. This is reasonable according to the occlusion definitions of the grid points, the sensor can not "see" all grid points from just one direction.

In Figures 7.8 the sensor footprint is illustrated as a contour plot, white represent the area that the sensor is "seing". The position of the vehicle is indicated by the star with circle. In Figures 7.9 the sensor is gazing at the detected object. Note the covariance ellipse of the object. Note that this covariance is based on the information state of the object; thus, it is not the position uncertainty. The pentagram shows the position of the vehicle when the object was "detected" and a replanning was performed.

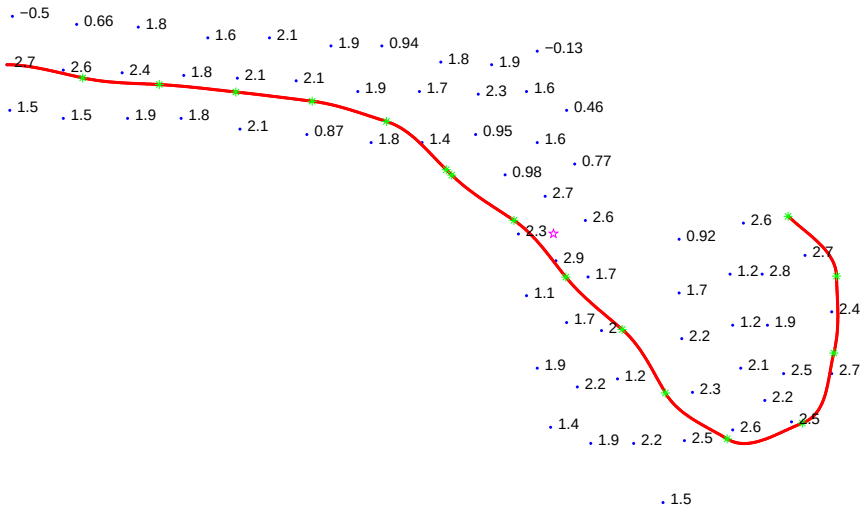


Figure 7.4. Vehicle path and information values at each grid point.

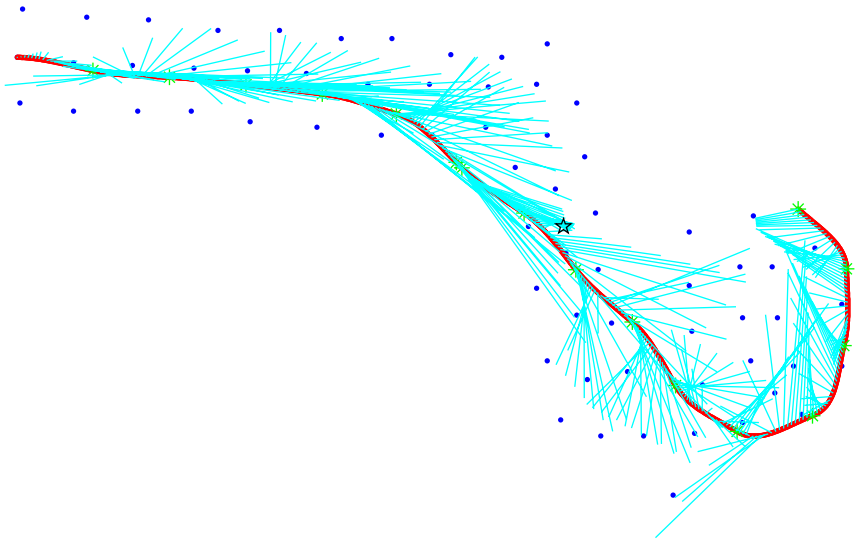


Figure 7.5. The gaze direction of the sensor at some points.

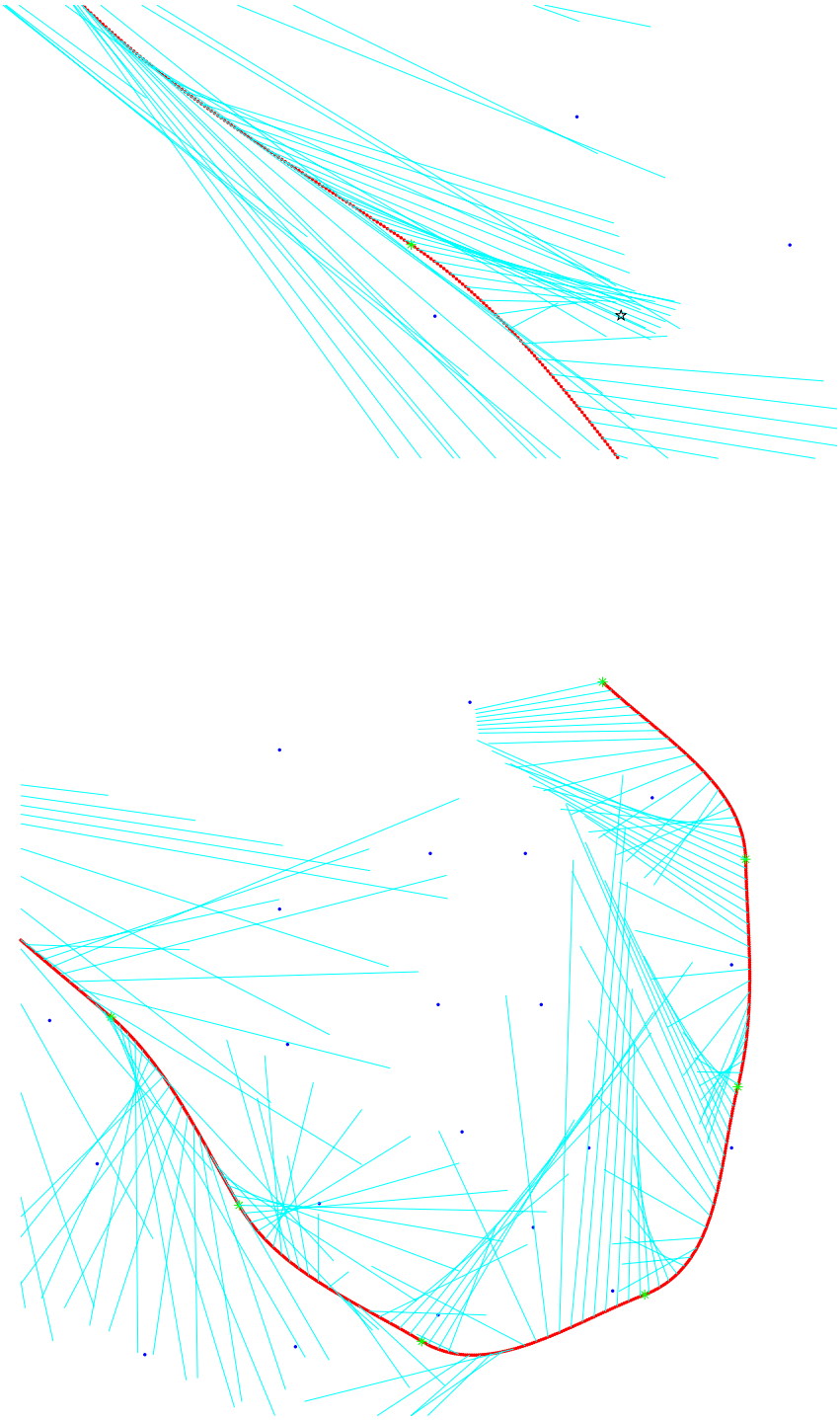


Figure 7.7. The gaze direction of the sensor at some points. Area around the building.

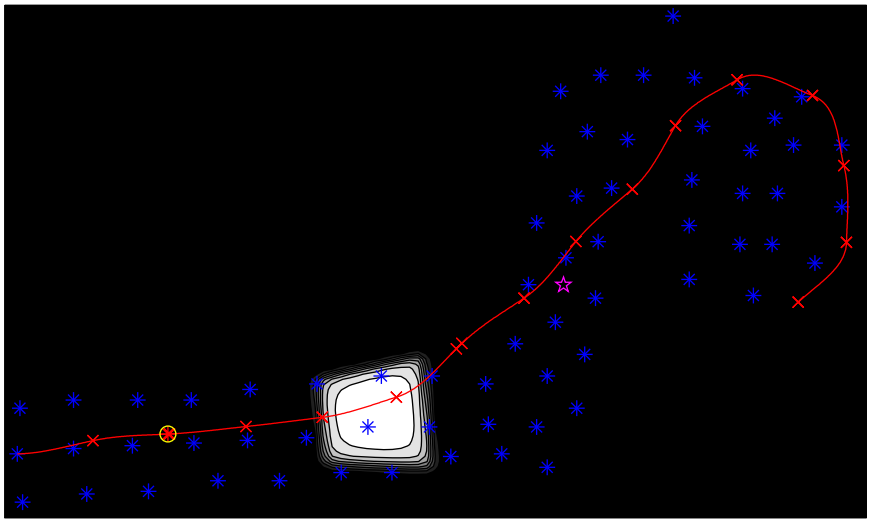


Figure 7.8. Sensor footprint illustrated as a contour plot.

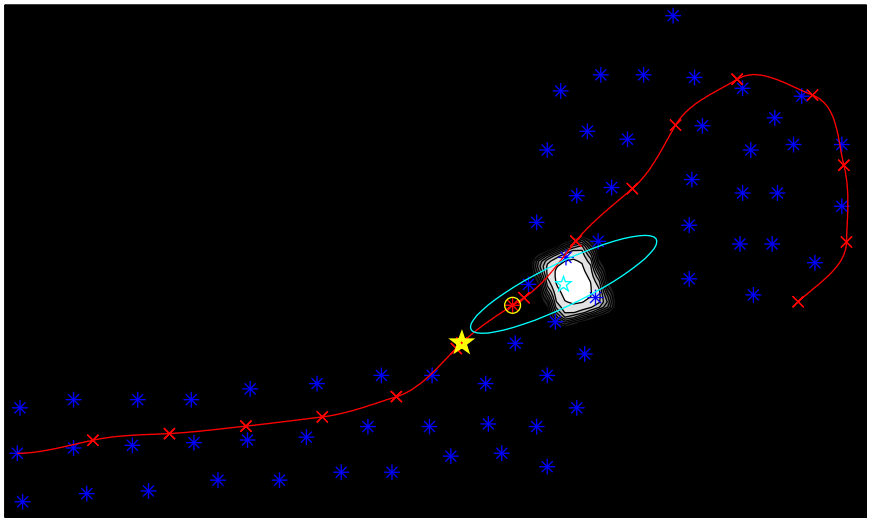


Figure 7.9. Sensor gazing at the object.

Chapter 8

Conclusions

This report is concerned with an information-theoretic approach to concurrent path and sensor planning for a UAV with gimballed EO/IR sensors.

8.1 Summary

Technically, *information* is a measure of the accuracy to which the value of a parameter or stochastic variable is known. There are two commonly used formal definitions of information, the *Entropic information* and *Fisher information*. The report is concerned with UAV surveillance, i.e. a UAV with an EO/IR-sensor searches for objects and once found, the object should be localized. An information measure is defined based on the feature (object and grid point) states. The goal in the planning is to found a UAV path and sensor gaze directions that maximizes this information measure.

The path planning problem is first formulated and solved as an optimal control problem. The second approach is using interpolation methods, like splines, to solve the path planning problem, as well as, the combined path and sensor planning problem. This approach is referred to *spline optimization* in this report, even if it is possible to use other interpolation methods.

For both approaches, an information matrix is constructed from the states representing the objects and grid points. A utility function is defined based on the information matrix and the planning process is to find variables (control signals or path points) that maximize the utility function at the planning horizon.

Different parameters were discussed, such as different utility functions, time horizon and prior information. The methods were applied to one object localization, n objects localization, as well as, area exploration. The spline approach was also extended with a sensor model and two paths were computed, one flight path of the vehicle and one sensor gazing path. Finally, a road scenario simulation was presented.

8.2 Conclusions

The work presented in this report aimed at investigating the usefulness of an information-theoretic approach to the combined path and sensor planning problem. Important sub-problems and questions were neglected and some parts are very "ad hoc" since our goal was a kind of proof of concept.

The information-theoretic approach has several advantages. It is straight forward to generate utility measures in term of information that captures the nature of the problem in an intuitive way. Information is also connected to probability distributions, which are used to describe the uncertainties of the feature estimation. For instance, in the object localization example, the information is appropriate since the higher information about an object, the better is the localization. The information-theoretic approach is also very suitable for decentralized and/or multi-sensor applications. Assuming that sensor observations are conditionally independent, the fusion step from different sensors becomes remarkably simple.

The problem considered in this report is very complex since the problem is a non-convex non-linear problem. Thus, there is no guarantee that the optimization finds the global optimum. However, it is not required that the global optimum is found, a "sufficiently good" local optimum is enough. The choice of optimization algorithm has, so far, been neglected, but this question is very important and must be considered in the future work.

The optimal control solution worked well for the path planning problem, but in the combined path and sensor planning problem the optimal control problem would have to be formulated over three control signals, two for the sensor and one for the vehicle. This seemed like a heavy computational load and therefore the spline optimization approach was introduced. The method of splines is promising since the calculation time is lower than for the optimal control problem, and spline optimization is able to solve the combined vehicle and sensor optimization problem. However, even when spline optimization is used the problem is very complex and the computation time is long. The tolerances of the solvers and the optimizer must be carefully chosen to avoid too long computation time or, on the other hand, result that is nonsense.

As seen in the simulations, different utility functions result in different paths. It is hard to choose one utility function as the best in all situations. In the example with only one object, the resulting paths are rather similar, but when the problem increases with more objects and grid points there are larger differences. More research and simulations must be done.

Several assumptions and simplifications have been done

- Gaussian distributions are assumed throughout the report. Linearized models are used and the filter is a (rewritten) extended Kalman filter.
- The observation noise for different features are assumed to be independent.
- The planning process is using the position of the object to compute the expected information. Current implementation are using the true position of the object, but this knowledge are, of course, not known and must be estimated if the object are detected and not known in advance. The planning should also take the uncertainty into account and show a cautious behaviour that will reduce the negative effects then the estimates are misleading.
- The vehicle model is very simple. The vehicle is assumed to fly at constant altitude with constant (optimal control) or almost constant (spline optimization) speed. Furthermore, the sensor gaze points must also lie in one plane.
- The navigation uncertainty is ignored. In order to get the simulations more realistic the navigation uncertainty of the vehicle must be considered, but this will lead to

a quite complex model and is saved for future work.

8.3 Future Work

There are a many questions and problems left for the future work:

- More research in the information-theoretic formulation and the choice of utility function must be done.
- The optimization algorithm should be replaced with a more suitable algorithm.
- Better models of environment (occlusion) must be developed to handle more realistic scenarios. Environment uncertainties should also be added.
- More constraints, for instance, tactical, fuel, and time constraints should be added.
- Better vehicle and sensor models must be developed. The navigation uncertainties should be taken under considerations.
- More research in the cooperating platforms problem.

Bibliography

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, 2000.

[2] T. Glad and L. Ljung. *Reglerteori. Flervariabla och olinjära metoder*. Studentlitteratur, Lund, 1997.

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

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

[5] J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Managment: A Decentralized Information-Theoretic Approach*. Ellis Horwood, London, 1994.

[6] J. Nygårds, P. Skoglar, J. Karlholm, M. Ulvklo, and R. Björström. Towards concurrent sensor and path planning - A survey of planning methods applicable to UAV surveillance. Scientific Report FOI-R–1711–SE, ISSN 1650-1942, Swedish Defence Research Agency (FOI), Divison of Sensor Technology, SE-581 11 Linköping, Sweden, 2005.

[7] Y. Oshman and P. Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(3):892–902, July 1999.

[8] 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.

[9] P. Skoglar. Modelling and control of EO/IR-gimbal for UAV surveillance applications. Scientific Report FOI-R–0893–SE, ISSN 1650-1942, Swedish Defence Research Agency (FOI), Divison of Sensor Technology, SE-581 11 Linköping, Sweden, June 2003.

[10] O. Trémois and J.-P. Le Cadre. Optimal observer trajectory in bearings-only tracking for maneuvering sources. *IEE Proceedings Radar, Sonar and Navigation*, 146(1):31–39, February 1999.

[11] 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.